Available online at www.CivileJournal.org

Civil Engineering Journal

(E-ISSN: 2476-3055; ISSN: 2676-6957)

Vol. 11, No. 01, January, 2025

Benchmarking Classical and Deep Machine Learning Models for Predicting Hot Mix Asphalt Dynamic Modulus

Waleed Zeiada [1, 2], Lubna Obaid [1*], Sherif El-Badawy [2], Ragaa Abd El-Hakim [3], Ahmed Awed [2]

[1] *Department of Civil and Environmental Engineering, University of Sharjah, Sharjah P.O. Box 27272, UAE.*

[2] *Department of Public Works Engineering, Mansoura University, Mansoura, Egypt.*

[3] *Department of Public Works Engineering, Tanta University, Tanta, Egypt.*

Received 08 August 2024; Revised 14 December 2024; Accepted 22 December 2024; Published 01 January 2025

Abstract

The dynamic modulus (|E*|) of hot-mix asphalt (HMA) is a crucial mechanistic characteristic essential in defining the strain response of asphalt concrete (AC) mixtures under varying loading rates and temperatures. This paper aims to conduct a comprehensive investigation of classical machine learning (ML) and deep learning (DL) algorithms as applied to the prediction of |E*| and compare their performance with renowned |E*| regression models (Witczak NCHRP 1-37A, Witczak NCHRP 1-40D, and Hirsch). Eight state-of-the-art ML and DL algorithms are attempted with diverse structures, including multiple linear regression (MLR), decision trees (DT), support vector regression (SVR), ensemble trees (ET), Gaussian process regression (GPR), artificial neural networks (ANN), recurrent neural networks (RNN), and convolutional neural networks (CNN). A comprehensive database was assembled, incorporating 50 AC mixtures, of which 25 were from the Kingdom of Saudi Arabia and 25 were from the state of Idaho, USA. This database encompasses an extensive dataset of 3,720 |E*| measurements, associated with thirteen input features representing the proposed AC mixtures' aggregate gradations, binder characteristics, and volumetric properties. This pioneering study surpasses existing research by examining various algorithms to predict |E*| on the same dataset, applying them with different structures and individual optimization to achieve optimal performance. The developed models are evaluated based on multi-stage assessment criteria, including the accuracy and complexity performance measures and rationality based on a sensitivity analysis. The multi-stage comparative analysis results reveal that the bagging ETs, GPR with exponential kernel, and DT record the highest prediction accuracy; however, only the bagging ETs yield the highest accuracy, lowest training and testing complexity, and rational trends throughout the sensitivity analysis. The research outcome has the potential to provide pavement engineers with advanced tools for predicting |E*| and, therefore, optimizing pavement designs and rehabilitations.

Keywords: Dynamic Modulus; Hot Mix Asphalt; Feature Engineering; Classical Machine Learning; Deep Learning; Comparative Analysis.

1. Introduction

Hot-mix asphalt (HMA) forms the backbone of transportation infrastructure, providing smooth and durable surfaces upon which people rely for safe and efficient travel. Ensuring the longevity and performance of HMA under various conditions necessitates accurate prediction of its mechanical properties. Among these properties, dynamic modulus (|E*|) stands as a crucial parameter, representing the temperature- and frequency-dependent stiffness of HMA [1]. |E*| holds fundamental importance within linear viscoelastic materials, particularly in the context of HMA. This critical parameter is determined under continuous haversine loading conditions in the frequency domain, defined mathematically as the absolute value of the complex modulus (E*) [2, 3].

* Corresponding author: lobaid@sharjah.ac.ae

http://dx.doi.org/10.28991/CEJ-2025-011-01-06

The significance of |E*| in predicting pavement performance and assessing the response of asphalt structures is underscored by its integration into the Mechanistic-Empirical Pavement Design Guide (MEPDG) framework and its associated AASHTOWare Pavement Mechanistic-Empirical Design (PMED) software. |E*| has been integrated within this framework as a critical input parameter for mechanistic response estimation and empirical performance prediction. However, the availability of |E*| values may be limited for projects of lower functional class. To address this, the MEPDG offers a hierarchical structure of inputs, ranging from level 1 (direct laboratory measurement) to levels 2 and 3 (empirical estimation based on mixture volumetrics and binder properties) [4, 5].

Existing testing methods for |E*|, such as laboratory-based dynamic modulus tests, are well-established but are also known for their drawbacks, including their time-consuming nature, high costs, and labor-intensive procedures [6]. These challenges often restrict practitioners from obtaining reliable |E| measurements during early design stages, particularly when materials are unavailable or when dealing with lower-class pavement projects. To mitigate these constraints, researchers have sought alternative methods to estimate |E| accurately, giving rise to a range of empirical and statistical models [7, 8].

Among these, regression-based predictive models have played a pivotal role in the evolution of |E| estimation [9]. For example, the viscosity-based (η) Witczak model, introduced under the National Cooperative Highway Research Program (NCHRP) 1-37A, is a foundational predictive tool incorporating asphalt mixture properties to estimate |E|. Developed using a robust dataset of 2,750 |E*| measurements from 205 HMA mixtures, this model has been instrumental in pavement design for decades [10]. Building on this, Bari and Witczak extended the model into the NCHRP 1-40D framework, incorporating advanced parameters such as shear modulus (G*) and phase angle (δ) to improve its accuracy [11]. These models were later selected as global prediction models of |E*| for the MEPDG's levels 2 and 3 analysis.

In parallel, models such as Hirsch and Alkhateeb have provided additional alternatives for |E| estimation. The Hirsch model, which utilizes fewer input variables, demonstrated superior performance under specific conditions and became a popular choice for quick and practical applications [10, 12-14], while the Alkhateeb model combined parallel and series composite theories to describe asphalt concrete behavior, offering reduced input requirements without sacrificing prediction quality [15]. These empirical models remain widely adopted due to their simplicity and accessibility; however, their reliance on predefined relationships limits their ability to capture nonlinearities inherent in asphalt mixture behavior.

Despite the deployment of these regression-based models, there has been growing recognition of the need to address their limitations. Traditional approaches often fail to adapt to the variability of asphalt mixtures and their response to diverse environmental and loading conditions. Consequently, research has increasingly turned to data-driven techniques, particularly machine learning (ML) and deep learning (DL), which have shown substantial promise in improving |E| prediction accuracy. These methods excel in uncovering complex, nonlinear patterns within datasets, allowing them to model intricate interactions between material properties and external factors. For instance, recent advancements in ML include hybrid optimization techniques such as the artificial hummingbird algorithm, which has been applied to boosted tree models to achieve enhanced accuracy and efficiency in |E| prediction [16]. Similarly, the application of artificial neural networks (ANNs) tailored for specific contexts, such as Colombian asphalt mixtures, has demonstrated the adaptability of DL methods to regional material characteristics and conditions [17].

Moreover, preprocessing and postprocessing strategies for refining input features have been introduced, offering significant improvements in model performance. For example, preprocessing methods that optimize feature extraction and noise reduction have proven particularly valuable for dynamic modulus datasets, as highlighted in recent work by researchers developing |E| data refinement protocols [18]. These advancements underscore the growing potential of ML and DL in overcoming the constraints of traditional regression models, paving the way for more accurate and scalable |E| prediction frameworks. Table 1 summarizes the ML- and DL-based models developed in recent studies.

Table 1. Previous |E*| ML predictive models from the literature

Predictive Model	References
Regression models	Zhang et al. [9], Khattab et al. [11], Al-Tawalbeh et al. [15], Sakhaeifar et al. [19], Singh et al. [20], and Chen et al. [21]
Decision trees, random forest, M5P tree models	Behnood & Daneshvar [22] and Daneshvar & Behnood [23]
Ensemble trees (ET)	Barugahare et al. [10] and Awed et al. [24]
Support vector regression (SVR)	Liu et al. [25] and Hu & Solanki. [26]
Gaussian process regression (GPR)	Uwanuakwa et al. [27]
Artificial neural networks (ANN)	El-Badawy et al. [14], Ceylan et al. [28-30], Gong et al. [31], Ghasemi et al. [32], Rezazadeh Eidgahee et al. [33], Zhang et al. [34], Barugahareet al. [35], Mohammadi Golafshani et al. [36]
DL: convolution neural networks (CNN)	Moussa & Owais [37]
DL: deep residual neural networks (RNN)	Moussa & Owais [38]
Comparison: SVR, kernel ridge regression (KRR), ANN, GPR, gradient boosting (GB), and eXtreme gradient boosting (XGBoost)	Liu et al. [39]

While considerable advancements have been made in predicting the |E| of HMA using traditional regression-based and emerging ML models, several critical gaps remain. First, most existing studies focus on a limited range of either classical ML or DL models, often neglecting to compare their predictive performance across diverse climatic conditions systematically. Second, the trade-offs between model accuracy, computational complexity, and practical applicability are rarely addressed, leaving practitioners without clear guidance on the most suitable methods for different scenarios. Third, there is a limited exploration of integrating advanced data preprocessing techniques and hybrid optimization algorithms, such as ensemble-based methods, to enhance the performance of ML/DL models.

To address these gaps, this study aims to provide a comprehensive benchmarking of classical (24 algorithms) and deep ML models (9 algorithms) for |E| prediction across three distinct datasets and climatic conditions. This study seeks to bridge the gap between theoretical advancements and practical applications in pavement engineering by evaluating a broader range of models and incorporating advanced data-driven techniques. These datasets were based on the knowledge gained from the Witczak NCHRP 1-37A η-based, Witczak NCHRP 1-40D G, δ-based, and Hirsch G*-based models. Unlike previous research, our study evaluates each model not only based on prediction accuracy but also considers the computational complexity and practical applicability of each algorithm, providing an understanding of the trade-offs involved. By doing so, the ultimate goal was to provide practitioners in the field of pavement engineering with enhanced tools for predicting |E*| with higher accuracy. To achieve this aim, the following specific objectives were defined:

- To compile and aggregate three distinct datasets, comprising 3,720 |E| measurements, from diverse climatic conditions in Saudi Arabia (hot climate) and Idaho (cold climate). These datasets were derived from established predictive models, including Witczak 1-37A (η-based), Witczak 1-40D (G, δ-based), and Hirsch (G*-based).
- To implement and benchmark a comprehensive range of predictive models, including 24 classical ML algorithms (e.g., linear regression, support vector regression, ensemble methods) and 9 DL architectures (e.g., convolutional and recurrent neural networks).
- To fine-tune the developed models through a meticulous optimization process involving their hyperparameter adjustments and structural refinements to improve their prediction performance.
- To systematically compare models based on multiple performance metrics, including prediction accuracy, computational complexity, and practical applicability, to provide practitioners with actionable insights into the trade-offs involved.

This study stands as an exceptional comparative analysis of regressors in predicting |E*| using the same dataset and optimizing the performance of each regressor individually to achieve optimal performance.

2. Methodology Framework

The methodology framework of this research consists of four steps, as demonstrated in Figure 1. In the first step, the |E*| database comprising 3,720 response features was retrieved from KSA and Idaho state Superpave AC mixtures. The collected datasets for 13 input features were thoroughly cleansed and pre-processed for the subsequent steps.

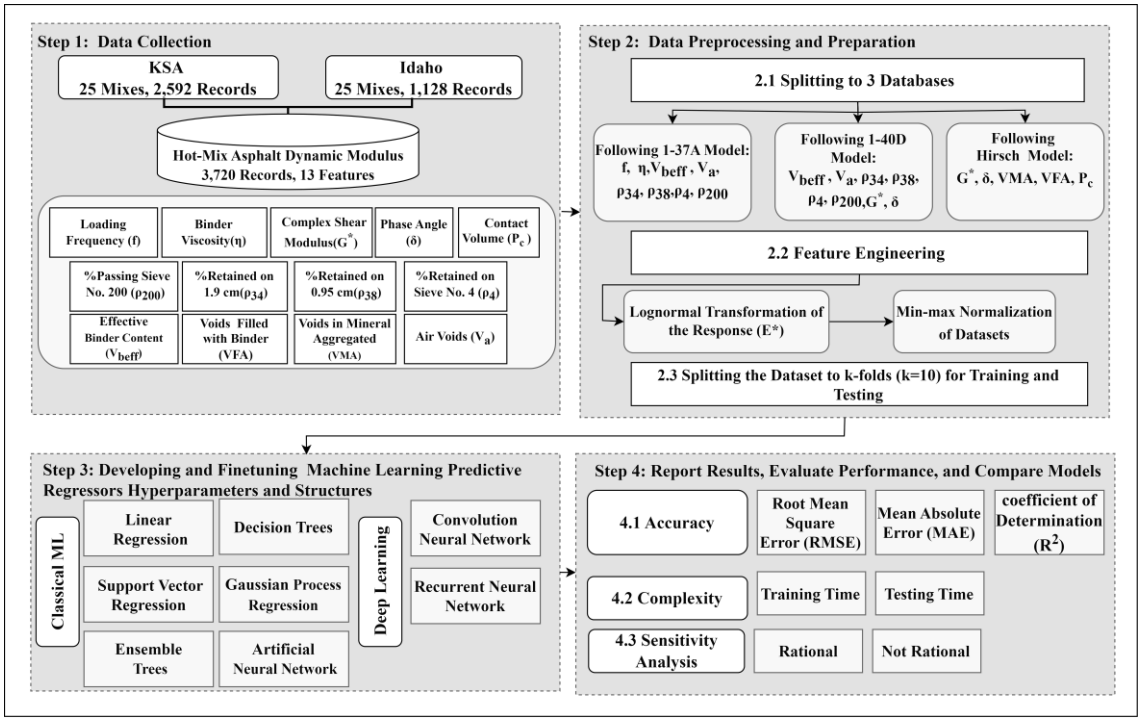


Figure 1. Methodology Framework

In the second step, the |E*| database was recategorized into three distinct databases based on the predictive models: Witczak NCHRP 1-37A η-based, Witczak NCHRP 1-40D G*, δ-based, and Hirsch G*-based. As data preprocessing and preparation for the modeling step, the following stages were implemented: a lognormal transformation of the response variable (|E*|), min-max normalization scaling of the entire datasets (input and response features), and lastly, dividing each dataset into ten folds, as recommended for large databases, for ML and DL models training and testing to eliminate bias in reporting the modeling performance.

In the third step, various classical ML and DL regressors were developed to predict the base sections over three different runs. The parameters of each regressor were fine-tuned based on prediction accuracy. The developed classical ML regressors included multiple linear regression (MLR), decision trees (DT), SVR, ET, GPR, and ANN. Moreover, different CNN and RNN DL structures were developed.

Lastly, the performance of the developed algorithms across the different databases was evaluated and compared using a multi-stage assessment framework encompassing accuracy, complexity, and rationality measurements of effectiveness. The results obtained were further compared to those of previously developed regression and ANN models. Accordingly, comprehensive conclusions were derived from the achieved results to address the primary research questions outlined in the research gap and objectives section.

2.1. Data Description and Preprocessing

To conduct the intended investigation and achieve the main objectives of the study, 3720 |E*| measurements were retrieved from 50 Superpave AC mixtures from KSA and Idaho records. These records contain a diverse range of aggregate gradations and binder performance grades that cover different climatic regions in both KSA and Idaho. Moreover, the dataset comprises a total of 13 continuous features, detailed in Table 2. These features cover different binder inputs' levels (i.e., cases) based on the pavement mechanistic-empirical (ME) design as follows: level 1a of conventional binder data, level 1b of Superpave performance grade binder, and level 3 of default binder data. For more details regarding the characteristics of the AC mixtures, please refer to [11, 14].

Table 2. Description of Database Features

Factor	Unit	NCHRP 1-37A	NCHRP 1-40D	Hirsch	Min.	Max.	Avg.	Std.
Loading Frequency **(f)**	Hz	✓	-	-	0.02	3.98	1.10	1.40
Binder Viscosity **(η)**	cP	✓	-	-	3.7E+05	2.7E+12	5.9E+11	1.1E+12
Effective Binder Content, by volume **(V_{beff})**	%	✓	✓	-	8.09	15.97	10.45	1.15
Air Voids in the Mix **(V_a)**	%	✓	✓	-	0.98	9.61	5.49	2.37
Cumulative Retained Weight on 1.9 cm **(ρ_{34})**	%	✓	✓	-	0.00	23.00	3.68	6.76
Cumulative Retained Weight on 0.95 cm **(ρ_{38})**	%	✓	✓	-	10.50	58.20	24.47	12.52
Cumulative Retained Weight on Sieve No. 4 **(ρ_4)**	%	✓	✓	-	37.40	66.70	49.39	8.53
Amount Passing a Sieve No. 200 **(ρ_{200})**	%	✓	✓	-	3.28	8.20	5.07	1.04
Complex Shear Modulus **(G*)**	psi	-	✓	✓	0.06	18164.1	2051.63	3630.9
Phase Angle **(δ)**	°	-	✓	✓	3.44	87.23	50.77	20.91
Voids in Mineral Aggregated **(VMA)**	%	-	-	✓	11.95	23.53	15.95	2.44
Voids in Mineral Aggregated Filled with Asphalt Binder **(VFA)**	%	-	-	✓	49.56	92.04	66.83	11.47
Contact Volume **(P_c)**	%	-	-	✓	0.01	0.73	0.22	0.20
Dynamic Modulus **(|E*|)**	Psi	Response			2.16 E+03	5.09 E+06	1.15 E+06	1.23 E+06
	MPa				1.49 E+01	3.51 E+04	7.93 E+03	8.49 E+03

The preprocessing of the datasets included the following:

(1) The database was categorized into three datasets based on the well-known |E*| predictive models:

The Witczak NCHRP 1-37A η-based [14]:

$$log_{10} E^* = -1.249937 + 0.02923(\rho_{200}) - 0.00176(\rho_{200})^2 - 0.002841\rho_4 - 0.058097V_a - 0.802208\frac{V_{beff}}{V_{beff}+V_a} + \left\{\frac{[3.871977-0.0021\rho_4+0.003958\rho_{38}-0.000017\rho_{38}^2+0.00547\rho_{34}]}{[1+e^{(-0.603313-0.313351\,log\,f-0.393532log\eta)}]}\right\} \quad (1)$$

Witczak NCHRP 1-40D G*, δ- based [14]:

$$\log_{10}E* = 0.02 + 0.758\,(|G_b^*|^{-0.0009}) \times \left(6.8232 - 0.03274\rho_{200} + 0.00431\rho_{200}^2 + 0.0104\rho_4 - 0.00012\rho_4^2 + 0.00678\rho_{38} - 0.00016\rho_{38}^2 - 0.0796V_a - 1.1689\left(\frac{V_{beff}}{V_a+V_{beff}}\right)\right) + \frac{1.437+0.03313V_a+0.6926\left(\frac{V_{beff}}{V_a+V_{beff}}\right)+0.00891\rho_{38}-0.00007\rho_{38}^2-0.0081\rho_{34}}{1+e^{\left(-4.5868-0.8176log\left|G_b^*\right|+3.2738log\delta\right)}} \quad (2)$$

Hirsch G*-based Model [14]:

$$E^* = P_c\left[4{,}2000{,}000\left(1-\frac{VMA}{100}\right)+3|G^*|\left(\frac{VFA\times VMA}{10{,}000}\right)\right]+(1-P_c)\times\left[\frac{1-\frac{VMA}{100}}{4{,}2000{,}000}+\frac{VMA}{3\times VFA\times|G^*|}\right]^{-1} \quad (3)$$

(2) The lognormal transformation was applied to the response variable to deal with the dataset imbalance and skewness, thus mitigating the impact of data skewness on ML and DL predictions and enhancing their overall performance. It has been widely employed for fitting originally continuous and positively skewed data distributions into a normal distribution to meet the normality assumption [40-42]. Such transformation is easy to perform, requires minimal expertise, and is computed using the following equation [40]:

$$|E^*|_t = \log_{10}(|E^*|) \quad (4)$$

where $|E^*|_t$ is the logarithmically transformed dynamic modulus.

(3) Min-max normalization was used for the three proposed datasets to address data variability since features encompass a wide variety of ranges. This method is widely recognized for achieving superior performance compared to other scaling techniques [43, 44]. This technique rescales input/output features from their original range to a new range of values, thereby converting all features to a fixed range while maintaining the original interval. Usually, the new scale lies between 0 to 1. The transformation is often achieved using a linear interpretation formula and calculated as presented in the following equation [45]:

$$\text{min-max}(x_{if}) = \frac{x_{if} - \min(f)}{\max(f)-\min(f)} \quad (5)$$

where x_{if} is the value of the feature that is considered for normalization at observation i, $\text{min-max}(x_{if})$ is the normalized value of the feature at observation i, and min(f) and max(f) are the minimum and maximum values of the feature, respectively.

(4) Lastly, prior to fitting the regressors to the |E*| dataset, the whole dataset was split 70% for training and underwent k-folds (10-folds) cross-validation, and the remaining 30% unseen datasets were kept aside for testing. This was performed to mitigate biases in testing results and compare unbiased results fairly between the proposed algorithms. Consequently, the issue of overfitting was avoided, and the generalization capability of the proposed predictive models was enhanced. The dataset was partitioned into k-subsets for training purposes; each subset was reserved while training the model over the remaining subsets in a loop that covered all subsets.

2.2. ML and DL Predictive Models

Machine learning modeling has revolutionized pavement engineering, propelling it into a new era of problem-solving and innovation by harnessing the power of artificial intelligence [46-50]. In the third step, multiple ML regressors were developed, and their hyperparameters and structures were fine-tuned based on prediction accuracy and complexity. This study encompassed six classical ML algorithms and two DL algorithms with varying structures and hyperparameters to compare the previously developed regression and ANN models comprehensively. The chosen methods include MLR, DT, SVR, bagging and boosting ET, GPR, feed-forward multilayer perceptron ANN, CNN, as well as RNN. For model development, training and testing were conducted using the Scikit-learn ML library, TensorFlow, Keras, and PyTorch DL libraries integrated into the Python programming language within the Jupyter Notebook platform. Additionally, each model's parameters were fine-tuned to select the optimal configuration for each tested algorithm. The following section provides the theoretical background for each regressor and details the parameters tested for each algorithm.

2.2.1. Classical ML Models

(1) *MLR:* is considered as the process of fitting models to data under the assumption that the relationship between the feature values and the target values is linear. MLR is the oldest and the most widely used approach among ML-based predictive models [51]. It assesses the strength of the relationship between the target and a series of changing variables [52]. In the MLR algorithm, the target is assumed to be a dependent variable, and the features are the variables' dataset matrix. Different MLR structures were assessed, including the simple linear, interactions linear, robust linear, and stepwise linear regression [53].

(2) *DTs:* are statistical supervised classical ML algorithms that utilize conditional probabilities to look for the relationship between features [54]. DTs are common due to their intelligibility and implementation ease [55]. DT comprises a hierarchy of nodes in a tree configuration. The most informative feature was utilized to partition the input data recursively, growing it into new branches. Then, the nodes in each new branch split again based on pre-specified criteria. The splitting proceeds till meeting stopping criteria or reaching a terminal node where the data is ideally pure [56]. Once the tree is constructed, it could be used for prediction by following the path from the root to the leaf node. To fine-tune the DT model, the commonly considered minimum leaf size parameter was varied to compare DT performance under different levels of regularization representing the minimum number of samples required in a leaf node. The minimum leaf size was set to 4, 12, and 36, representing fine, medium, and coarse regression trees.

(3) *SVR: is a common supervised ML algorithm that has* proved its efficiency in multiple pavement performance and condition prediction with low generalization errors and results interpretation [57]. Through modeling SVR, a hyperplane was constructed in a high-dimensional feature space to seek the best approximation of the relationship between input and target features. The best approximation is found by minimizing the distance between the hyperplane and the input training datapoints [58]. To do so, a kernel was used to transform the input features space, which consists of a subset of the training dataset, known as support vectors. In this research, six kernel functions were developed and investigated, including linear (1^{st}-degree linear polynomial), quadratic (2^{nd}-degree quadratic polynomial), and cubic polynomials (3^{rd}-degree cubic polynomial), fine (Radial Basis Function "RBF" SVR, Kernel scale of 0.71), medium (RBF SVR, Kernel scale of 2.8), and coarse (RBF SVR, Kernel scale of11) Gaussian kernels.

(4) *ET:* forms a combination of multiple weaker regressors in an ensemble, providing more accurate predictions [59]. It proved its efficiency in pavement performance prediction applications [60]. There are two main types of ETs, namely bagging and boosting. The main difference between them is the way the regressors are ensembled. In bagging, multiple models are created at the same time and then combined with replacement [25]. In boosting, data is partitioned into multiple subsets of the original data that was used to develop the model, then the performance is boosted by combining them through a specific cost function, and models are generated in a sequential manner [61]. In this research, bagging (random forests) and boosting ET, including gradient boosting (GBM), extreme (XGBoost), and light (LightGBM), were trained. The main difference is that GBM is an ET algorithm that builds an additive model by sequentially training a series of weak learners (typically DT) by utilizing gradient descent optimization to minimize the loss function. The XGBoost is a highly efficient and scalable algorithm that incorporates L1 and L2 regularization to control overfitting [62, 63].

Meanwhile, the LightGBM is another GBM algorithm that is designed for efficiency and speed. It uses leaf-wise growth, which can be faster but may lead to overfitting if not controlled [64]. Through modeling, a grid search was conducted over multiple hyperparameters for the ETs, including (1) The number of estimators from 0 to 500 with an increment of 100; a higher number of trees can enhance the model robustness but may also affect the training time, (2) The maximum depth (3, 5, 7, 9, 11, 15), greater the depth can capture more complex patterns but may also lead to overfitting, (3) The learning rate (0.01, 0.05, 0.1, 0.2), indicating the step size of each iteration, and (4) The number of leaves (31, 63), controlling number of leaf nodes in each decision tree, larger values allow more complex trees. The best parameters were found to be the number of estimators of 30, a learning rate of 0.1, the maximum depth of the individual trees of 3, and the maximum number of leaves per tree of 31.

(5) *GPR:* GPR is a common supervised learning ML algorithm based on probabilistic nonparametric learning in which the output target is normally distributed [65]. It has proved its prediction efficiency in multiple successful pavement engineering applications [66]. It models the relationship between the input features and target using a GPR as a multivariate Gaussian distribution before the mean function parametrizes it. It is efficient for nonlinear data using kernel functions. Through the analysis, multiple kernels were adopted, including rational quadratic (with a length scale of 1), squared exponential (RBF with a length scale of 1), Matern 5/2 (with a length scale of 1), and exponential (with a length scale of 2).

(6) *ANN:* a multilayer perceptron (MLP), is a common supervised learning ML algorithm. ANN was sufficiently used in various pavement engineering applications [67]. It consists of multiple layers of neurons. Each layer is composed of a set of nodes that are fully connected with the previous layer. The weights of the connections between the layers are trained and updated using a training algorithm. The activation function is deployed in the neuron arrangement to vary the input features and their impact on the target using the weight inputs. In this research study, one input layer was used containing eight nodes for the 1-30A and 1-40D datasets and five for the Hirsch dataset (based on the number of input features), one and two hidden layers were attempted, and the number of neurons was varied from 10 to 50 with an increment of 5, the solver (optimizer) was set to Adam, and three activation functions were attempted including Logistic (Sigmoid), Tanh, Relu to fine-tune the MLP results and one output layer that produces the final target layer (the HMA $|E^*|$).

2.2.2. DL Models

(1) *CNN:* One of the most utilized types of distinctive DL architecture is the CNN [68]. CNNs are designed for processing grid-structured data, like images, videos, and sequential data. In CNN, each module contains a sequence of convolutional layers, pooling layers, densifying (fully connected layers), activation layers, dropout, and batch normalization layers. Generally, the modules (layers) are stacked one on top of the other or with a deep neural network on top to form a deep model [69]. CNN can be grouped into different architectures, each with a unique structure and variations in the number and arrangement of layers. In this research, the most widely used and common CNN structures were tested, and their structures were optimized, including LeNet-5, AlexNet, VGGNet, GoogLeNet (Inception), Residual Network (ResNet), ResNeXt, Dense networks (DenseNet), and EfficientNet, as listed in Table 3.

Table 3. CNN Structures

Structure	Structure (Layers) Description
Deep CNN	Model Initialization, Convolutional Layer (filters=512, kernel size=3, activation= Relu), Max-Pooling Layer (pool size=2), Convolutional and two Max-Pooling Layers (filters =256, pool size=1), Global Average Pooling Layer, Dense Layers (neurons= 512, 256, 125, activation= ReLU, linear), Output Layer.
LeNet-5 [70, 71]	Model Initialization, Convolutional Layer (filters=512, kernel size=3, activation= ReLU), Max-Pooling Layer (pool size=2), Convolutional and Max-Pooling Layers (filters =256, pool size=1), Flatten Layer, Dense Layers (number of neurons= 120, activation= ReLU, linear), Output Layer.
AlexNet [72]	Input Layer, Convolutional Layer (filters=96, kernel size=3, strides=4, activation= ReLU), Max-Pooling Layer (pool size=2, strides=2), Additional Convolutional Layers (filters =256, 384), Additional Max-Pooling Layers (pool size=1, strides=2), Flatten Layer, Dense Layers (neurons= 120, activation function= ReLU, linear), Dropout Layers (rate=0.5), Output Layer.
VGGNet [73]	Input Layer, Convolutional Layer (filters=128, kernel size=3, activation= ReLU), Max-Pooling Layer (pool size=2, strides=2), Flatten Layer, Dense Layers (neurons= 256, 128, activation function= ReLU, linear), Dropout Layers (rate=0.5), Output Layer.
GoogLeNet (Inception) [74]	Input layer, convolutional layer (filters= 32, kernel size= 3, activation= Relu), Two Inception modules (filters=64), max-pooling and dropout layers, Two more Inception modules (filters=128), Max-pooling and dropout layers, Flatten Layer, Two fully connected layers (neurons=256, activation= ReLU, Dropout), Output layer.
ResNet [75]	Input layer, convolutional layer (filters=128, kernel size= 3, padding), Two residual blocks (filters=128, kernel size= 3), Flatten Layer, fully connected layer (neurons=128, activation= ReLU, L2 regularization), Dropout Layers (rate=0.5), Output layer.
ResNeXt [76]	Input layer, 1D convolutional layer (filters=64, a kernel size= 7, strides=2), ReLU activation, Max-pooling (pool size= 3, strides= 2). Three residual blocks are stacked: The first block (filters=64, kernel size=3, cardinality= 8), the second block (filters=128, kernel size=3, cardinality= 8), and the third block (filters=256, kernel size=3, cardinality= 8), global average pooling after each block, fully connected layer (neurons=256, activation= ReLU), Dropout Layer (rate=0.5), Output layer.
DenseNet [77]	Input layer, convolutional layer (filters=512, kernel size= 3, activation= ReLU), Three dense blocks, and two transition blocks are stacked: Dense Blocks: two convolutional layers (filters=128, kernel size=3, activation= ReLU, and dropout rate=0.2), Transition Blocks: convolutional layer (filters=256, activation= ReLU, Max-pooling pool size= 2, dropout rate = 0.2), Flatten layer, fully connected layer (neurons=256, activation= ReLU), Dropout Layers (rate=0.5), Output layer.
EfficientNet [78]	Input layer, convolutional layer (filters=32, kernel size= 3, activation= ReLU), convolutional layer (filters=64, kernel size= 3, activation= ReLU), Max-pooling (pool size= 2), Two consecutive blocks of convolutional layers: two convolutional layers (filters=128, kernel size= 3, and activation= ReLU), three convolutional layers (filters=256, kernel size= 3, and activation = ReLU), max-pooling (pool size= 2), three convolutional layers (filters=512, a kernel size = 3, activation= ReLU), Global average pooling layer, fully connected layer (dense, neurons= 512, activation= ReLU), Dropout Layers (rate=0.5), Output layer.

(2) *RNN:* RNN is another class of DL designed for processing sequential data. RNNs have an internal memory that allows them to maintain information about past observations and use it to make predictions or decisions at each time step [79]. Thus, they are well-suited for tasks where the order and context of data points matter, such as natural language processing, time series analysis, and more. Like CNN, RNN consists of a sequence of layers grouped for a specific purpose. These modules (layers) are stacked one on top of the other or with a deep neural network on top of it to form an RNN model [69]. RNN can be grouped into different architectures, each with a unique structure and variations in the number and arrangement of layers. In this research, the most widely used and common RNN structures were tested, and their structures were optimized, including Vanilla RNN [80], Long Short-Term Memory (LSTM) [81], Gated Recurrent Unit (GRU) [82], and Deep RNN (Table 4).

Table 4. RNN structures

Structure	Layers
Vanilla RNN	Three SimpleRNN layers are stacked (number of units =128, 64, 32), Dropout Layers (rate = 0.2), Two Dense layers (number of units =16,1, activation = ReLU).
LSTM	Three LSTM layers (number of units = 128, 64, 32), Dropout Layers (rate = 0.2), Two Dense layers (number of units =16,1, activation = ReLU).
GRU	Three GRU layers (number of units =128, 64, 32), Dropout Layers (rate = 0.2), Two Dense (number of units =16, 1, activation = ReLU).
Deep RNN	Four LSTM layers (number of units =128, 64, 32, 16), Dropout Layers (rate=0.2), Dense Layer (one unit).

2.3. Assessment Criteria

During the fourth step of the proposed framework, the performance of the aforementioned ML and DL predictive models was evaluated based on multi-stage assessment criteria, which included different assessments such as models' accuracy, modeling complexity, and predictions' rationality. Prior to the assessment, a reverse lognormal transformation and reverse min-max normalization were implemented to make a fair parallel comparison between the actual and predicted observations on an arithmetic scale. Then, the initial stage of the ML models assessment focused on computing multiple widely used accuracy performance measurements of effectiveness (MOE), including the root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2) to judge the models' performance as follows [83]:

$$\text{MSE}= \frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2 \quad (6)$$

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|\hat{y}_t - y_t| \quad (7)$$

$$R^2 = 1 - \frac{\text{The sum of Squared Residuals (SSR)}}{\text{Total Sum of Squares (TSS)}} = 1 - \frac{\sum_{t=1}(y_t-\hat{y}_t)^2}{\sum_{t=1}(y_t-\bar{y})^2} \quad (8)$$

$$Weighted\ Average = \sum W_i \times x_i^+ + W_i \times (1 - x_i^-) \quad (9)$$

where *n is* the number of records, $\hat{y}_t$ is the predicted response, y_t *is* the measured response, $\bar{y}$ is the average of measured response, W_i is the importance weight assigned for each evaluation criterion, x_i^+ is the normalized criteria that higher values are preferable as the R^2 and the models' prediction's rationality, and x_i^- is the normalized criteria that lower values are preferred as the error rates, time, and other complexity measures. RMSE represents the square root of the average squared differences between predicted and actual values; a lower RMSE indicates better predictive performance. MAE represents the average of the absolute differences between predicted and actual values; a lower MAE signifies better predictive accuracy. R^2 represents the proportion of the variance in the dependent variable that is explained by the independent variables; a higher R^2 value suggests a better fit of the model to the data.

The second stage of the assessment process focused on recording the time required for training the different ML and DL models as an indication of the modeling complexity. The reported time is in minutes. Based on the reported prediction accuracy and complexity performance measures, a thorough comparison was conducted between the developed model and the ones reporting the best performance, as well as previously developed regression and ANN |E*| predictive models.

In the third stage of the assessment of ML and DL models, after evaluating the accuracy and complexity of the models, a sensitivity analysis of the prediction results was performed to assess the prediction rationality, the impact of the considered features on |E*|, and to predict potential overfitting or memorization of regressors. It is anticipated that a model will demonstrate an outstanding performance in terms of modeling accuracy and outperform other modeling algorithms. However, the sensitivity analysis may reveal unexpected trends, indicating that model failure is attributed to potential overfitting during the modeling process. Thus, sensitivity analysis aids in excluding irrational models. Four features were assessed for each dataset. A rationality score ranging between 0 and 1 was assigned to each model based on its trends, where 0 demonstrated that the model could not follow the expected trend over all four features, and one indicated that the features could capture the accurate trend in all the investigated features.

Lastly, a weighted average was computed for each performance measure to find its overall performance based on the pre-specified multi-objective criteria. An equal weight of 16.67 % was assigned for each criterion since this research focuses on assessing the accuracy and complexity trade-offs as well as modeling rationality.

3. Results and Discussion

3.1. ML Prediction Accuracy and Complexity Performance Results

This section presents a summary and discussion of the accuracy and complexity performance results obtained for all regressors developed in this study for the |E*| predictions. Results were reported for the three datasets of input features (i.e., Witczak NCHRP 1-37A, Witczak NCHRP 1-40D, and Hirsch) for each regressor separately. Then, the regressors with optimized structures and hyperparameters were compared to each other. Figure 2 demonstrates the obtained results for the fine-tuning of the regressors hyperparameters and structures in terms of prediction accuracy (RMSE, MAE, and R^2) and complexity (training and testing times).

3.1.1. Classical ML Models

(1) *MLR:* Four diverse structures of MLR models were attempted, namely simple linear, interactions linear, robust linear, and stepwise linear regression, as illustrated in Figure 2-a. Among these, the interactions linear regression model consistently demonstrated superior predictive accuracy, achieving the highest R^2 values of 0.57, 0.77, and 0.79 for the

1-37A, 1-40D, and Hirsch datasets, respectively. This performance highlights the importance of incorporating interaction terms to capture relationships between input features, particularly in more complex datasets like 1-40D and Hirsch, which include binder-specific parameters such as G* and δ.

While the robust linear regression model accounted for potential outliers in the datasets, its predictive performance was notably lower, with R^2 values ranging from 0.40 to 0.68 across the datasets. This indicates that the robust method's emphasis on minimizing the influence of extreme values may overly simplify dynamic modulus relationships, particularly in datasets with intricate feature interactions. The Hirsch dataset consistently achieved the highest R^2 values across all linear regression models, followed by the 1-40D and 1-37A datasets. This trend reflects the added predictive power of features like G* and δ in the Hirsch dataset, compared to the more volumetric-focused features of 1-37A. The consistent R^2 values around 0.77–0.79 for interactions linear regression on the Hirsch and 1-40D datasets demonstrate that linear models, even with limited complexity, can effectively predict |E*| when datasets include well-structured, physically meaningful features.

In terms of computational complexity, all linear regression models demonstrated exceptional efficiency, with training times between 0.01 and 0.02 minutes and testing times around 0.002 minutes across all datasets. This makes linear regression a viable option for large-scale or time-sensitive projects requiring rapid dynamic modulus predictions.

(2) *DT:* Three distinct DT model configurations were aimed by changing the minimum leaf size parameter, classified explicitly as fine, medium, and coarse DT, as shown in Figure 2-b. Across all datasets, the fine DT model consistently demonstrated the best predictive performance, achieving the highest R^2 values of 0.93, 0.94, and 0.94 for the 1-37A, 1-40D, and Hirsch datasets, respectively. The fine DT's superior accuracy can be attributed to its smaller leaf size (minimum leaf size = 4), which enabled it to capture intricate variations in the input-output relationships. This level of granularity was particularly effective for datasets like 1-40D and Hirsch, where complex binder parameters (e.g., G* and δ) require models that are sensitive to subtle feature interactions.

In terms of computational complexity, the fine DT demonstrated remarkable efficiency, with training times ranging from 0.01 to 0.02 minutes and testing times of approximately 0.001 minutes across all datasets. This computational advantage makes fine DT models highly practical for real-world applications where rapid predictions are essential. While slightly less accurate with R^2 values ranging from 0.88 to 0.92, the medium and coarse DT configurations still performed competitively and maintained comparable computational efficiency, suggesting their potential suitability for simpler prediction tasks or scenarios with less variability in input features. The high and consistent R^2 values across all datasets indicate that DT models effectively captured key input-output relationships. However, the slightly reduced R^2 for coarse DT models suggests a loss of predictive precision due to their larger minimum leaf size, which limits the model's ability to adapt to variations in the dataset.

(3) *SVR:* In the investigation of the SVR models, six distinct configurations were experimented with by varying the kernel parameter (hyperplane), as demonstrated in Figure 2-c. These included linear polynomial, quadratic polynomial, cubic polynomial, fine gaussian, medium gaussian, and coarse gaussian SVR. Among these variations, the coarse Gaussian SVR consistently demonstrated superior performance across all datasets, achieving the highest R^2 values of 0.72, 0.86, and 0.85 for the Witczak NCHRP 1-37A, 1-40D, and Hirsch datasets, respectively. Its kernel scale hyperparameter value of 11 allowed the model to effectively capture nonlinear relationships in the input-output mapping, particularly in feature-rich datasets like 1-40D and Hirsch. The medium Gaussian SVR showed competitive accuracy, with slightly lower R^2 values of 0.66 to 0.82 across the datasets, but demonstrated better generalizability than polynomial kernels, which consistently underperformed with R^2 values below 0.61. Polynomial kernels, particularly cubic, suffered from overfitting tendencies in datasets with high feature variability, as evidenced by their reduced accuracy and increased error metrics.

In terms of computational complexity, coarse Gaussian SVR maintained a balance between performance and efficiency, with training times ranging from 0.23 to 0.44 minutes and testing times between 0.003 and 0.007 minutes. These values are notably efficient, given the nonlinear nature of the kernel function. Fine Gaussian SVR, while marginally faster, exhibited reduced accuracy, indicating that kernel scale optimization plays a crucial role in balancing prediction performance and computational cost. The consistent R^2 values for coarse Gaussian SVR across datasets highlight its adaptability to diverse input features, making it an ideal choice for predicting dynamic modulus in both binder- and volumetric-focused datasets. However, the slightly higher training times compared to linear models suggests its application might be better suited for scenarios where accuracy is prioritized over computational simplicity.

(4) *ET:* Four ET structures were evaluated, each differentiated by data training and assembly approach variations. These encompassed bagging (random forest), GBM, XGBoost, and LightGBM, as represented in Figure 2-d. Among these, the bagging trees model emerged as the top performer across all datasets, achieving the highest R^2 values of 0.94, 0.95, and 0.94 for the Witczak NCHRP 1-37A, 1-40D, and Hirsch datasets, respectively. The model's optimal configuration, with a minimum leaf size of 8 and a learner count of 30 decision trees, allowed it to balance predictive accuracy and computational efficiency effectively.

The boosted tree models, including GBM, XGBoost, and LightGBM, demonstrated slightly lower R^2 values, ranging between 0.87 and 0.90 across the datasets. This marginal drop in accuracy highlights the additional complexity introduced by boosting techniques, which may not have been necessary for datasets where bagging could effectively capture the underlying data relationships. However, these models maintained high performance and demonstrated robustness, particularly in the Hirsch dataset, where ensemble methods had better model intricate binder and aggregate interactions.

In terms of computational complexity, the bagging trees were notably efficient, with training times between 0.33 and 0.44 minutes and a minimal testing duration of 0.01 minutes. While boosted models such as XGBoost and LightGBM required marginally longer training times, their rapid testing times (~0.01 minutes) and scalability make them attractive for real-time prediction scenarios. These models excel in scenarios with large-scale datasets due to their parallelized training capabilities, even though bagging maintained a slight edge in simplicity and interpretability. The consistent performance of bagging trees across datasets and their computational efficiency reinforces their practicality for practitioners seeking robust |E*| predictions with minimal resource requirements

(5) *GPR:* Four distinguished GPR models were evaluated, each tailored by modifying the kernel parameter. These variants included Exponential, Squared Exponential, Matern 5/2, and Rational Quadratic GPR. Among these, the Exponential GPR model demonstrated the most consistent and superior performance across all datasets, achieving an impressive R^2 score of 0.95 across the Witczak NCHRP 1-37A, 1-40D, and Hirsch datasets, as shown in Figure 2-e. This high accuracy underscores the ability of the Exponential kernel to effectively capture the nonlinear relationships present in the |E*| datasets, particularly when dealing with the intricate binder and aggregate interactions emphasized in the 1-40D and Hirsch datasets.

The other kernel configurations, including Squared Exponential, Matern 5/2, and Rational Quadratic, also achieved competitive R^2 values, consistently ranging between 0.94 and 0.95. However, the slightly better performance of the Exponential kernel can be attributed to its flexibility in handling variations in feature distributions and its capacity to generalize well across diverse datasets.

In terms of computational complexity, the Exponential GPR model proved to be the most efficient within the GPR framework. Training times ranged from 138 to 227 minutes, and testing times varied between 0.14 and 0.25 minutes. While these durations are significantly higher than those of other models (e.g., DTs or SVR), they are justified by the high predictive accuracy achieved. This trade-off highlights GPR's suitability for applications where precision is prioritized over computational simplicity. The consistency in R^2 values across all GPR kernels and datasets emphasizes the robustness of this modeling approach. However, the relatively long training times observed for all GPR configurations suggest that the practical deployment of these models may require optimization strategies, such as reducing dataset dimensionality or employing parallelized computing techniques. Overall, the Exponential GPR model emerged as the optimal choice for |E*| predictions when accuracy is critical, particularly for datasets that exhibit nonlinear behaviors.

(6) *ANN:* A total of 54 Artificial Neural Network (ANN) configurations were evaluated by varying the number of hidden layers, neurons per layer, and activation functions (Sigmoid, Tanh, and Relu), as depicted in Figure 2(f). Among these configurations, the ANN with two hidden layers (45 neurons per layer) and a Relu activation function consistently emerged as the most effective across all three datasets. This configuration achieved R^2 values of 0.76, 0.87, and 0.86 for the Witczak NCHRP 1-37A, 1-40D, and Hirsch datasets, respectively. These results highlight the significance of selecting an optimal network depth and activation function for achieving robust predictions of |E*|.

The superior performance of the Relu activation function can be attributed to its ability to mitigate the vanishing gradient problem, particularly in deeper networks. This advantage was further complemented by the use of two hidden layers, which balanced model complexity with the capacity to capture nonlinear relationships in the data. In contrast, models using Sigmoid or Tanh activations demonstrated relatively lower R^2 values, particularly for datasets with high variability (e.g., Hirsch), indicating that these activation functions struggled to capture the intricate feature interactions present in such datasets. While this optimal ANN configuration demonstrated strong predictive accuracy, it required longer computational times compared to classical ML models. Training times for the best ANN configuration ranged from 0.5 to 1.5 minutes, with testing durations consistently around 0.001 minutes across all datasets.

The performance trends across different configurations emphasize the critical role of model architecture tuning in ANN development. While one-layer configurations exhibited faster training times, their reduced R^2 values suggest a lack of capacity to model the nonlinear dynamics of |E*|. Conversely, models with more than 50 neurons per layer showed diminishing returns in predictive accuracy, likely due to overfitting, as evidenced by slightly increased RMSE and MAE values. These findings underscore the importance of balancing network complexity with generalization performance. Overall, the ANN configuration with two hidden layers and Relu activation served as an optimal choice for |E*| prediction, particularly for datasets that require capturing complex binder and aggregate interactions.

3.1.2. DL Models

(1) *CNN:* Nine distinct CNN architectures were evaluated to predict |E*| values, with structural variations explored across Deep CNN, LeNet-5, AlexNet, VGGNet, GoogLeNet (Inception), ResNet, ResNeXt, DenseNet, and EfficientNet, as showcased in Figure 2-g. The Deep CNN architecture consistently demonstrated superior performance across all three datasets. It achieved robust R^2 values of 0.84, 0.90, and 0.90 for the Witczak NCHRP 1-37A, 1-40D, and Hirsch datasets, respectively, highlighting its capacity to model complex nonlinear relationships inherent in |E*| prediction tasks.

The high R^2 values of the Deep CNN reflect its ability to extract detailed feature representations through deeper layer hierarchies, which are particularly advantageous for datasets like 1-40D and Hirsch that incorporate intricate binder and aggregate properties. In comparison, LeNet-5 and AlexNet exhibited lower R^2 values (0.73 to 0.80), indicating their relatively shallow architectures are less effective for modeling the complexity of |E*| datasets.

Training times for the Deep CNN ranged from 28 to 75 minutes, with testing durations spanning 0.03 to 0.35 minutes. While these training times are significantly longer compared to classical ML models, the superior accuracy achieved underscores the suitability of CNN architectures for high-stakes predictive modeling where precision is critical. EfficientNet, although achieving R^2 values comparable to DenseNet (0.86 to 0.88), exhibited marginally higher training times, suggesting that deeper and more complex CNN configurations may not always yield substantial accuracy improvements.

Performance trends also highlight that lightweight architectures such as GoogLeNet and ResNet can achieve competitive results ($R^2 \approx 0.85$ to 0.88) with reduced computational overhead, making them viable alternatives for practitioners seeking a balance between accuracy and efficiency. However, the consistent top performance of the Deep CNN across all datasets demonstrates its adaptability and effectiveness, particularly for scenarios involving detailed feature interactions like those present in the Hirsch dataset. These findings emphasize that while simpler architectures (e.g., AlexNet) may suffice for preliminary |E*| estimations, deeper architectures like Deep CNN are necessary for capturing intricate data patterns, especially in feature-rich datasets. Furthermore, the results indicate that advanced optimization strategies, such as reducing layer redundancy or exploring hybrid CNN frameworks, could further enhance computational efficiency without sacrificing accuracy.

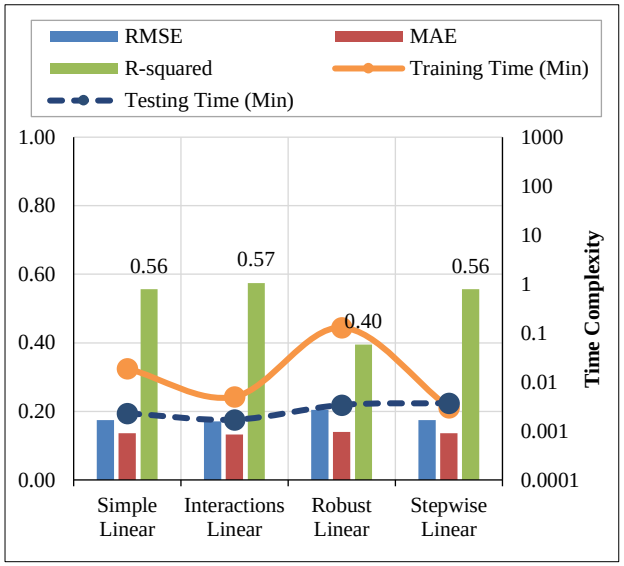
(2) *RNN:* Four Recurrent Neural Network (RNN) architectures, Vanilla RNN, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Deep RNN, were evaluated to predict |E*| values, with variations in their layer structures and configurations, as shown in Figure 2-h. Among these, the GRU RNN consistently emerged as the most effective model across all datasets, achieving R^2 values of 0.78, 0.88, and 0.88 for the Witczak NCHRP 1-37A, 1-40D, and Hirsch datasets, respectively. The high accuracy of GRU RNN highlights its ability to capture temporal and sequential relationships in the data, particularly in feature-rich datasets like 1-40D and Hirsch.

The GRU architecture's superior performance can be attributed to its efficient gating mechanism, which reduces computational complexity compared to LSTM while maintaining the capacity to model long-term dependencies. In contrast, the Vanilla RNN achieved lower R^2 values (0.73–0.85), likely due to its inability to manage gradient vanishing issues, which are critical in modeling complex |E*| relationships. Similarly, while the Deep RNN achieved comparable R^2 values (up to 0.88), its increased complexity resulted in longer training times without significant performance gains over GRU.

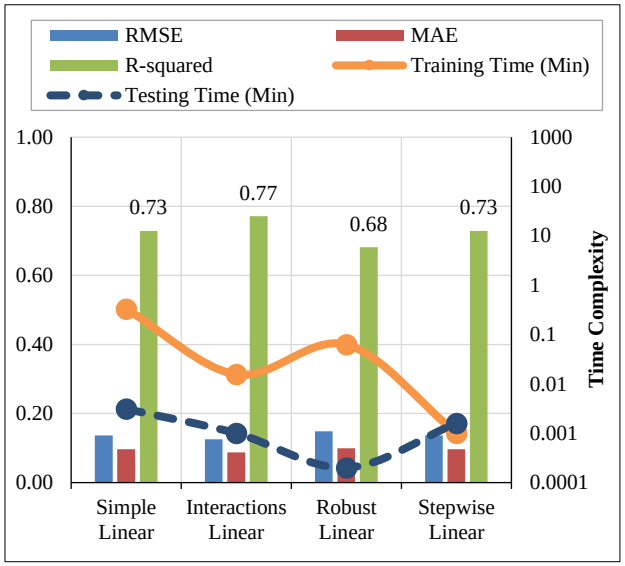
In terms of computational efficiency, GRU demonstrated a balance between accuracy and training/testing times. Training times for GRU ranged from 170 to 370 minutes, while testing durations were between 0.2 and 1.2 minutes. Although these times are longer than those for CNNs and classical ML models, the GRU's consistently high accuracy across all datasets justifies its application in scenarios where precision is paramount. LSTM, while slightly more complex than GRU, achieved similar R^2 values (0.75–0.88) but required marginally longer training times, indicating that GRU may be the more practical choice for |E*| predictions in most use cases. Overall, the GRU RNN stands out as the optimal RNN configuration for |E*| prediction, combining high accuracy with manageable computational requirements

3.2. Actual Measurements versus Best-Performing Models' Predictions

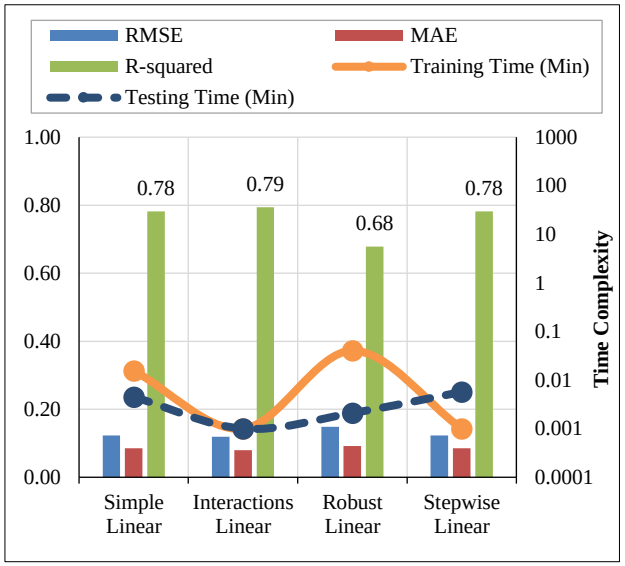
Based on the prediction results visualized in Figure 2, a conclusion was reached regarding the optimal parameters used with each classical ML and DL algorithm and a ranking of all the ML techniques for the suitability of |E*| prediction. An additional investigation was conducted by plotting the actual |E*| measurements versus the predicted values generated by the proposed ML models. The comparisons of predicted results to the measured values (targets) of these finely tuned best-performing models are presented in Figure 3. The illustrated results demonstrate alignment with the conclusions reached in the preceding sections.



1-37A

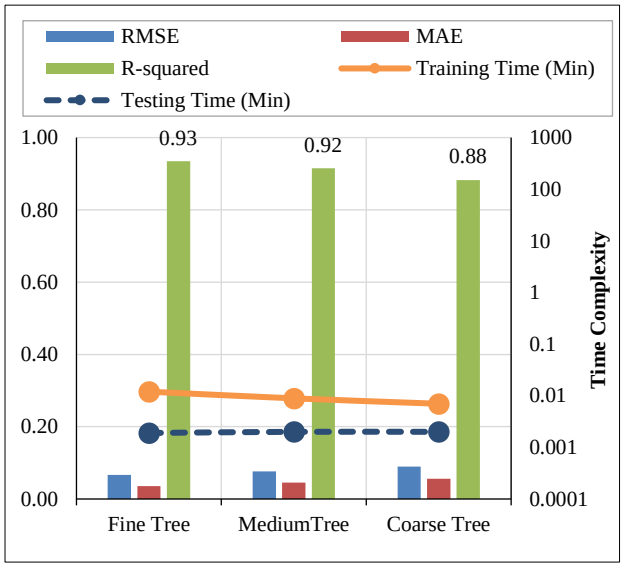


1-40D

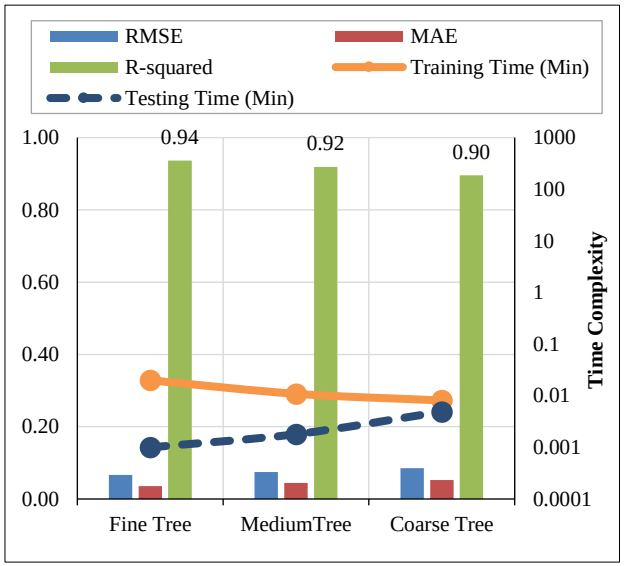


Hirsch

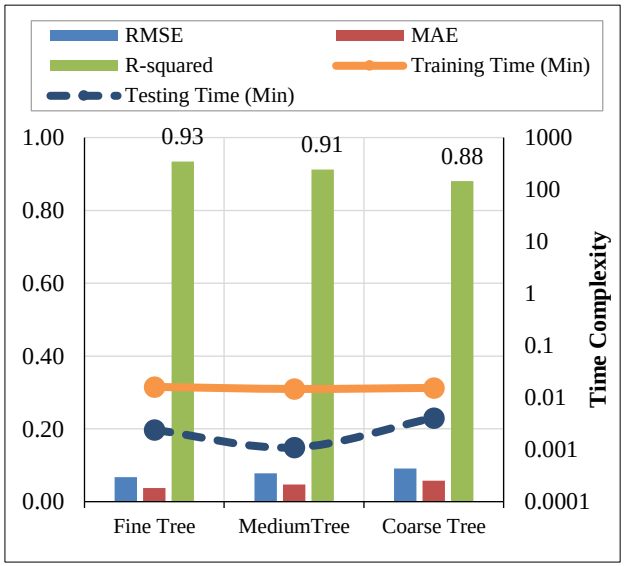
(a) Linear Regression



1-37A

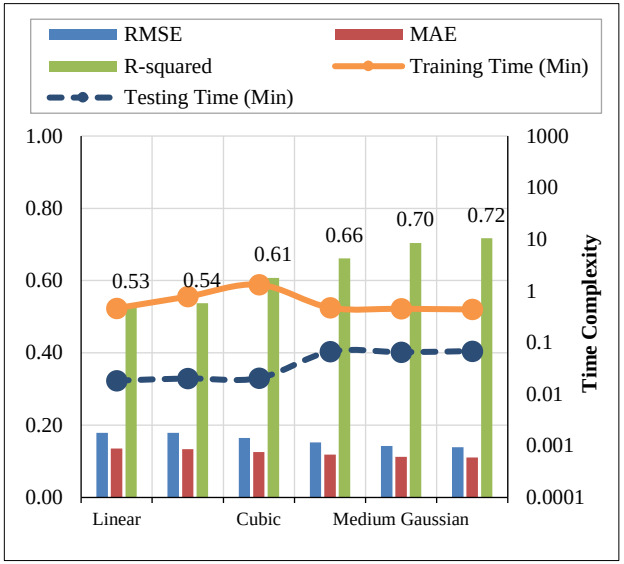


1-40D

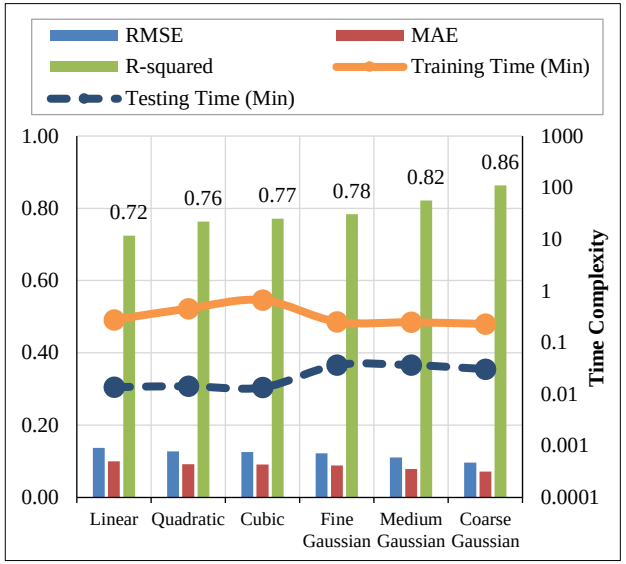


Hirsch

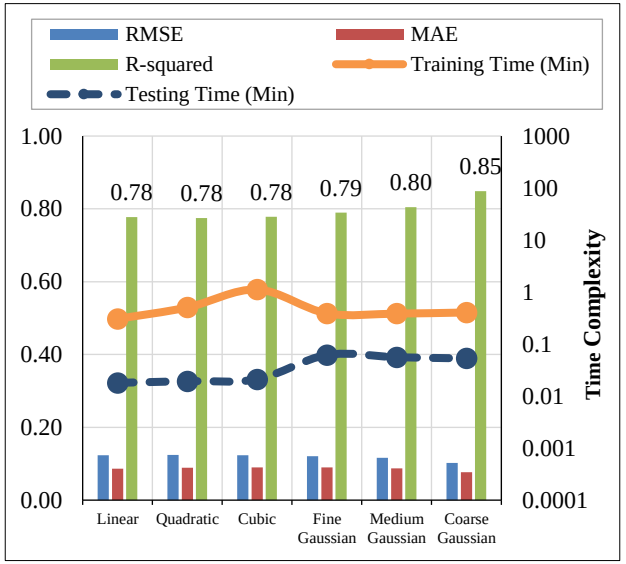
(b) Regression Trees



1-37A

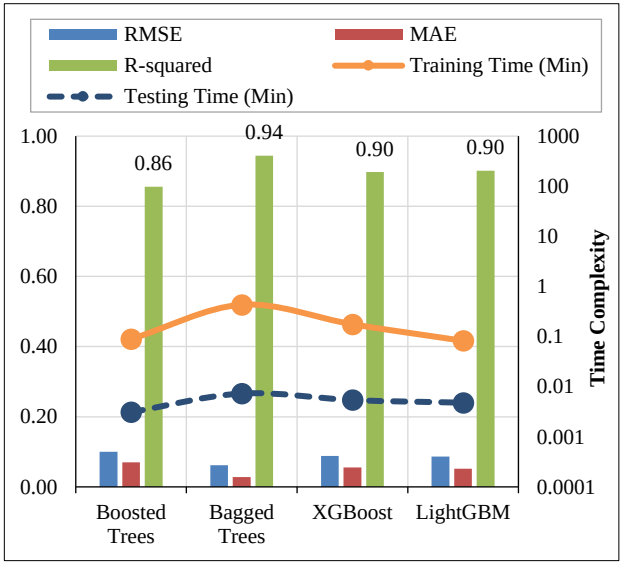


1-40D

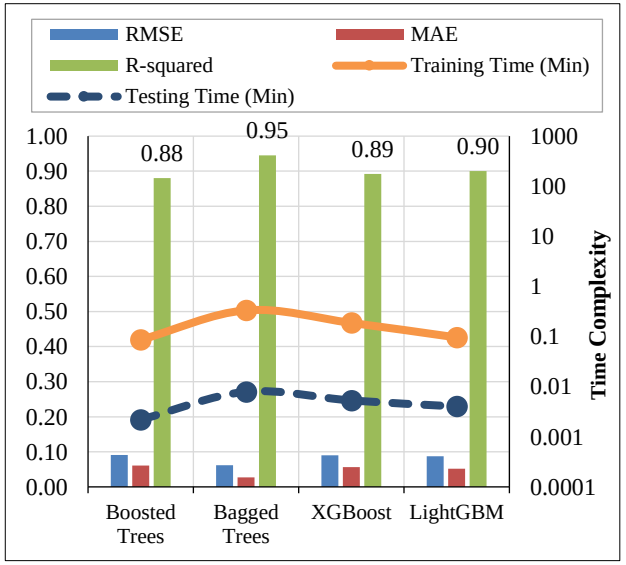


Hirsch

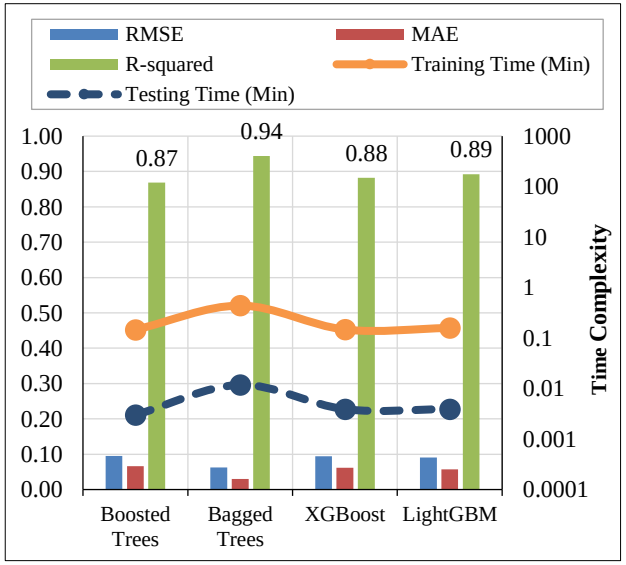
(c) Support Vector Regression



1-37A

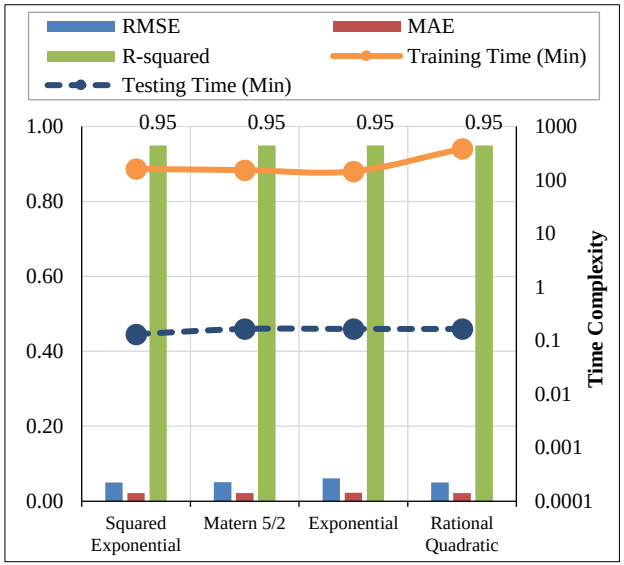


1-40D

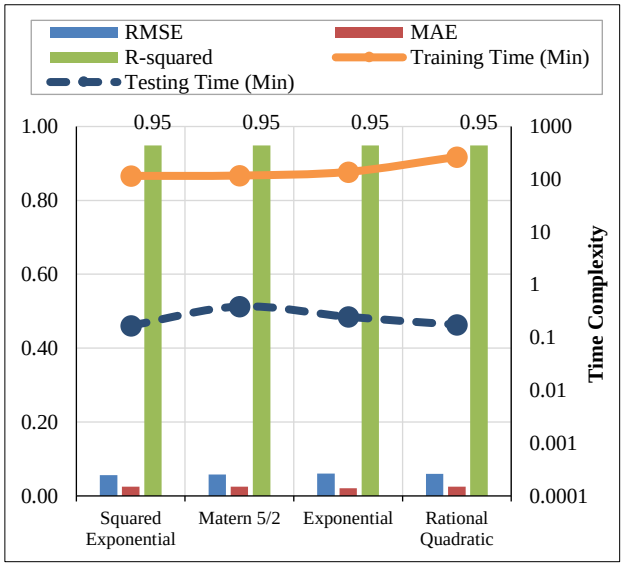


Hirsch

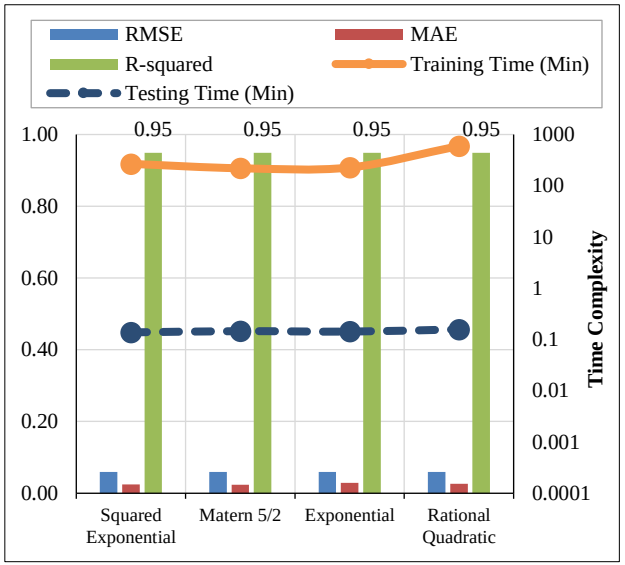
(d) Regression ET



1-37A

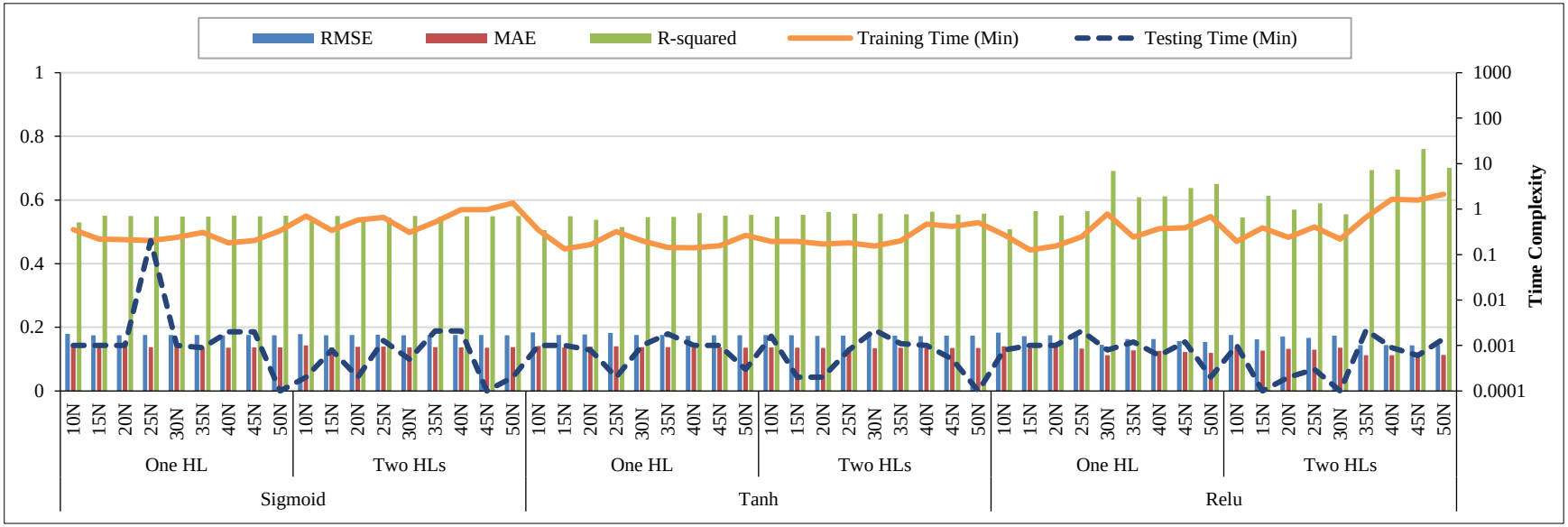


1-40D

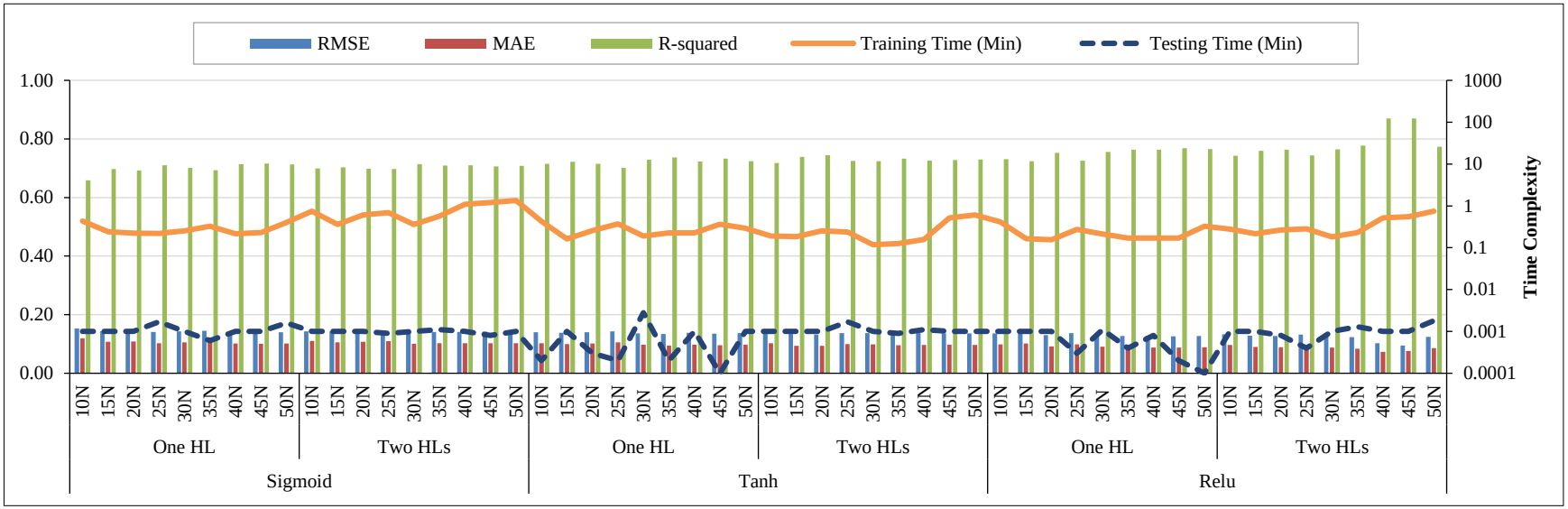


Hirsch

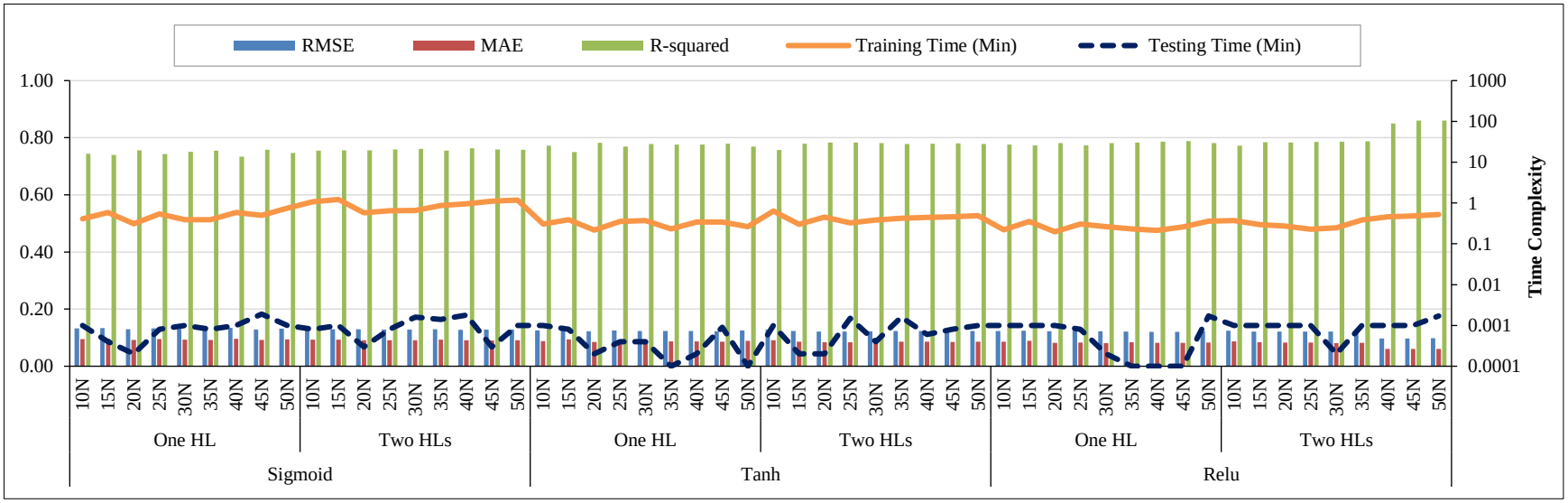
(e) Gaussian Process Regression



1-37A

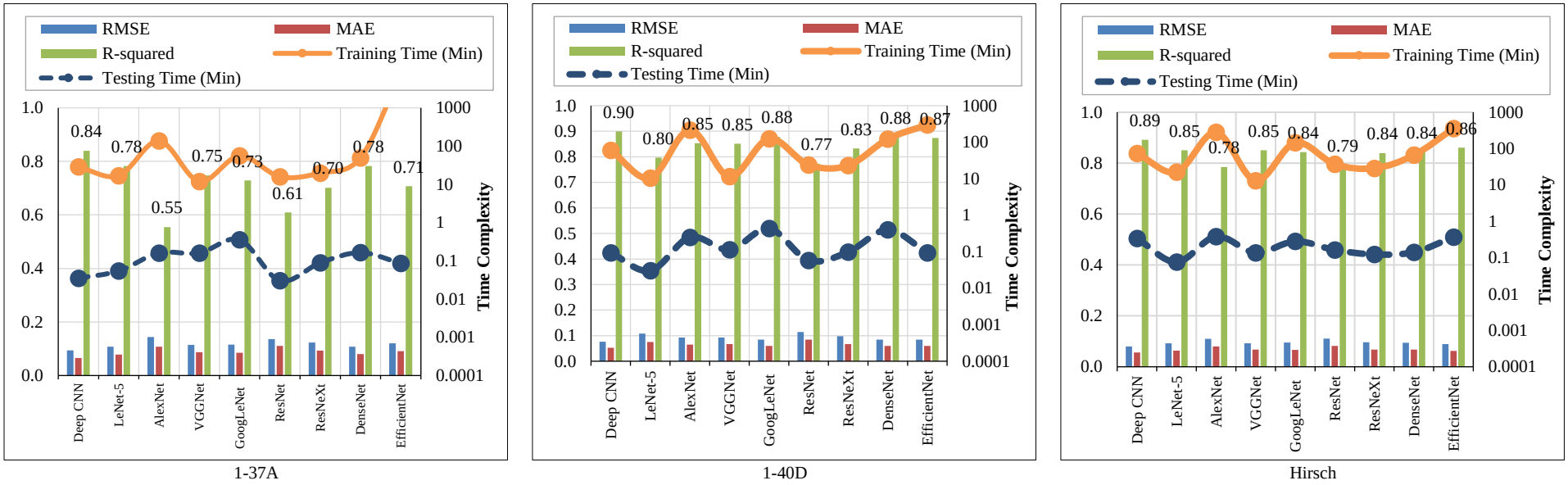


1-40D

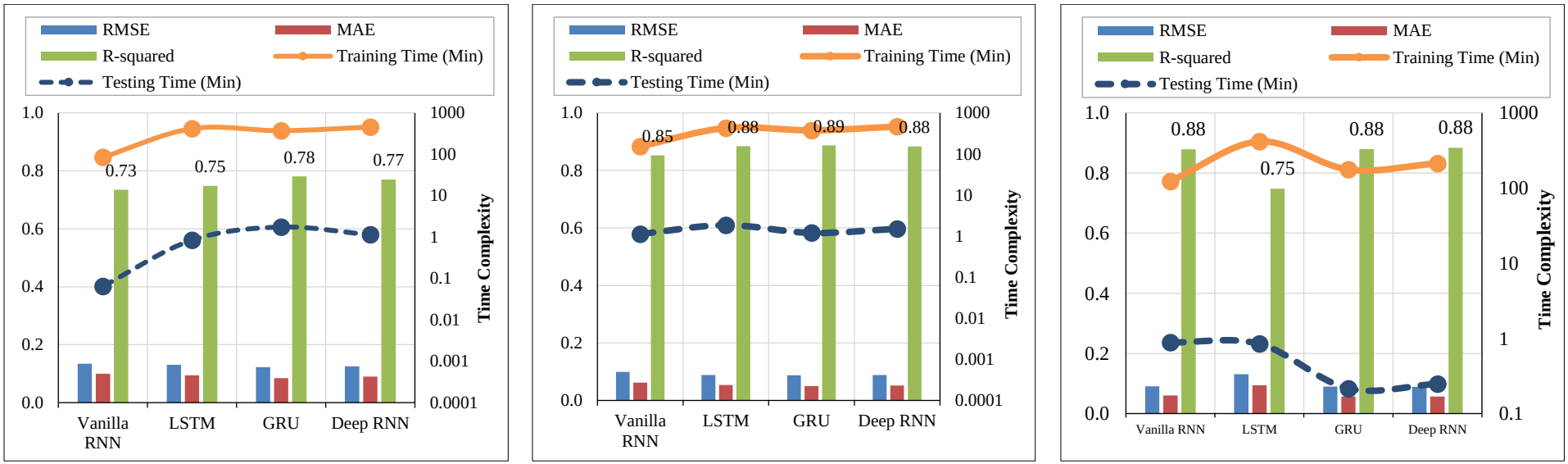


Hirsch

(f) Feedforward MLPANN

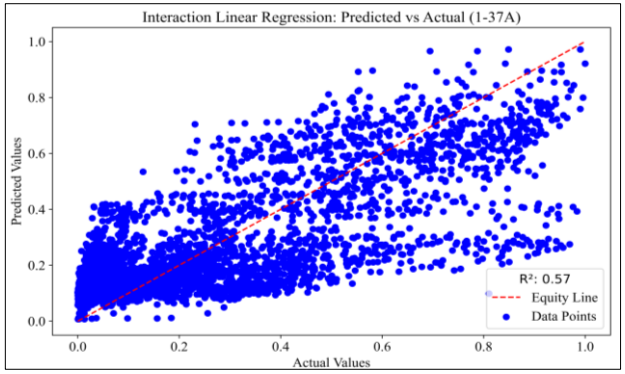


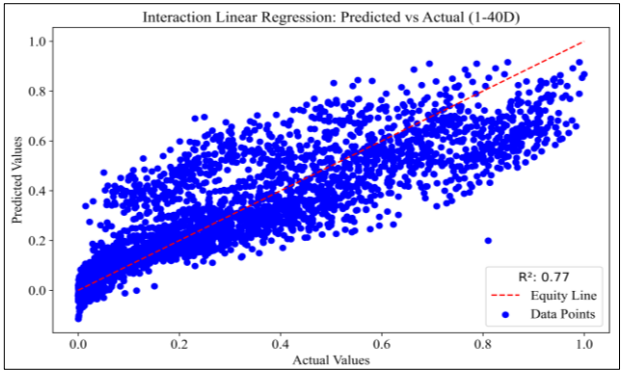
(g) Convolutional Neural Networks

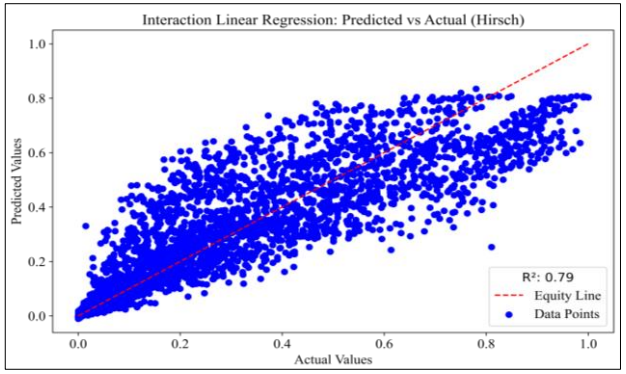


(h) Recurrent Neural Networks

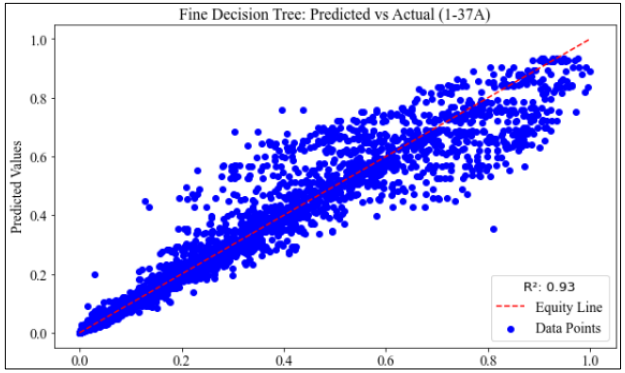
Figure 2. Performance Results of the Proposed Multiple-structured ML and DL Models

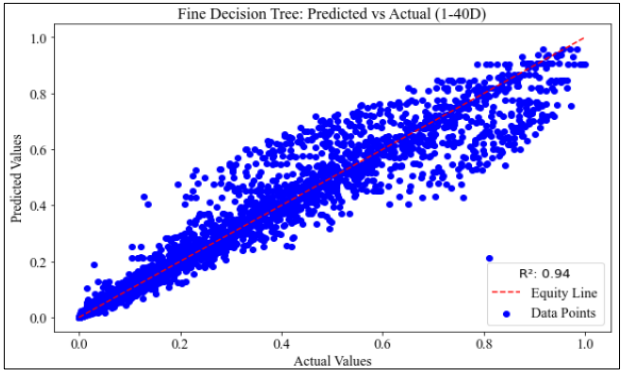


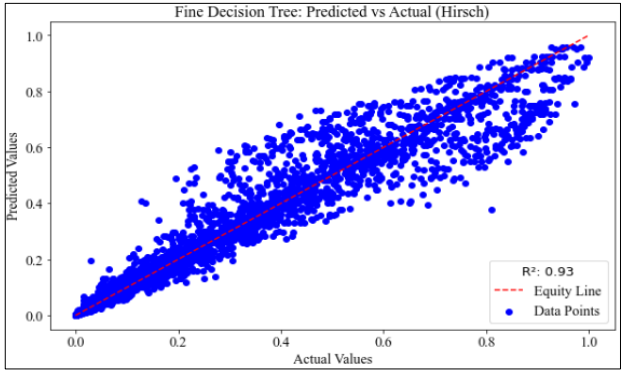




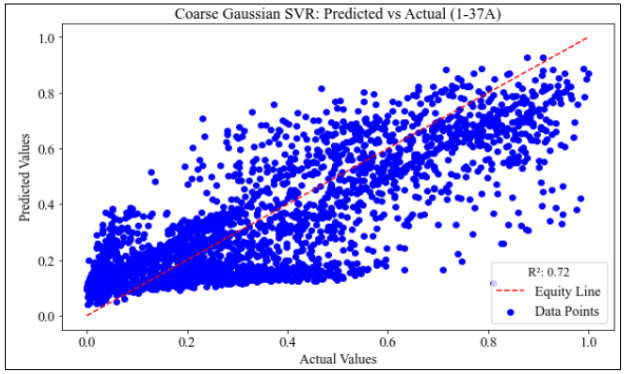
(a) Interactions Linear Regression (MPa)

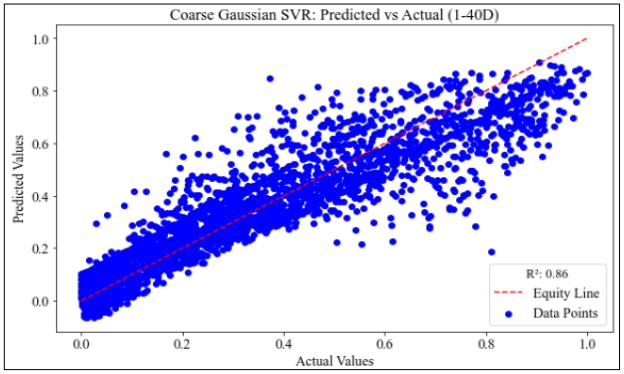


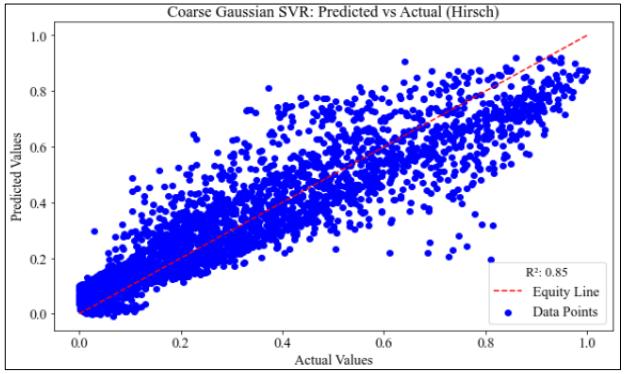




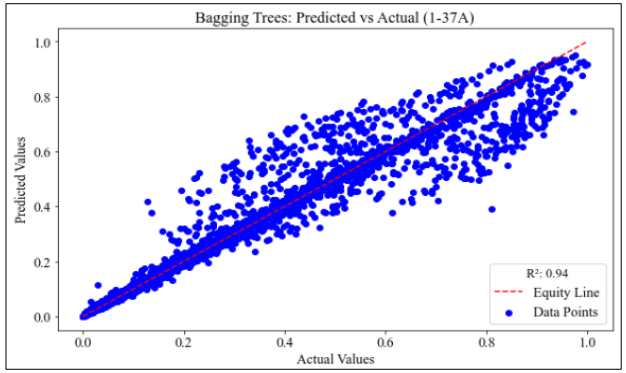
(b) Fine Regression Trees (MPa)

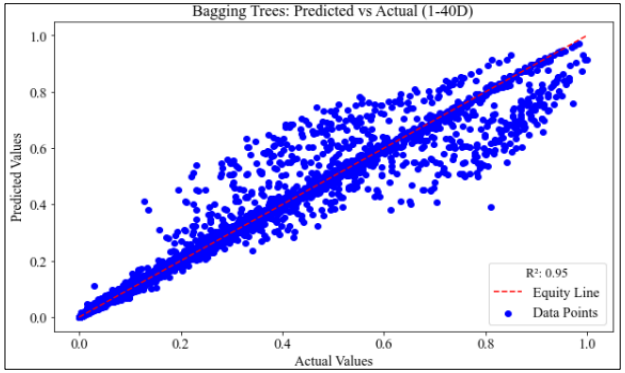


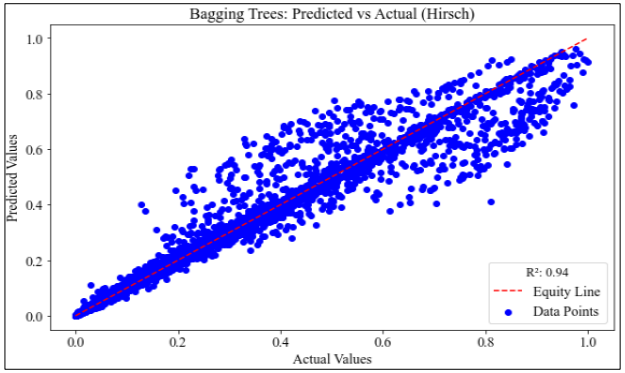




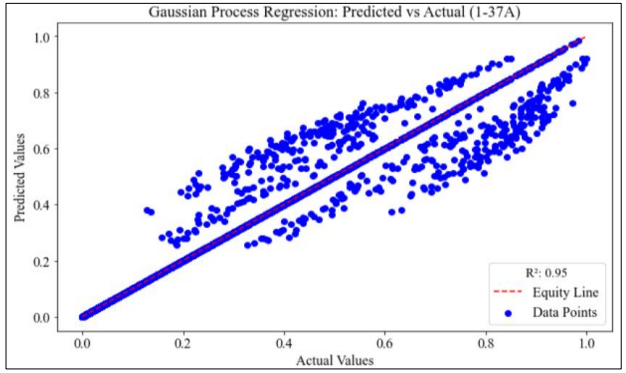
(c) Coarse Gaussian Support Vector Regression (MPa)

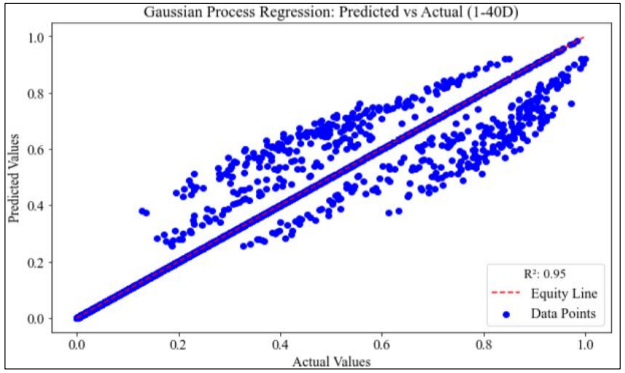


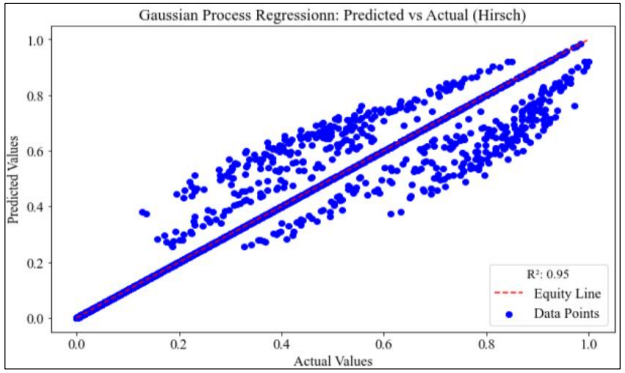




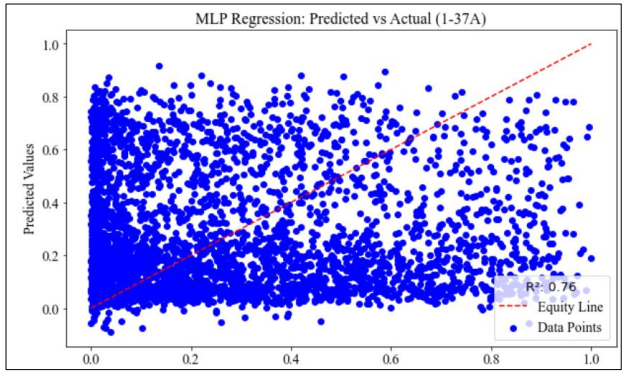
(d) Bagged ETs (MPa)

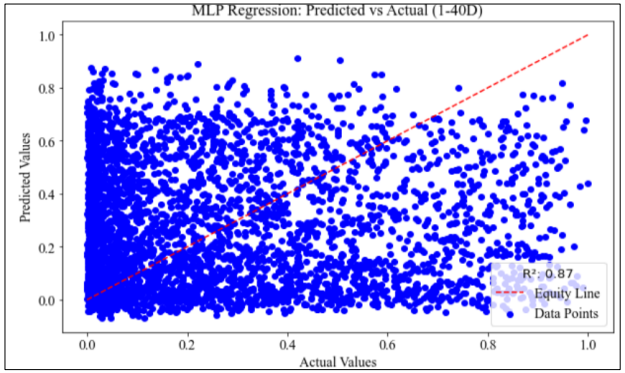


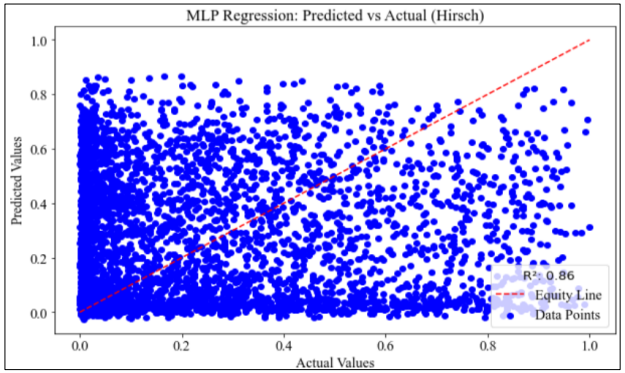




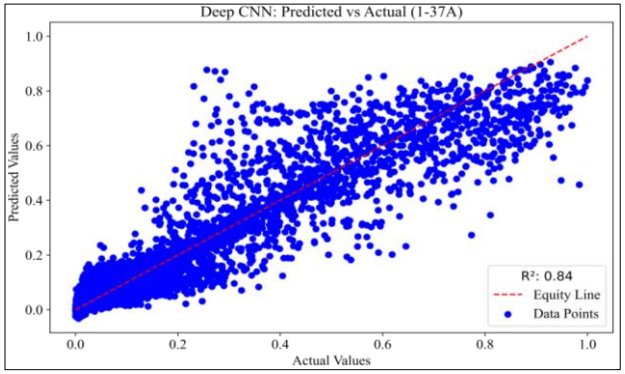
(e) Exponential GPR (MPa)

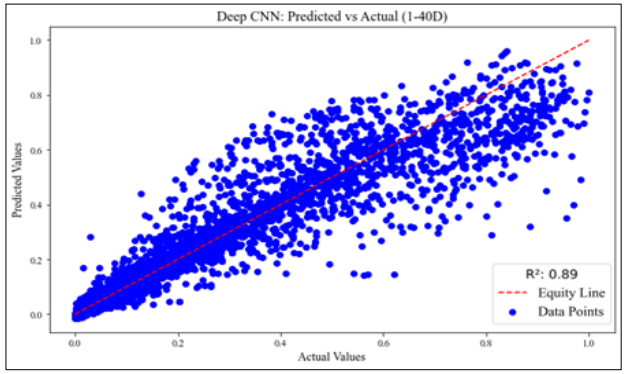


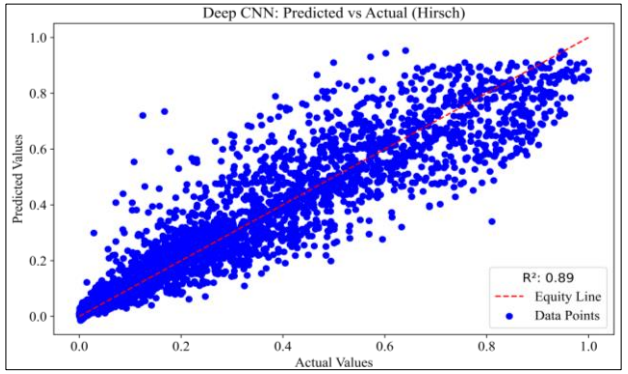




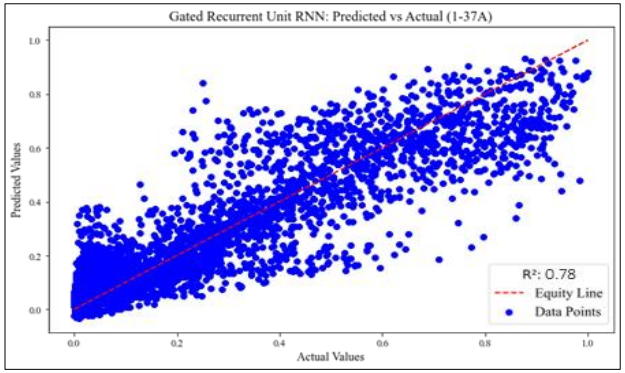
(f) Feed-forward MLP ANN (MPa)

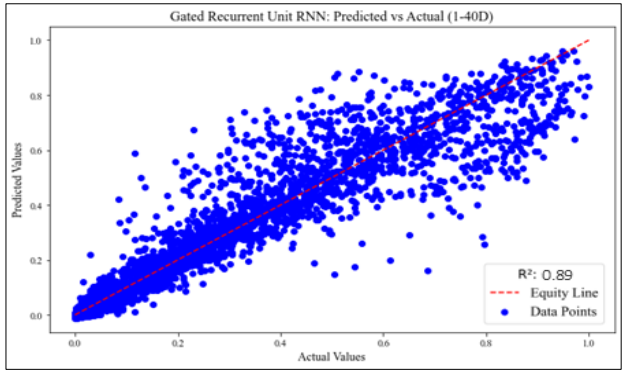


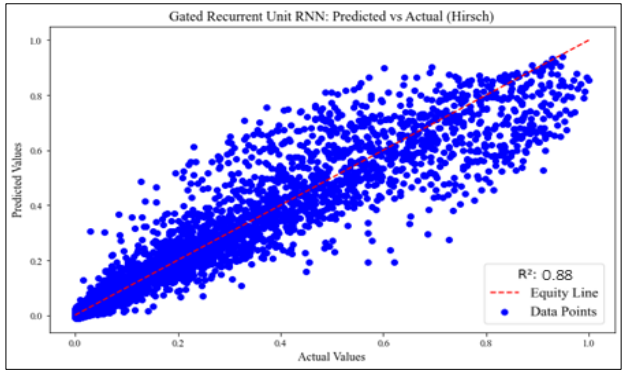




(g) Deep CNN (MPa)







(h) GRU RNN (MPa)

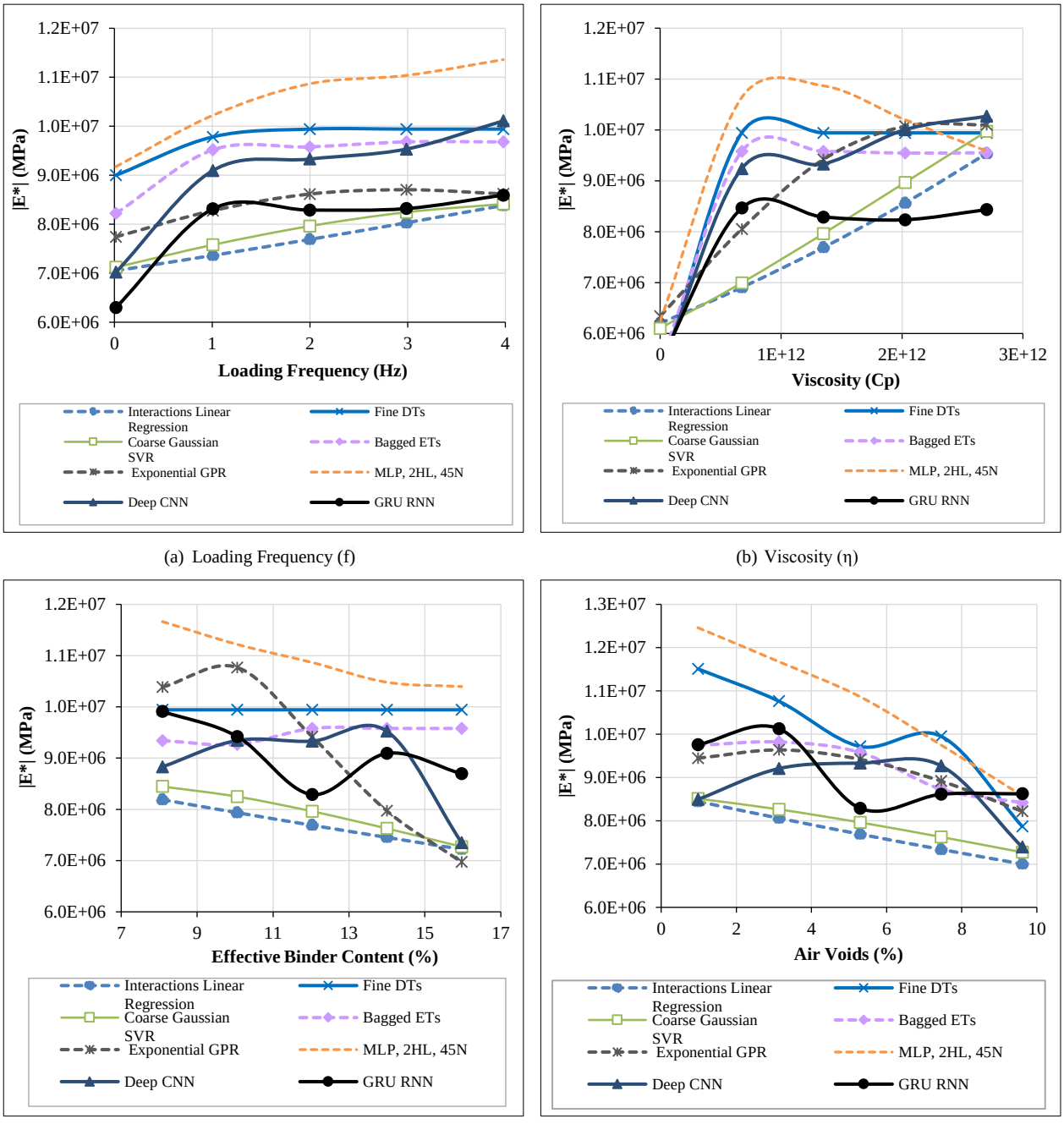
Figure 3. Predicted versus Actual |E*| Values for Best-Performing Models

3.3. Sensitivity Analysis

This section discusses the findings obtained throughout the sensitivity analysis. The idea behind sensitivity analysis is to examine and assess the influence of varying individual features on the trend of |E*|. This is carried out to check if the trends obtained from the ML models align well with the anticipated trends based on expert experience and well-established |E*| regression models such as the one considered in this study. This, in turn, ensures that no overfitting occurs, thereby confirming the rationality of the developed ML models, allowing them to be applied to any other sets of |E*| data. In the sensitivity analysis, four features from each dataset were randomly selected, and to assess the rationality of |E*| prediction, the value of one of these features was varied while reserving the values of the other features constant (i.e., conducting a controlled experiment). This test aids in detecting overfitting, even if a high accuracy performance was achieved, as there is still a risk of overfitting.

3.3.1. Witczak NCHRP 1-37A Model

For the initial dataset derived from Witczak NCHRP 1-37A parameters, the frequency, η, V_{beff}, and V_a percentage features were investigated. Figure 4 depicts the sensitivity outcomes for the 1-37A dataset. The analysis of the top-performing models reveals intricate relationships between the dynamic modulus |E*| and the assessed features. Nevertheless, the outcomes affirm the hypothesized trends for all attributes. These trends denote that |E*| escalates with an increase in loading frequency (the rate at which loads are applied), as exhibited in Figure 4-a, which indicates a stiffer asphalt mixture under more rapid loading conditions. Regarding the η feature (the resistance to flow of the asphalt binder used in the HMA mixture), depicted in Figure 4-b, a rise in viscosity, which hinders the flow of the asphalt binder, leads to an enhanced |E*|, mirroring the expectation that a higher viscosity equates to greater mixture stiffness. Additionally, as illustrated in Figure 4-c, an augmentation in V_{beff} (the amount of asphalt binder present in the mixture) correlates with diminished |E*| values across all models, barring some discrepancies noted for the GRU RNN model, suggesting a potential overfitting issue. Lastly, the V_a feature (the spaces within the AC mixture that are filled with air), shown Figure 4-d, demonstrates that an increased air void content weakens the predicted |E*| values for the majority of models, aligning with the premise that more air voids result in a less dense and thus less rigid AC mixture.



(a) Loading Frequency (f)

(b) Viscosity (η)

(c) Effective Binder Content (V_{beff})

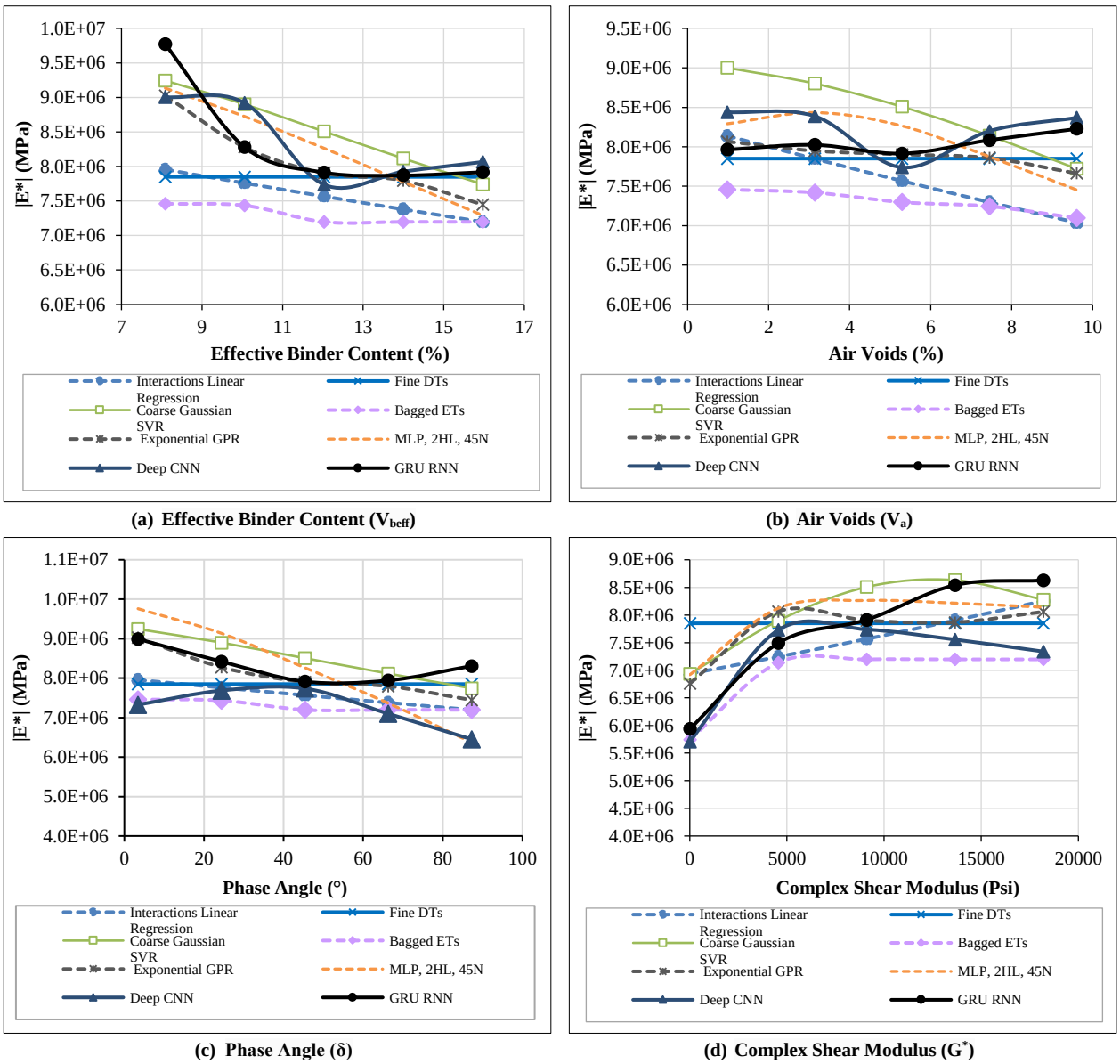
(d) Air Voids (V_a)

Figure 4. 1-37A dataset sensitivity analysis

In terms of rationality, the ET and DT models consistently reflected high rationality across all four tested features. Their outputs corresponded well with the expected physical trends, such as the increase in |E*| with loading frequency and viscosity and the decrease with more effective binder content and air voids. The MLP and Coarse Gaussian SVR models also displayed a strong alignment with the expected trends, although with slightly less consistency than the ET and DT models. On the other hand, the GRU RNN model exhibited less rational behavior, as indicated by the unexpected variations in its sensitivity to the $V_{beff\ and}$ Va features. This inconsistency hints at the model's overfitting to the training data, causing it to learn noise and anomalies rather than the underlying physical relationships. The Deep CNN, while not as erratic as the GRU RNN, showed occasional deviations from expected trends, which may indicate a need for further calibration of the model to improve its interpretability and alignment with physical principles.

3.3.2. Witczak NCHRP 1-40D Model

For the second dataset, derived from Witczak NCHRP 1-40D, the sensitivity of model predictions to variations in Vbeff, AV percentage, δ, and G* features were explored. The corresponding sensitivity analysis is encapsulated in Figure 5. The analysis revealed that the |E*| decreases as the V_{beff} increases, as depicted in Figure 5-a. This trend, evident across most models, suggests that a higher binder content typically leads to a softer asphalt mixture. Similarly, the results for the V_a feature, presented in Figure 5-b, reinforced the expected inverse relationship between air voids and |E*|, whereby an increase in air voids leads to a reduction in mixture stiffness. In examining the phase angle (a measure of the viscoelastic behavior of asphalt binder), δ, the models generally indicated a decrease in |E*| with an increase in δ, as illustrated in Figure 5-c. This outcome is consistent with the understanding that a higher phase angle reflects a more viscous and less stiff asphalt binder, translating to a softer mixture. Finally, when evaluating the G* feature (which is a measure of the asphalt binder resistance to deformation under shear stress), the models, for the most part, correctly predicted an increase in |E*| with higher G* values, as shown in Figure 5-d. This finding aligns with the notion that a stiffer binder, indicated by a higher G*, contributes to a stiffer asphalt concrete mixture.



(a) Effective Binder Content (V_{beff}) **(b) Air Voids (V_a)**

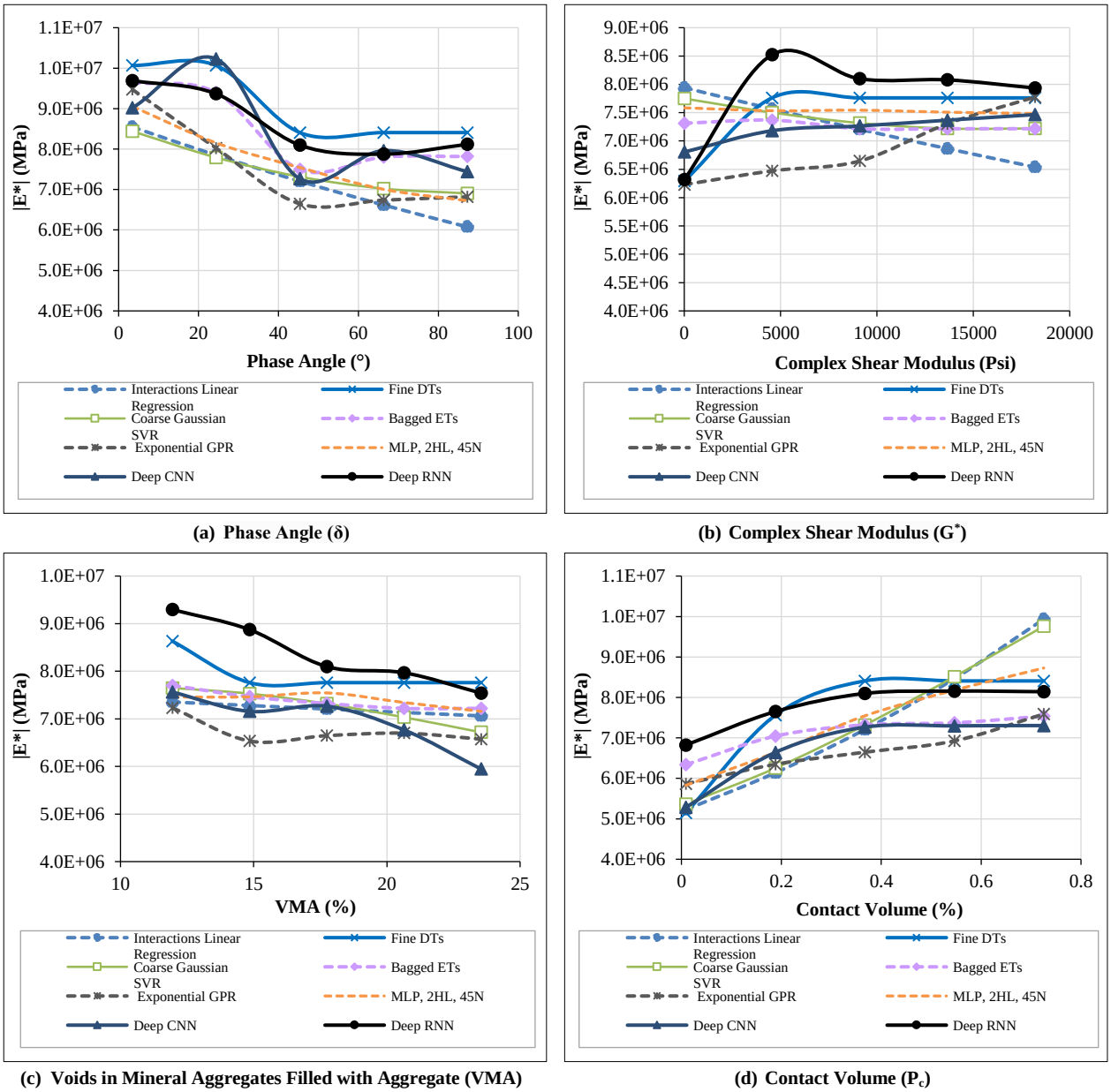
(c) Phase Angle (δ) **(d) Complex Shear Modulus (G^*)**

Figure 5. 1-40D dataset sensitivity analysis

In terms of model rationality, the ET and DT consistently demonstrated a high degree of rationality, with model outputs aligning well with expected physical behaviors across all four factors. Also, The Interactions between MLR and the exponential GPR model showcased a consistent and expected response in the sensitivity analysis, adhering to the anticipated physical relationships across all tested features of the 1-40D dataset. The MLP and Coarse Gaussian SVR models, while showing strong alignment for the most part, had occasional minor discrepancies. In contrast, the GRU RNN and Deep CNN models exhibited patterns that suggest a divergence from the expected physical relationships, which was particularly noticeable in their responses to $V_{beff\ and}$ V_a.

3.3.3. Hirsch Model

In the Hirsch dataset analysis, the sensitivity of the models to changes in the δ, G*, VMA, and Pc was evaluated. Figure 6 details these relationships and indicates that an increase in δ generally leads to softer asphalt mixtures with lower |E*| values, a finding consistent with the viscoelastic properties of asphalt binders (Figure 6-a). However, the Deep CNN, GRU RNN, and Exponential GPR models showed atypical responses to δ changes, suggesting potential overfitting. The trend for G* was as anticipated, with higher values correlating with increased |E*|, indicating stiffer asphalt mixtures (Figure 6-b). The VMA analysis, shown in Figure 6-c, confirmed that a higher VMA tends to weaken the mixture's cohesion, causes compromised interlocking, reduced cohesion, and increased susceptibility to deformation, leading to lower |E*| values. Conversely, a rise in P_c (which is the amount of surface area within the asphalt mixture where the aggregates come into contact with each other) results in a greater amount of interlocking and interaction between the individual aggregate particles, enhancing the mix's structural capacity, as shown in Figure 6-d, which was uniformly captured across the models.



(a) Phase Angle (δ) **(b) Complex Shear Modulus (G^*)**

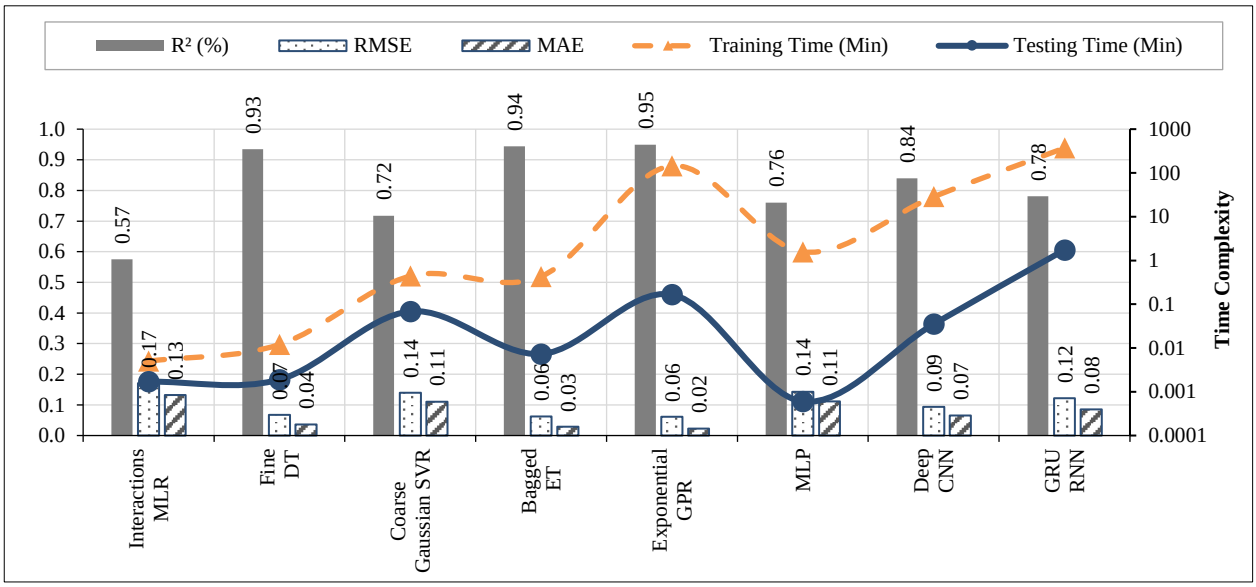
(c) Voids in Mineral Aggregates Filled with Aggregate (VMA) **(d) Contact Volume (P_c)**

Figure 6. Hirsch dataset sensitivity analysis

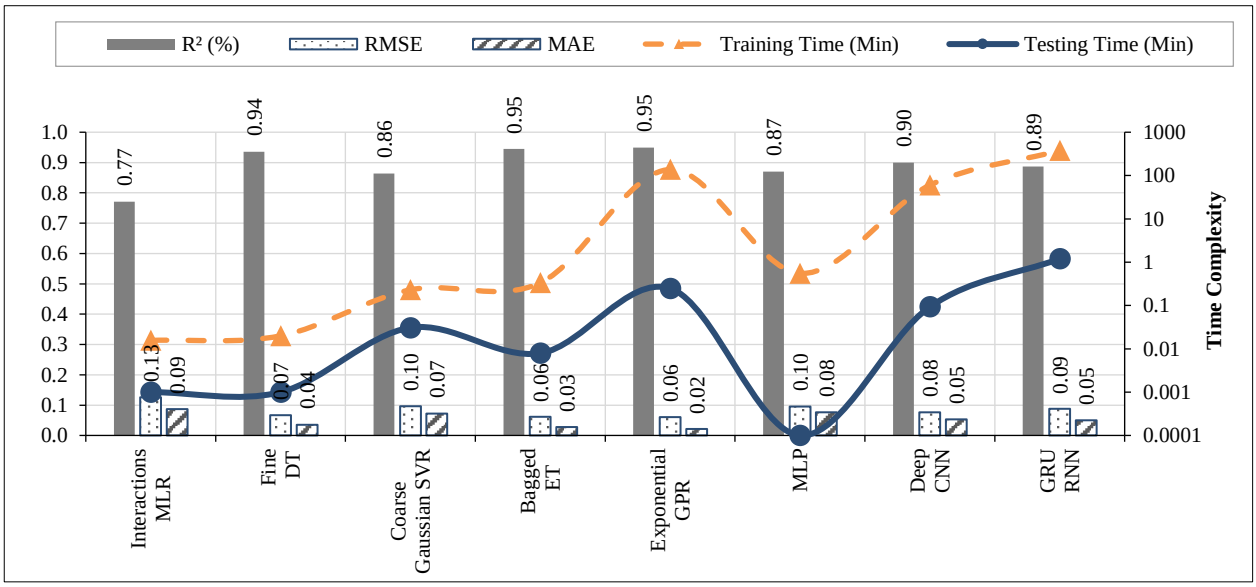
Comparing models' predictions' rationality, the ET and Fine DT models again exhibited strong rationality, consistently reflecting the physical expectations for all factors. The MLP and Coarse Gaussian SVR models largely followed the expected trends, displaying a high degree of understanding of the underlying physical phenomena. In contrast, the Deep CNN, GRU RNN, and Exponential GPR models occasionally diverged from these trends, particularly in their response to δ and G*, indicating that these models might be sensitive to the training data's precision or prone to overfitting. This divergence highlights the need for caution when interpreting their outputs, as it may affect the models' applicability to real-world scenarios without additional refinement.

3.4. Comparison of the Best-Performing Models for Each Regressor Type

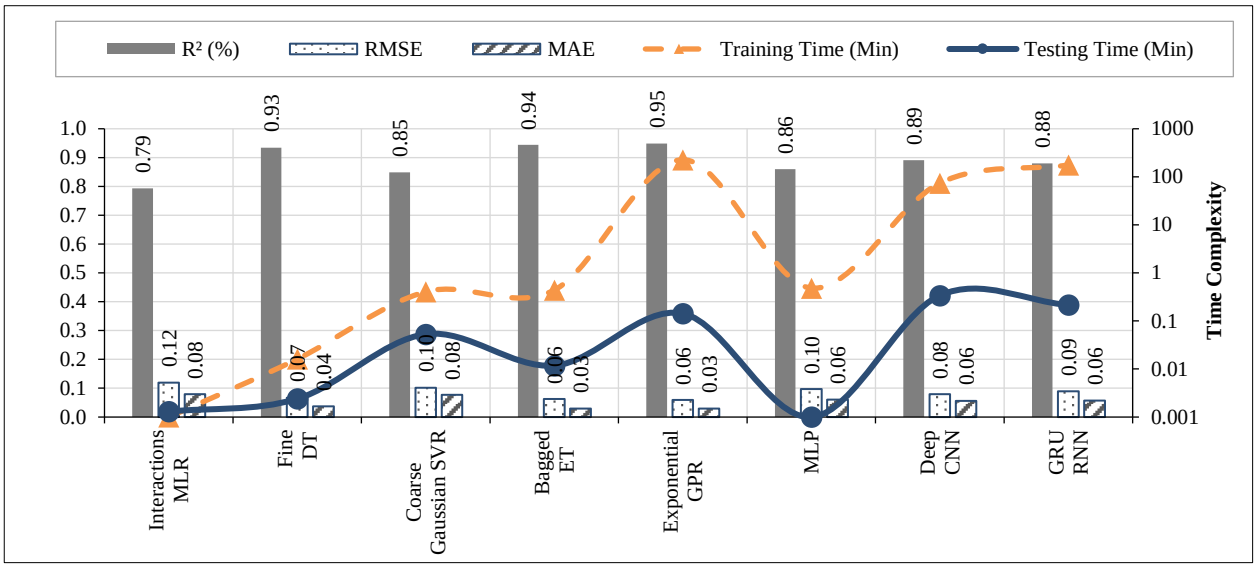
This section evaluates and compares the performance of best-performing models for each regressor type using various model performance measures. A total of eight different classical and DL regressors were optimized and evaluated: interactions MLR, fine DT with a minimum leaf size of 4, SVR with a coarse gaussian kernel and a kernel scale of 11, bagged ET (random forests) with a minimum leaf size of 8 and number of DT learners of 30, GPR with exponential kernel, feed-forward ANN with two hidden layers (45 neurons each and Relu activation function), as well as deep CNN structure, and gated RNN structure. Figure 7 summarizes the results obtained for comparing the fine-tuned models at the three datasets (i.e., Witczak NCHRP 1-37A, Witczak NCHRP 1-40D, and Hirsch-based datasets). Table 5 also summarizes the comparative results obtained and the final assessment scores computed for the fine-tuned regressors using the multi-criteria weighted average assessment.



(a) Based on Witczak NCHRP 1-37A input features



(b) Based on Witczak NCHRP 1-40D input features



(c) Based on Hirsch's input features

Figure 7. Comparative analysis of ML and DL models

Table 5. Comparative Results Summary Table

		Interactions MLR	Fine DT	Coarse Gaussian SVR	Bagged ET	Exponential GPR	MLP	Deep CNN	GRU RNN
1-37A Dataset	RMSE	0.17	0.07	0.14	0.06	0.06	0.14	0.09	0.12
	MAE	0.13	0.04	0.11	0.03	0.02	0.11	0.07	0.08
	R^2	0.57	0.93	0.72	0.94	0.95	0.76	0.84	0.78
	Training Time (Min)	0.01	0.01	0.44	0.43	145.11	1.57	28.59	370.90
	Testing Time (Min)	0.001	0.001	0.07	0.01	0.17	0.00	0.04	1.74
	Rationality	1.00	1.00	0.75	1.00	0.50	0.75	0.50	0.50
	Weighted Average	**0.878**	**0.97**	**0.863**	**0.974**	**0.814**	**0.876**	**0.847**	**0.513**
1-40D Dataset	RMSE	0.13	0.07	0.10	0.06	0.06	0.10	0.08	0.09
	MAE	0.09	0.04	0.07	0.03	0.02	0.08	0.05	0.05
	R^2	0.77	0.94	0.86	0.95	0.95	0.87	0.90	0.89
	Training Time (Min)	0.02	0.02	0.23	0.33	138.06	0.55	60.00	375.57
	Testing Time (Min)	0.001	0.001	0.03	0.01	0.25	0.00	0.09	1.20
	Rationality	1.00	1.00	0.75	1.00	0.75	0.75	0.50	0.50
	Weighted Average	**0.925**	**0.972**	**0.902**	**0.975**	**0.841**	**0.906**	**0.839**	**0.541**
Hirsch Dataset	RMSE	0.12	0.07	0.10	0.06	0.06	0.10	0.08	0.09
	MAE	0.08	0.04	0.08	0.03	0.03	0.06	0.06	0.06
	R^2	0.79	0.93	0.85	0.94	0.95	0.86	0.89	0.88
	Training Time (Min)	0.001	0.02	0.41	0.44	226.87	0.48	74.09	176.19
	Testing Time (Min)	0.001	0.001	0.05	0.01	0.14	0.00	0.34	0.22
	Rationality	1.00	1.00	0.75	1.00	0.50	0.75	0.50	0.50
	Weighted Average	**0.931**	**0.969**	**0.879**	**0.97**	**0.658**	**0.908**	**0.654**	**0.634**

It was observed that the predicted performance results were consistent over the three datasets. Starting with the 1-37A dataset, the bagged ET (random forests) and DT demonstrated exceptional predictive accuracy with R^2 values of 0.94 and 0.93, respectively. The exponential GPR also performed notably with an R^2 of 0.95, albeit at the cost of a significantly higher testing time, which ranged from 138 to 226 minutes. The ET model was not only accurate but also efficient, with minimal training and testing times of 0.43 minutes and 0.01 minutes, respectively, along with the full weight of the rationality score, yielding a weighted average score of 0.974. In contrast, the GRU RNN lagged in performance with a weighted average of 0.513, indicating a suboptimal balance of accuracy and complexity for this dataset.

Moving to the 1-40D dataset, results revealed a continuation of the high performance from ET and DT, with R^2 values at 0.95 and 0.94, respectively. The GPR maintained its accuracy with an R^2 of 0.95. However, the computational demand for GPR remained high, with a prolonged testing time of 0.25 minutes. The ET, once again, emerged as a model of high accuracy, low complexity, and superior rationality, reflected by its top tier weighted average score of 0.975. Meanwhile, the GRU RNN's efficiency slightly improved compared to the 1-37A dataset, but it still held the lowest weighted average score at 0.542.

Similarly, in the Hirsch dataset, ET and GPR continued to excel in accuracy with R^2 values of 0.94 and 0.95, respectively. The ET confirmed its dominance in efficiency, with training and testing times well below the benchmark and the strongest rationality performance, contributing to a high weighted average score of 0.97. Despite its accuracy, the GPR was markedly less efficient, with the highest training time among all models, culminating in a lower weighted average score of 0.659. The GRU RNN's performance saw an uptick, yet it remained the least efficient model with a weighted average score of 0.634.

When we synthesize the findings across all datasets, the ET consistently ranks as the top performer, showcasing an unparalleled blend of high accuracy, operational efficiency, and prediction rationality. The Fine DT also stands out as a robust model with impressive accuracy and a good complexity profile. The MLP and Deep CNN are recognized for their robustness across datasets, indicating their suitability for complex problems where their architectural depth can be leveraged. On the other hand, while the Exponential GPR scores highly in accuracy, its computational demand detracts from its appeal, particularly in time-sensitive environments. Despite its potential for capturing complex patterns in data, the GRU RNN has not shown the same level of efficiency or accuracy as its counterparts.

3.5. Comparison with Previous Developed Models from Literature

In prior research by El-Badawy et al. [14], the best-performing model was identified as an artificial neural network (ANN) with two hidden layers, each consisting of 36 neurons, using a sigmoid activation function. This model achieved an accuracy of 91% for the Witczak NCHRP 1-37A and 1-40D datasets, and 90% for the Hirsch dataset. In comparison, the models developed in this study, particularly the GPR, ETs, and DT, demonstrated superior predictive capabilities, achieving R^2 values approximately 4% higher than those reported in the earlier study. This improvement can be attributed to the integration of advanced optimization techniques and a systematic evaluation of hyperparameters, which significantly enhanced the accuracy and robustness of the predictions. Furthermore, including a broader dataset encompassing diverse climatic conditions and material properties in this study provided a more comprehensive evaluation of model performance, ensuring greater generalizability of the predictive models. Unlike the earlier work, which relied solely on ANN architectures, this study explored a wider spectrum of machine learning and deep learning models, highlighting the trade-offs between computational complexity and accuracy to offer tailored solutions for practical applications in pavement design.

3.6. Threats to Validity

The threats to internal and external validities are dependent upon the study design. This section delves into the validities of the approach employed in this research, shedding light on the potential benefits and constraints associated with the |E*| prediction framework.

3.6.1. Internal Threats

- The inclusion of a diverse dataset comprising over 3720 |E*| records from AC mixtures collected across different climatic regions (Idaho state and KSA) and featuring various aggregate gradations and binder performance grades enhances the robustness and representativeness of the study.
- The careful selection of features aligns with the most widely used NCHRP and Hirsch regression models, ensuring a comprehensive and fair investigation. This approach enhances the study's relevance using features that resonate with established industry standards.
- The study employs the k-fold cross-validation technique for training and testing to address potential biases in dataset separation. This robust approach contributes to internal validity by ensuring that the model's performance is evaluated across multiple subsets of the dataset, reducing the risk of overfitting.
- Despite using widely accepted performance metrics, the choice of MAE, RMSE, R^2, and training time may not capture all aspects of model performance.

3.6.2. External Threats

- The study's external validity is fortified by the potential applicability of the developed |E*| prediction framework. Different departments of transportation and highway agencies can adopt this framework, offering a practical and generalizable solution for |E*| prediction.

- The generalizability of the framework assumes that different transportation and highway agencies will adopt and implement it. However, practical challenges, institutional differences, or resistance to change may hinder widespread adoption.
- While the dataset is diverse and captures varying climatic conditions (hot and cold climates) and material properties, it is limited to Superpave mixes from specific geographical locations (Idaho state and KSA). This geographic restriction might impact the model's universality, particularly in regions with distinct asphalt binder properties or construction practices. However, including datasets from significantly different climatic regions enhances the framework's representativeness and potential applicability to other locations with similar climates.
- The study acknowledges the external threat of generalizability. While the framework may apply to different departments, each region's specific conditions and characteristics might not be fully captured, limiting the external validity of the results.

4. Summary and Conclusions

This research study utilized |E*| database collected from Idaho State and the Kingdom of Saudi Arabia related to AC mixtures and asphalt binders to investigate the performance of state-of-the-art classic ML and DL algorithms (including MLR, DT, SVM, ET, GPR, and ANN, as well as CNN and RNN) for the development of more accurate and rational predictive |E*| models. A total of 13 features affecting the |E*| were aggregated, considered as the independent features, and regrouped for the |E*| prediction based on three renewed regression models, including Witczak NCHRP 1-37A η-based, Witczak NCHRP 1-40D G*, δ- based, and Hirsch G*-based predictive models. These features include f, η, V_{beff}, V_a, ρ_{34}, ρ_{38}, ρ_4, ρ_{200}, G*, δ, VMA, VFA, and P_c.

The developed and fine-tuned models were evaluated based on multi-stage assessment criteria. Firstly, the developed ML and DL regressors were compared in terms of various modeling accuracy and complexity performance measures of effectiveness. Secondly, a sensitivity analysis of the predicted results was conducted to test the rationality of the prediction, hence testing the impact of the considered features on |E*| prediction, and to predict possible regressors overfitting or memorization. In this stage, the developed best-performing models with the highest accuracy, lowest complexity, and results rationality were identified based on a weighted average score representing the overall multi-criteria rank. The main findings of this study can be summarized as follows:

- The fine-tuned ML and DL models' structures included the interactions linear regression, fine DT with a minimum leaf size of 4, SVR with a coarse Gaussian kernel and scale value of 11, the bagged ET, the GPR with exponential kernel, the ANN with two hidden layers, 45 neurons each, and ReLU activation function, deep CNN and GRU RNN.
- The best-performing models over the fine-tuned hyperparameters in terms of modeling accuracy were the GPR with exponential kernel followed by the bagged ET, fine DT with a minimum leaf size of 4, which reported statistical measures of R^2 of 0.95, 0.94, 0.94, and 0.91, respectively.
- The best-performing model in terms of modeling complexity was the bagged ET, with a training time ranging between 0.33 and 0.44 minutes and a testing time of 0.01 minutes.
- The predicted versus actual |E*| plots aligned with the preceding results and illustrated that the GPR and Bagged ET were the best outperforming models.
- Comparing the developed regressors in this study with the regression and ANN modeling from previous literature studies showed that the developed models outperformed previous ones in terms of prediction performance with 4% higher R^2.
- The sensitivity analysis of the |E*| prediction across the three datasets showed that the deep CNN, GRU RNN, and the GPR had some variations from typical |E*| trends, indicating a possibility of overfitting occurrence.
- The sensitivity analysis also showed that the ET and DT consistently demonstrated a high degree of rationality over all the tested datasets.

The results presented demonstrate that the classic ML and DL algorithms developed in this study produce predictions with higher accuracy, lower complexity, and greater rationality compared to existing models in the literature. This demonstrates the applicability of the proposed best-performing models for improved |E*| prediction. Pavement designers and practitioners can adopt the developed feature engineering-ML and DL-based approach to estimate more accurate and rational |E*| predictions.

Future studies should integrate datasets encompassing a broader range of AC mixtures, asphalt binders, and aggregates to enhance the generalization capability of ML and DL models. This will improve the accuracy of |E*| predictions and enhance their application in modern pavement analysis. Including data from different design methodologies (e.g., Marshall or Hveem), non-Superpave mixtures, and innovative materials such as RAP, WMA, and polymer-modified binders will address the study's current limitations and extend the framework's applicability to diverse materials and practices.

To improve model robustness, advanced feature selection techniques, such as recursive feature elimination (RFE), principal component analysis (PCA), or mutual information, should be explored to identify the most significant predictors. This would enhance both accuracy and computational efficiency. Additionally, mitigating overfitting in deep learning models is critical. Regularization techniques like dropout, weight decay, early stopping, and simplifying model architectures could improve generalization. Incorporating hybrid approaches that combine mechanistic models with data-driven methods could further align model outputs with physical principles, reducing sensitivity to noise and anomalies.

Furthermore, while bagging methods demonstrated an effective balance of accuracy and efficiency, scenarios involving high variability or complex, non-linear relationships may justify more advanced models like boosting techniques or hybrid ensembles. Exploring such models alongside scalable architectures will ensure adaptability to larger and more complex datasets.

Lastly, collaboration with transportation agencies and industry practitioners is essential to gather region-specific data and address external threats to validity. This will enhance the framework's acceptance and usability across diverse regional and institutional contexts.

5. Declarations

5.1. Author Contributions

Conceptualization, W.Z. and S.E.; methodology, L.O.; software, L.O.; validation, S.E., R.A.E., and A.A.; formal analysis, L.O.; investigation, W.Z.; resources, S.E.; writing—original draft preparation, L.O.; writing—review and editing, W.Z., S.E., R.A.E., and A.A.; visualization, L.O. and W.Z.; supervision, W.Z. All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available in the article.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Conflicts of Interest

The authors declare no conflict of interest.

6. References

[1] Bi, Y., Guo, F., Zhang, J., Pei, J., & Li, R. (2021). Correlation analysis between asphalt binder/asphalt mastic properties and dynamic modulus of asphalt mixture. Construction and Building Materials, 276, 122256. doi:10.1016/j.conbuildmat.2021.122256.

[2] Bhattacharjee, S., & Mallick, R. (2012). Determining damage development in hot-mix asphalt with use of continuum damage mechanics and small-scale accelerated pavement test. Transportation Research Record, 2296, 125–134. doi:10.3141/2296-13.

[3] Jamshidi, A., White, G., & Hosseinpour, M. (2021). Revisiting the correlation between the dynamic modulus and the flexural modulus of hot mixture asphalt. Construction and Building Materials, 296, 123697. doi:10.1016/j.conbuildmat.2021.123697.

[4] Rodezno, M. C., & Kaloush, K. E. (2009). Comparison of asphalt rubber and conventional mixture properties: Considerations for mechanistic-empirical pavement design guide implementation. Transportation Research Record, 2126, 132–141. doi:10.3141/2126-16.

[5] El-Hakim, R. A., El-Badawy, S. M., Gabr, A. R., & Azam, A. M. (2016). Influence of Unbound Material Type and Input Level on Pavement Performance Using Mechanistic–Empirical Pavement Design Guide. Transportation Research Record, 2578(1), 21–28. doi:10.3141/2578-03.

[6] ARMAĞAN, K., SALTAN, M., TERZİ, S., & KIRAÇ, N. (2021). Comparison of dynamic elastisty modulus with different prediction approaches for Karaman – Konya highway pavement. Journal of Innovative Transportation, 2(1), 2102. doi:10.53635/jit.849544.

[7] Singh, D., Zaman, M., & Commuri, S. (2013). Artificial Neural Network Modeling for Dynamic Modulus of Hot Mix Asphalt Using Aggregate Shape Properties. Journal of Materials in Civil Engineering, 25(1), 54–62. doi:10.1061/(asce)mt.1943-5533.0000548.

[8] Rahman, A. S. M. A., & Tarefder, R. A. (2016). Dynamic modulus and phase angle of warm-mix versus hot-mix asphalt concrete. Construction and Building Materials, 126, 434–441. doi:10.1016/j.conbuildmat.2016.09.068.

[9] Zhang, M., Zhao, H., Fan, L., & Yi, J. (2022). Dynamic modulus prediction model and analysis of factors influencing asphalt mixtures using gray relational analysis methods. Journal of Materials Research and Technology, 19, 1312–1321. doi:10.1016/j.jmrt.2022.05.120.

[10] Barugahare, J., Amirkhanian, A. N., Xiao, F., & Amirkhanian, S. N. (2020). Predicting the dynamic modulus of hot mix asphalt mixtures using bagged trees ensemble. Construction and Building Materials, 260, 120468. doi:10.1016/j.conbuildmat.2020.120468.

[11] Khattab, A. M., El-Badawy, S. M., Al Hazmi, A. A., & Elmwafi, M. (2014). Evaluation of Witczak E* predictive models for the implementation of AASHTOWare-Pavement ME Design in the Kingdom of Saudi Arabia. Construction and Building Materials, 64, 360–369. doi:10.1016/j.conbuildmat.2014.04.066.

[12] Yu, H., & Shen, S. (2012). An investigation of dynamic modulus and flow number properties of asphalt mixtures in Washington State. Report No. TNW, 709867.

[13] Khattab, A. M., El-Badawy, S. M., Al Hazmi, A. A., & Elmwafi, M. (2015, April). Comparing Witczak NCHRP 1-40D with Hirsh E* predictive models for Kingdom of Saudi Arabia asphalt mixtures. The 3rd Middle East Society of Asphalt Technologists (MESAT) Conference, 6-8 April, 2015, Dubai, United Arab Emirates.

[14] El-Badawy, S., Abd El-Hakim, R., & Awed, A. (2018). Comparing Artificial Neural Networks with Regression Models for Hot-Mix Asphalt Dynamic Modulus Prediction. Journal of Materials in Civil Engineering, 30(7), 1–11. doi:10.1061/(asce)mt.1943-5533.0002282.

[15] Al-Tawalbeh, A., Sirin, O., Sadeq, M., Sebaaly, H., & Masad, E. (2022). Evaluation and calibration of dynamic modulus prediction models of asphalt mixtures for hot climates: Qatar as a case study. Case Studies in Construction Materials, 17, 1580. doi:10.1016/j.cscm.2022.e01580.

[16] Uwanuakwa, I. D., Amir, I. Y., & Umba, L. N. (2024). Enhanced asphalt dynamic modulus prediction: A detailed analysis of artificial hummingbird algorithm-optimised boosted trees. Journal of Road Engineering, 4(2), 224–233. doi:10.1016/j.jreng.2024.05.001.

[17] Acharjee, P. K., Souliman, M. I., Freyle, F., & Fuentes, L. (2024). Development of Dynamic Modulus Prediction Model Using Artificial Neural Networks for Colombian Mixtures. Journal of Transportation Engineering, Part B: Pavements, 150(1), 1402. doi:10.1061/jpeodx.pveng-1402.

[18] Owais, M. (2024). Preprocessing and postprocessing analysis for hot-mix asphalt dynamic modulus experimental data. Construction and Building Materials, 450, 138693. doi:10.1016/j.conbuildmat.2024.138693.

[19] Sakhaeifar, M. S., Richard Kim, Y., & Kabir, P. (2015). New predictive models for the dynamic modulus of hot mix asphalt. Construction and Building Materials, 76, 221–231. doi:10.1016/j.conbuildmat.2014.11.011.

[20] Singh, D., Zaman, M., & Commuri, S. (2011). Evaluation of predictive models for estimating dynamic modulus of hot-mix asphalt in Oklahoma. Transportation Research Record, 2210(2210), 57–72. doi:10.3141/2210-07.

[21] Chen, H., Saba, R. G., Liu, G., Barbieri, D. M., Zhang, X., & Hoff, I. (2023). Influence of material factors on the determination of dynamic moduli and associated prediction models for different types of asphalt mixtures. Construction and Building Materials, 365, 130134. doi:10.1016/j.conbuildmat.2022.130134.

[22] Behnood, A., & Daneshvar, D. (2020). A machine learning study of the dynamic modulus of asphalt concretes: An application of M5P model tree algorithm. Construction and Building Materials, 262, 120544. doi:10.1016/j.conbuildmat.2020.120544.

[23] Daneshvar, D., & Behnood, A. (2022). Estimation of the dynamic modulus of asphalt concretes using random forests algorithm. International Journal of Pavement Engineering, 23(2), 250–260. doi:10.1080/10298436.2020.1741587.

[24] Awed, A. M., Awaad, A. N., Kaloop, M. R., Hu, J. W., El-Badawy, S. M., & Abd El-Hakim, R. T. (2023). Boosting Hot Mix Asphalt Dynamic Modulus Prediction Using Statistical and Machine Learning Regression Modeling Techniques. Sustainability (Switzerland), 15(19). doi:10.3390/su151914464.

[25] Liu, J., Liu, F., Zheng, C., Zhou, D., & Wang, L. (2022). Optimizing asphalt mix design through predicting effective asphalt content and absorbed asphalt content using machine learning. Construction and Building Materials, 325(December), 126607. doi:10.1016/j.conbuildmat.2022.126607.

[26] Hu, X., & Solanki, P. (2021). Predicting Resilient Modulus of Cementitiously Stabilized Subgrade Soils Using Neural Network, Support Vector Machine, and Gaussian Process Regression. International Journal of Geomechanics, 21(6), 04021073. doi:10.1061/(asce)gm.1943-5622.0002029.

[27] Uwanuakwa, I. D., Busari, A., Ali, S. I. A., Mohd Hasan, M. R., Sani, A., & Abba, S. I. (2022). Comparing Machine Learning Models with Witczak NCHRP 1-40D Model for Hot-Mix Asphalt Dynamic Modulus Prediction. Arabian Journal for Science and Engineering, 47(10), 13579–13591. doi:10.1007/s13369-022-06935-x.

[28] Ceylan, H., Gopalakrishnan, K., & Kim, S. (2008). Advanced approaches to hot-mix asphalt dynamic modulus prediction. Canadian Journal of Civil Engineering, 35(7), 699–707. doi:10.1139/L08-016.

[29] Ceylan, H., Gopalakrishnan, K., & Kim, S. (2009). Looking to the future: The next-generation hot mix asphalt dynamic modulus prediction models. International Journal of Pavement Engineering, 10(5), 341–352. doi:10.1080/10298430802342690.

[30] Ceylan, H., Schwartz, C. W., Kim, S., & Gopalakrishnan, K. (2009). Accuracy of Predictive Models for Dynamic Modulus of Hot-Mix Asphalt. Journal of Materials in Civil Engineering, 21(6), 286–293. doi:10.1061/(asce)0899-1561(2009)21:6(286).

[31] Gong, H., Sun, Y., Dong, Y., Han, B., Polaczyk, P., Hu, W., & Huang, B. (2020). Improved estimation of dynamic modulus for hot mix asphalt using deep learning. Construction and Building Materials, 263, 119912. doi:10.1016/j.conbuildmat.2020.119912.

[32] Ghasemi, P., Aslani, M., Rollins, D. K., & Williams, R. C. (2019). Principal component neural networks for modeling, prediction, and optimization of hot mix asphalt dynamics modulus. Infrastructures, 4(3), 2019. doi:10.3390/infrastructures4030053.

[33] Rezazadeh Eidgahee, D., Jahangir, H., Solatifar, N., Fakharian, P., & Rezaeemanesh, M. (2022). Data-driven estimation models of asphalt mixtures dynamic modulus using ANN, GP and combinatorial GMDH approaches. Neural Computing and Applications, 34(20), 17289–17314. doi:10.1007/s00521-022-07382-3.

[34] Zhang, C., Ildefonzo, D. G., Shen, S., Wang, L., & Huang, H. (2023). Implementation of ensemble Artificial Neural Network and MEMS wireless sensors for In-Situ asphalt mixture dynamic modulus prediction. Construction and Building Materials, 377, 131118. doi:10.1016/j.conbuildmat.2023.131118.

[35] Barugahare, J., Amirkhanian, A. N., Xiao, F., & Amirkhanian, S. N. (2022). ANN-based dynamic modulus models of asphalt mixtures with similar input variables as Hirsch and Witczak models. International Journal of Pavement Engineering, 23(5), 1328–1338. doi:10.1080/10298436.2020.1799209.

[36] Mohammadi Golafshani, E., Behnood, A., & Karimi, M. M. (2021). Predicting the dynamic modulus of asphalt mixture using hybridized artificial neural network and grey wolf optimizer. International Journal of Pavement Engineering, 1–11. doi:10.1080/10298436.2021.2005056.

[37] Moussa, G. S., & Owais, M. (2020). Pre-trained deep learning for hot-mix asphalt dynamic modulus prediction with laboratory effort reduction. Construction and Building Materials, 265, 120239. doi:10.1016/j.conbuildmat.2020.120239.

[38] Moussa, G. S., & Owais, M. (2021). Modeling Hot-Mix asphalt dynamic modulus using deep residual neural Networks: Parametric and sensitivity analysis study. Construction and Building Materials, 294, 123589. doi:10.1016/j.conbuildmat.2021.123589.

[39] Liu, J., Liu, F., Wang, Z., Fanijo, E. O., & Wang, L. (2023). Involving prediction of dynamic modulus in asphalt mix design with machine learning and mechanical-empirical analysis. Construction and Building Materials, 407, 133610. doi:10.1016/j.conbuildmat.2023.133610.

[40] Broothaerts, W., Cordeiro, F., Corbisier, P., Robouch, P., & Emons, H. (2020). Log transformation of proficiency testing data on the content of genetically modified organisms in food and feed samples: is it justified? Analytical and Bioanalytical Chemistry, 412(5), 1129–1136. doi:10.1007/s00216-019-02338-4.

[41] Bridges, W. C., Calkin, N. J., Kenyon, C. M., & Saltzman, M. J. (2022). Log transformations: What not to expect when you're expecting. Communications in Statistics - Theory and Methods, 51(5), 1514–1521. doi:10.1080/03610926.2020.1771368.

[42] Manandhar, B., & Nandram, B. (2021). Hierarchical Bayesian models for continuous and positively skewed data from small areas. Communications in Statistics - Theory and Methods, 50(4), 944–962. doi:10.1080/03610926.2019.1645853.

[43] Jayalakshmi, T., & Santhakumaran, A. (2011). Statistical Normalization and Back Propagationfor Classification. International Journal of Computer Theory and Engineering, 3(1), 89–93. doi:10.7763/ijcte.2011.v3.288.

[44] Shalabi, L. Al, Shaaban, Z., & Kasasbeh, B. (2006). Data Mining: A Preprocessing Engine. Journal of Computer Science, 2(9), 735–739. doi:10.3844/jcssp.2006.735.739.

[45] Yi, B. J., Lee, D. G., & Rim, H. C. (2015). The Effects of Feature Optimization on High-Dimensional Essay Data. Mathematical Problems in Engineering, 421642. doi:10.1155/2015/421642.

[46] Alnaqbi, A. J., Zeiada, W., Al-Khateeb, G. G., Hamad, K., & Barakat, S. (2023). Creating Rutting Prediction Models through Machine Learning Techniques Utilizing the Long-Term Pavement Performance Database. Sustainability (Switzerland), 15(18), 13653. doi:10.3390/su151813653.

[47] Zeiada, W., Hamad, K., Omar, M., Underwood, B. S., Khalil, M. A., & Karzad, A. S. (2019). Investigation and modelling of asphalt pavement performance in cold regions. International Journal of Pavement Engineering, 20(8), 986–997. doi:10.1080/10298436.2017.1373391.

[48] Zeiada, W., Dabous, S. A., Hamad, K., Al-Ruzouq, R., & Khalil, M. A. (2020). Machine Learning for Pavement Performance Modelling in Warm Climate Regions. Arabian Journal for Science and Engineering, 45(5), 4091–4109. doi:10.1007/s13369-020-04398-6.

[49] Mirou, S. M., Elawady, A. T., Ashour, A. G., Zeiada, W., & Abuzwidah, M. (2023). Visibility Prediction through Machine Learning: Exploring the Role of Meteorological Factors. 2023 Advances in Science and Engineering Technology International Conferences, ASET 2023, 1–6. doi:10.1109/ASET56582.2023.10180539.

[50] Dabous, S. A., Hamad, K., Al-Ruzouq, R., Zeiada, W., Omar, M., & Obaid, L. (2022). a Case-Based Reasoning and Random Forest Framework for Selecting Preventive Maintenance of Flexible Pavement Sections. Baltic Journal of Road and Bridge Engineering, 17(2), 107–134. doi:10.7250/bjrbe.2022-17.562.

[51] Hamad, K., Obaid, L., Haridy, S., Zeiada, W., & Al-Khateeb, G. (2023). Factorial design–machine learning approach for predicting incident durations. Computer-Aided Civil and Infrastructure Engineering, 38(5), 660–680. doi:10.1111/mice.12883.

[52] Navid, M. (2018). Multiple Linear Regressions for Predicting Rainfall for Bangladesh. Communications, 6(1), 11. doi:10.11648/j.com.20180601.11.

[53] Alsheyab, M. A., & Khasawneh, M. A. (2024). Statistical Modeling of Asphalt Pavement Surface Friction Based on Aggregate Fineness Modulus and Asphalt Mix Volumetrics. International Journal of Pavement Research and Technology, 17(5), 1093–1111. doi:10.1007/s42947-023-00289-9.

[54] Kang, M., Kim, M., & Lee, J. H. (2010). Analysis of rigid pavement distresses on interstate highway using decision tree algorithms. KSCE Journal of Civil Engineering, 14(2), 123–130. doi:10.1007/s12205-010-0123-7.

[55] Madeh Piryonesi, S., & El-Diraby, T. E. (2021). Using Machine Learning to Examine Impact of Type of Performance Indicator on Flexible Pavement Deterioration Modeling. Journal of Infrastructure Systems, 27(2), 62. doi:10.1061/(asce)is.1943-555x.0000602.

[56] Hamad, K., Obaid, L., Nassif, A. B., Abu Dabous, S., Al-Ruzouq, R., & Zeiada, W. (2023). Comprehensive evaluation of multiple machine learning classifiers for predicting freeway incident duration. Innovative Infrastructure Solutions, 8(6), 177. doi:10.1007/s41062-023-01138-1.

[57] Babagoli, R., & Rezaei, M. (2022). Development of prediction models for moisture susceptibility of asphalt mixture containing combined SBR, waste CR and ASA using support vector regression and artificial neural network methods. Construction and Building Materials, 322, 126430. doi:10.1016/j.conbuildmat.2022.126430.

[58] Obaid, L., Hamad, K., Khalil, M. A., & Nassif, A. B. (2024). Effect of feature optimization on performance of machine learning models for predicting traffic incident duration. Engineering Applications of Artificial Intelligence, 131, 107845. doi:10.1016/j.engappai.2024.107845.

[59] Molavi Nojumi, M., Huang, Y., Hashemian, L., & Bayat, A. (2022). Application of Machine Learning for Temperature Prediction in a Test Road in Alberta. International Journal of Pavement Research and Technology, 15(2), 303–319. doi:10.1007/s42947-021-00023-3.

[60] Justo-Silva, R., Ferreira, A., & Flintsch, G. (2021). Review on machine learning techniques for developing pavement performance prediction models. Sustainability (Switzerland), 13(9), 5248. doi:10.3390/su13095248.

[61] Sadat Hosseini, A., Hajikarimi, P., Gandomi, M., Moghadas Nejad, F., & Gandomi, A. H. (2021). Optimized machine learning approaches for the prediction of viscoelastic behavior of modified asphalt binders. Construction and Building Materials, 299(January), 124264. doi:10.1016/j.conbuildmat.2021.124264.

[62] Luo, Z., & Li, S. (2023). An interpretable prediction model for pavement performance prediction based on XGBoost and SHAP. Proc. SPIE, March 2023, 55. doi:10.1117/12.2671361.

[63] Nhat-Duc, H., & Van-Duc, T. (2023). Computer Vision-Based Severity Classification of Asphalt Pavement Raveling Using Advanced Gradient Boosting Machines and Lightweight Texture Descriptors. Iranian Journal of Science and Technology - Transactions of Civil Engineering, 47(6), 4059–4073. doi:10.1007/s40996-023-01138-2.

[64] Pei, L., Yu, T., Xu, L., Li, W., & Han, Y. (2022). Prediction of Decay of Pavement Quality or Performance Index Based on Light Gradient Boost Machine. Advances in Intelligent Automation and Soft Computing. IASC 2021, Lecture Notes on Data Engineering and Communications Technologies, 80, Springer, Cham, Switzerland. doi:10.1007/978-3-030-81007-8_135.

[65] Heidarabadizadeh, N., Ghanizadeh, A. R., & Behnood, A. (2021). Prediction of the resilient modulus of non-cohesive subgrade soils and unbound subbase materials using a hybrid support vector machine method and colliding bodies optimization algorithm. Construction and Building Materials, 275, 122140. doi:10.1016/j.conbuildmat.2020.122140.

[66] Huang, Y., Molavi Nojumi, M., Hashemian, L., & Bayat, A. (2023). Evaluation of a Machine Learning Approach for Temperature Prediction in Pavement Base and Subgrade Layers in Alberta, Canada. Journal of Transportation Engineering, Part B: Pavements, 149(1), 1–12. doi:10.1061/jpeodx.pveng-1010.

[67] Deng, Y., & Shi, X. (2022). An Accurate, Reproducible and Robust Model to Predict the Rutting of Asphalt Pavement: Neural Networks Coupled with Particle Swarm Optimization. IEEE Transactions on Intelligent Transportation Systems, 23(11), 22063–22072. doi:10.1109/TITS.2022.3149268.

[68] Nassif, A. B., Elnagar, A., Shahin, I., & Henno, S. (2021). Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities. Applied Soft Computing, 98, 106836. doi:10.1016/j.asoc.2020.106836.

[69] Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech Recognition Using Deep Neural Networks: A Systematic Review. IEEE Access, 7(February), 19143–19165. doi:10.1109/ACCESS.2019.2896880.

[70] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2323. doi:10.1109/5.726791.

[71] Li, H., Peng, W., Adumene, S., & Yazdi, M. (2023). An Improved LeNet-5 Convolutional Neural Network Supporting Condition-Based Maintenance and Fault Diagnosis of Bearings. Intelligent Reliability and Maintainability of Energy Infrastructure Assets. Studies in Systems, Decision and Control, 473, Springer, Cham, Switzerland. doi:10.1007/978-3-031-29962-9_4.

[72] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90. doi:10.1145/3065386.

[73] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR 2015), 7-9 May, 2015, San Diego, United States.

[74] Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1–9. doi:10.1109/cvpr.2015.7298594.

[75] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 90. doi:10.1109/cvpr.2016.90.

[76] Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 634. doi:10.1109/cvpr.2017.634.

[77] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 243. doi:10.1109/cvpr.2017.243.

[78] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning, 9-15 June, 2019, California, United States.

[79] Denny Prabowo, Y., Warnars, H. L. H. S., Budiharto, W., Kistijantoro, A. I., Heryadi, Y., & Lukas. (2018). LSTM and Simple Rnn Comparison In The Problem Of Sequence To Sequence On Conversation Data Using Bahasa Indonesia. 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), 51–56. doi:10.1109/inapr.2018.8627029.

[80] Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. Neural Networks, 1(4), 339–356. doi:10.1016/0893-6080(88)90007-X.

[81] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[82] Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation, 1409(1), 103–111. doi:10.3115/v1/w14-4012.

[83] Ali, A., & Milad, A. (2023). Application of Machine Learning Techniques for Asphalt Pavement Performance Prediction. Journal of Pure & Applied Sciences, 22(3), 35–40. doi:10.51984/jopas.v22i3.2733.