Proceedings of the Canadian Society for Mechanical Engineering International Congress
32nd Annual Conference of the Computational Fluid Dynamics Society of Canada
Canadian Society of Rheology Symposium
CSME/CFDSC/CSR 2025
May 25–28, 2025, Montréal, Québec, Canada

TEMPORAL KOLMOGOROV-ARNOLD NETWORKS FOR CONTROL-ORIENTED MODELING OF HYDROGEN/DIESEL DUAL-FUEL ENGINES

Hossein Mehnatkesh[1*], David Gordon[1], Charles Robert Koch[1]

[1]Mechanical Engineering Department, University of Alberta, Edmonton, Canada T6G 1H9
[*]mehnatke@ualberta.ca

***Abstract*—**Hydrogen/Diesel Dual-Fuel (HDDF) engines are being investigated as a promising solution for reducing carbon dioxide and providing a transition to zero carbon fuels. However, modeling these engines presents challenges due to their nonlinear dynamics and complex interactions among the two fuels. Computational fluid dynamics modeling is often inaccurate across various operating conditions, conventional data-driven approaches frequently face issues related to model size and lack of interpretability, and based on the complexity of the system, there is a lack of accurate physics-based modeling. This study explores Temporal Kolmogorov-Arnold Networks (T-KANs) as an alternative to conventional Machine Learning (ML) models to capture the behavior of HDDF engines. T-KANs offer a structured framework for function approximation that can efficiently learn the underlying dependencies with historical data while ensuring interpretability. A T-KAN model is developed and trained using experimental HDDF engine data, demonstrating its ability to predict the performance metrics of indicated mean effective pressure and oxides of nitrogen emissions. A comparative analysis of seven different models of well-known ML methods highlights the advantages of T-KANs in terms of accuracy, generalization, and computational efficiency. Given its computational efficiency and a coefficient of determination of 0.956, the T-KAN network with a 10-lookback period is suitable for a model-based controller. This configuration effectively utilizes historical data while functioning without reliance on sensor feedback.

***Keywords-component*—temporal Kolmogorov-Arnold network; deep neural network; hydrogen/diesel dual-fuel engine; emissions modeling; time series prediction**

I. INTRODUCTION

Internal combustion engines play a crucial role in our daily lives, especially in the transportation and agriculture sectors. The depletion of petroleum resources and environmental issues drive a growing shift towards alternative energy sources. Hydrogen (H_2) is one of the promising solutions among renewable energy resources [1]. Hydrogen-Diesel Dual-Fuel (HDDF) engines are gaining attention because hydrogen-enriched air in a diesel engine leads to reduced pollution and improved performance [2]. Using hydrogen as a dual fuel alongside diesel can leverage the low auto-ignition temperature of diesel for pilot ignition while benefiting from the clean-burning properties of hydrogen, which could reduce particulate matter and carbon emissions [3]. Optimizing the combustion process in these systems is challenging due to the complex interactions between fuel injection timing, the differing properties of hydrogen and diesel fuels, and higher combustion temperatures. These higher temperatures increase Nitrogen Oxides (NO_x) emissions [4], which can emphasize the need for accurate transient emission models for effective model-based control. By modeling emissions and the Indicated Mean Effective Pressure (IMEP) and embedding the models in the controller, sudden increases in emissions and cylinder pressure can be prevented. Additionally, these models can contribute to Hardware-In-the-Loop (HIL) setups for engine optimization and calibration [5], [6].

Traditional physics-based models, based on thermodynamic principles and phenomenological models, have been widely used to simulate the behavior of HDDF engines. A detailed validation of the numerical model for HDDF engine operation has been conducted in [7]. While they offer valuable insight into engine operation, they can be computationally expensive and challenging to scale across various operating conditions. Three-dimensional Computational Fluid Dynamics (CFD) software has been employed for numerical simulations of direct-injection Compression Ignition (CI) engines. Instead of diesel, a reduced-reaction mechanism for n-heptane was utilized [8]. CFD techniques are widely adopted since they accurately describe the chemical and physical processes occurring in the combustion chamber and provide a detailed overview of the flow field [9]. However, these CFD models may not fully capture the complex and non-linear relationships between varying operating points, and they are computation-

ally expensive, which makes them unsuitable for control-oriented modeling.

Data-driven models, including Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs), have been increasingly utilized to complement or even replace traditional physics-based models in NO_x prediction [10]. These Machine Learning (ML) models are highly effective at learning complex relationships from data. DNNs serve as a fundamental algorithm in deep learning, structured as a multi-layered Artificial Neural Network (ANN) with at least two hidden layers. Unlike DNNs, RNNs are explicitly designed for sequential data processing, making them useful for time-series forecasting [12]. RNNs process past information as inputs for future predictions, enabling them to capture temporal dependencies. However, RNNs suffer from gradient vanishing problems when handling long sequences. To address this limitation, the Long Short-Term Memory (LSTM) model was introduced, which incorporates three gates—input, forget, and output—along with a cell state to retain long-term dependencies [13]. The DNN used for emission prediction achieved a coefficient of determination (R^2) value of 0.9671, whereas LSTM networks outperformed it with an R^2 of 0.9777 [11]. However, the DNN model significantly outperformed LSTM regarding computation speed (0.36 s versus 1381.0 s) [11]. The ability of LSTM to capture long-term time dependencies has made it a popular choice for modeling engine emissions and performance [14], [15]. To simplify LSTM architectures, the Gated Recurrent Unit (GRU) was proposed, which retains essential historical information while discarding non-essential data. By eliminating the cell state and reducing the number of gates, GRUs offer computational efficiency without significantly compromising accuracy compared to LSTM [16]. Which is useful for real-time engine modeling. For instance, Convolutional Neural Network (CNN)-GRU models have been employed to predict emissions in diesel engines, leveraging convolutional feature extraction of CNN and sequential learning capabilities of GRU [17]. Temporal Convolutional Networks (TCNs) provide another alternative to traditional RNN-based models. Unlike LSTMs and GRUs, which rely on sequential processing, TCNs utilize causal convolutions to capture temporal dependencies in parallel. This allows improved handling of long-term dependencies while maintaining computational efficiency. Recent studies have introduced TCN-Nonlinear Autoregressive Exogenous (NARX) architectures, which integrate TCNs with nonlinear auto-regressive models to improve time-series forecasting for dynamic systems [18]. Researchers have explored hybrid and optimized ML architectures to enhance prediction accuracy further and by integrating optimization techniques, network performance can be tailored to specific applications. For example, evolutionary algorithms and hyperparameter tuning have been applied to optimize neural network architecture [19], [20].

Despite the success of ML models, traditional neural networks, such as DNNs and RNNs, do not generalize well for HDDF engines. These networks often overfit the training data, performing well under known conditions but poorly with unseen data. Additionally, the computational complexity of these networks is high, requiring costly hardware for real-time control. Kolmogorov-Arnold Networks (KANs) represent a promising alternative to conventional neural networks for engine modeling. KANs are models based on the Kolmogorov-Arnold representation, a mathematical framework for approximating any continuous function using a series of simple elementary functions [21].

This study uses the Temporal Kolmogorov-Arnold Network (T-KAN) to predict time series data for HDDF engine operation.A control-oriented model of HDDF engine using T-KAN models is described. Specifically, a T-KAN-based framework is proposed to predict combustion characteristics and emissions by utilizing input variables of fuel injection timings. The contributions of this work are as follows:

- Development of a T-KAN control-oriented model to predict IMEP and NO_x in HDDF engines. This model is computationally efficient and independent of sensor feedback, making it suitable for model-based controllers.
- A comparison of the proposed T-KAN model against six other conventional deep learning methods (TCN, TCN+KAN, DNN, LSTM, GRU, and RNN) to quantify the generalization and computational improvements.
- An investigation into the effect of past horizon on the T-KAN accuracy and computation.

II. Experimental Setup

A. Setup

The experimental apparatus consists of a Cummins QSB 4.5-liter Direct-Injection (DI) diesel engine modified to include port-injected hydrogen and emission measurement equipment. This naturally aspirated DI diesel engine does not utilize aftertreatment or Exhaust Gas Recirculation (EGR). Hydrogen is injected into the intake port of cylinder one through the air runner as a port injection. The exhaust line for cylinder one is isolated from the central exhaust system to facilitate emission measurement. Therefore, all reported emissions are derived solely from cylinder one.

To measure NO_x emissions from the engine, a Bosch sensor (ECM P/N: 06-05) is used. The experimental testing integrates a dSPACE MicroAutoBox II (MABX II) prototyping Engine Control Unit (ECU) with other components instead of commercial ECU. The MABX II collects data from all engine sensors, while the dSPACE RapidPro controls all actuators, including the diesel injectors, hydrogen injector, and fuel pump. In-cylinder pressure is measured using a Kistler 6124A piezoelectric pressure transducer mounted between the exhaust valves of cylinder one. The MABX II Field-Programmable Gate Array (FPGA) monitors the in-cylinder pressure sensors and calculates the IMEP [22], [23]. An eddy current dynamometer operating in speed control mode regulates engine speed. Additional details regarding the experimental setup can be found in [10], [24], [25].

B. Dataset

The key variables related to hydrogen and diesel injection timing, which are used to generate pseudorandom data are illustrated in Fig. 1. The HDDF engine was tested using these four control variables (each of which affects the injection process and combustion characteristics):

- $x_1, \mathrm{DDOI_m}$: Duration of the main diesel injection, which determines the total mass of diesel fuel delivered, from 0.17 to 0.5 in ms
- $x_2, \mathrm{DP2M}$: Time interval between the end of the diesel pre-injection and the start of the main diesel injection, which aids in hydrogen ignition, from 0.25 to 1.2 in ms
- $x_3, \mathrm{DSOI_m}$: Start of the main diesel injection, from -15 to 10 in Crank Angle Degrees (CAD)
- $x_4, \mathrm{H2DOI}$: Duration of the hydrogen injection, which regulates the contribution of hydrogen to the combustion process, from 0 to 5 in ms.

In this study, the diesel pre-ignition duration is fixed at $\mathrm{DDOI_p} = 0.17$ ms. To ensure variability and robustness in the dataset, these parameters are varied using a pseudorandom sampling approach. This method enables a comprehensive exploration of potential injection timings while adhering to realistic constraints based on engine operating conditions. The system outputs are considered as $\mathrm{y_1}$ for IMEP (which ranges from 1 to 11 bar) and $\mathrm{y_2}$ for $\mathrm{NO_x}$ emissions (which ranges from 10 to 1860 ppm).

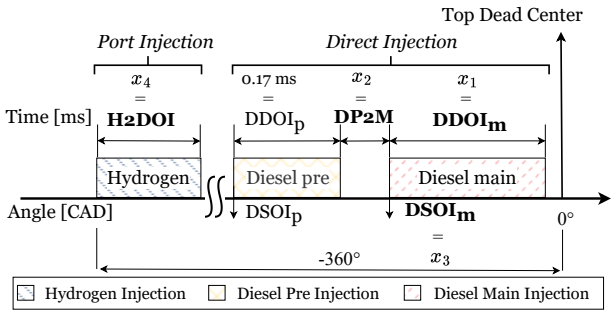

Figure. 1. Variables associated with hydrogen and diesel injection timings in the pseudorandom data collection - $x_1, \mathrm{DDOI_m}$: duration of the main diesel injection. $x_2, \mathrm{DP2M}$: time interval from the end of diesel pre-injection to the beginning of main diesel injection. $x_3, \mathrm{DSOI_m}$: start of the main diesel injection. $x_4, \mathrm{H2DOI}$: duration of the hydrogen injection.

III. Methodology

A. Data Preprocessing

The degree of dependence between the inputs ($X = [x_1, x_2, x_3, x_4]$) and output ($Y = [y_1, y_2]$) is illustrated in Fig. 2, which uses lagged inputs data for each output. The blue line represents the first output, IMEP, while the red line represents the second output, $\mathrm{NO_x}$. Dependence is quantified using Mutual Information (MI), a statistical measure that captures nonlinear relationships between variables. The MI between two random variables X and Y is defined as:

$$\mathrm{MI}(X;\ Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \qquad (1)$$

where, $P(x, y)$ is the joint probability distribution of X and Y and $P(x)$ and $P(y)$ are the marginal probability distributions of X and Y, respectively.

An MI value of zero indicates complete independence, whereas higher values suggest more significant dependencies. Fig. 2 displays the MI values from 1 to 50 lagged inputs data points, highlighting the dependency of each output on the lagged versions of the inputs. To generate each point in Fig. 2, the variable X is changed 50 times (maximum lag), taking into account the delayed data associated with each system input and the number of lags. To analyze the effect of historical information, three lookback periods (n_l = 1, 5, and 10) were selected for comparison, as MI values sharply decreased for both outputs with ten lags. Shorter lookback periods emphasize immediate correlations, while longer periods may capture extended temporal dependencies. Applying these three different lookback periods to the seven ML models (T-KAN, TCN, TCN+KAN, DNN, LSTM, GRU, and RNN) reveals how past data influences the predictive relationships between the inputs and outputs.

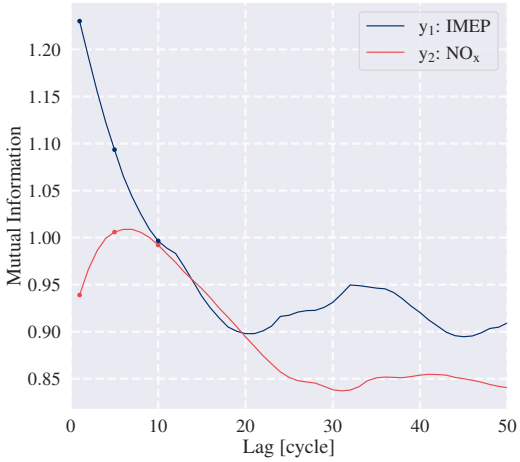

Figure. 2. The degree of dependence between all inputs using lagged data and each output by calculating mutual information, where a zero value indicates independence.

The dataset with 176,800 engine cycles is divided into three subsets: 70% for training, 15% for validation, and 15% for testing. The training set is used to learn model parameters, while the validation set assists in hyperparameter tuning and model selection to prevent overfitting. The test set remains unseen during training and is utilized for the final model evaluation. All input variables are normalized to ensure uniform feature scaling using statistics computed from the training data. Specifically, min-max scaling is applied to transform variables within a fixed range [0,1]. This normalization ensures that all datasets maintain a consistent scale.

IV. Temporal Kolmogorov-Arnold Network

The T-KAN architecture designed for time-series prediction is depicted in Fig. 3. The T-KAN is based on the Kolmogorov-Arnold representation theorem, which states that any multivariate continuous function can be decomposed into a finite number of univariate functions [26]. This allows the network to approximate complex nonlinear relationships while maintaining interpretability efficiently. This network architecture has a single hidden layer comprising ten nodes, an input layer whose size is determined by the lookback period, and two output nodes representing the predicted values for the next step (*t+1*). The T-KAN formulation for the function f, which is characterized as a multivariate continuous function [21], is presented as follows:

$$f(X^t, \ldots, X^{t-k}, \ldots, X^{t-L}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{pq}(x_p) \right) \quad (2)$$

where Φ_q and ϕ_{pq} are continuous functions that can be learned during training. The activation functions (ϕ) between each node (edge) are parameterized as a linear combination of B-splines and basis functions. In this study, the hyperparameters spline order (k) and grid intervals (G) are considered to be 3 and 10, respectively:

$$\phi(x) = w_b b(x) + w_s \sum_{i=0}^{12} c_i B_i(x) \quad (3)$$

where $c_i, w_b,$ and w_s are trainable parameters.

The input layer has a size that is four times the number of lookbacks, which allows the model to incorporate historical dependencies effectively. The lookback period (n_l) is a crucial hyperparameter and is tested for 1, 5, and 10. This variation of lookback is used to show how past information influences predictive performance. The hidden layer captures nonlinear interactions among input features, while the output layer provides output predictions for one step ahead. This network architecture is a compromise between model complexity and computational efficiency. It uses only one hidden layer while ensuring adequate representation of time-series dynamics.

A. Training and Evaluation

All seven ML models are trained using a consistent set of hyperparameters to ensure fair comparisons. The hyperparameters for each model are heuristically tuned on validation data to maximize accuracy. The training lasts for a maximum of 5,000 epochs, with early stopping implemented to stop training if no improvement is observed in the validation loss for 50 consecutive epochs. A learning rate scheduler reduces the learning rate if no improvement occurs after 20 epochs by 80%, starting with an initial learning rate of 0.001. The Adam optimizer efficiently updates model parameters [27]. Mini-batch training is conducted with a batch size of 1,024, which promotes stable convergence while maintaining computational efficiency. Upon completing the training, the final model is evaluated using three key performance metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2.

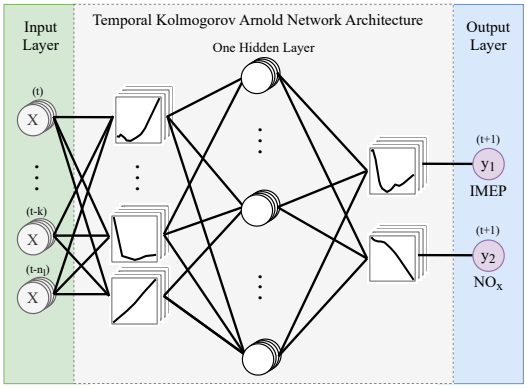

Figure. 3. Temporal Kolmogorov-Arnold network architecture consists of one hidden layer with ten nodes, an input layer consisting of four times the number of lookback nodes, and two output nodes - *t* represents the current time, *t+1* indicates a one-step prediction, and n_l is the maximum lookback period, which varies among 1, 5, and 10.

V. Results and Discussion

The R^2 score across ten runs of the same data but with different random seed numbers for all models is presented in Fig. 4, analyzed against (a) varying lookback values and (b) the average evaluation time per batch on CPU (Intel® Core™ i7-12700K, 3.60 GHz) in seconds, with model size represented by different circle sizes. All models are trained on a GPU (NVIDIA GeForce RTX™ 4090 Ti, 24.0 GB). In Fig. 4 (a), the R^2 values generally improve as the lookback length increases, indicating that longer historical inputs enhance predictive accuracy for all models. However, this improvement in accuracy leads to an increase in model size for T-KAN and DNN, while the sizes of other models remain constant across different lookback values. The rate of improvement decreases from lookback values of 1 to 5 and from 5 to 10, and the sharp decline in MI observed in Fig. 2 suggests that a proper lookback choice would be 10. Some models, such as DNN, exhibit more significant variance across different seed numbers, showing sensitivity to initialization. In contrast, the T-KAN model consistently presents a low error margin across all lookback values. Fig. 4 (b) illustrates the relationship between R^2 and evaluation time. It indicates that among the high-accuracy models T-KAN, TCN, and TCN+KAN, the T-KAN has less evaluation time while having better accuracy. The T-KAN model achieves a high R^2 with relatively low evaluation time, making it a computationally efficient choice. All other models exhibit low evaluation time at different lookback values. The TCN and TCN+KAN models, without an increase in model size, experience a decrease in evaluation time as the lookback values rise from 1 to 5.

The Seven ML models are compared in Fig. 5 using four key metrics: (a) R^2 score, (b) MAE, (c) MSE, and (d) the number of model parameters. The values are displayed in bar plots, with a color matching Fig. 4. The error bars represent the variability in results across 10 runs with different seed numbers. These results are consistent for lookback periods of

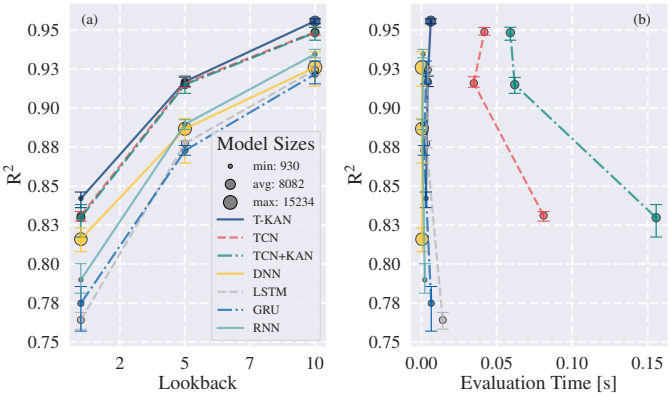

Figure. 4. Coefficient of determination (R^2) for ten runs with different seed numbers, comparing all models against (a) varying lookbacks and (b) the average evaluation time per batch on CPU, along with model size.

1, 5, and 10. In Fig. 5 (a) the R^2 score shows that the T-KAN model achieves the highest predictive performance across all lookback periods, with a score of 0.956 for a lookback of 10. The RNN model also exhibits a higher R^2 score than other recurrent models like GRU and LSTM. MAE values, where lower values indicate better performance are shown in Fig. 5 (b). The T-KAN model demonstrates the smallest MAE at 0.025 for a lookback of 10, while the LSTM model has the highest MAE at 0.054 for a lookback of 1. MSE is shown in Fig. 5 (c), which follows a similar trend, with the T-KAN model having the lowest MSE. The number of parameters in each model is illustrated in Fig. 5 (d). The DNN model has the highest number of parameters at 15,234 for a lookback of 10, while the RNN model is the most compact with only 930 parameters. In this case, since there is no long-term dependency (longer than 10 engine cycle), the RNN performs better with fewer parameters than LSTM and GRU in terms of R^2, MAE, and MSE. The T-KAN model achieves a compromise between accuracy and complexity, having half the parameters of the DNN model (7,980 for a lookback of 10). These results highlight the trade-off between accuracy and model efficiency and show that the T-KAN model is advantageous for applications requiring both accuracy and computational efficiency. Overall, the error bars for the T-KAN model are consistently lower than other models.

The T-KAN model prediction performance with 10 lookbacks is illustrated for the test data in Fig. 6. The blue dots represent the output IMEP, and the red dots represent the output NO_x. Fig. 6 (a) displays the predicted versus actual values, where points closely following the diagonal line indicate accurate predictions. Most predictions align well with the actual values, with minor deviations. The right side the distribution of predicted values, showing a consistent concentration with the actual data distribution (top histogram). In Fig. 6 (b) the residuals (prediction errors) versus actual values provide insight into systematic biases. The residuals have the appearance of a random distribution, suggesting minimal bias. The histogram on the right Fig. 6 (b) displays a

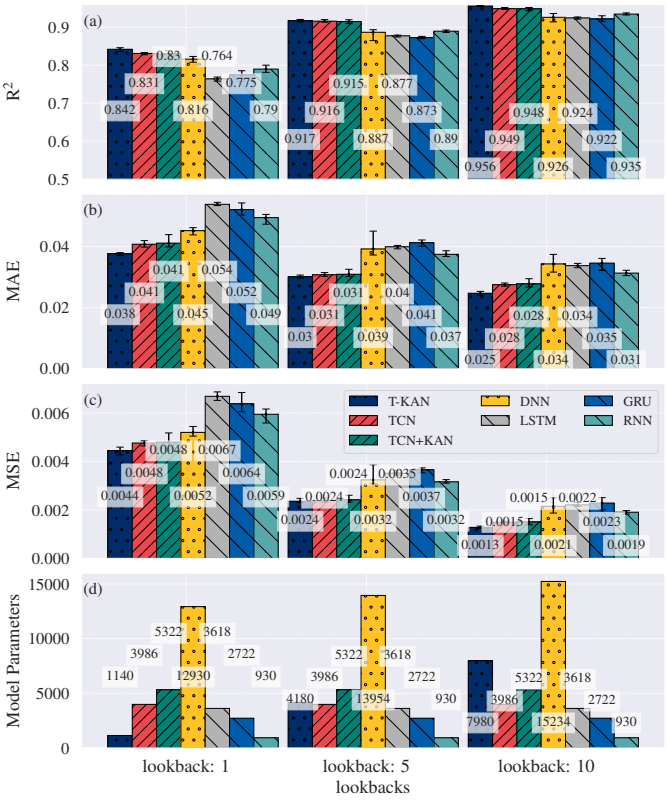

Figure. 5. Different metrics for model comparison are categorized based on various lookbacks for all models: (a) coefficient of determination - R^2, (b) mean absolute error - MAE, (c) mean squared error - MSE, and (d) model parameters.

nearly Gaussian distribution of residuals centered around zero, which indicates that the errors are symmetrically distributed. Together with an MAE of 0.025 for the test data, it confirms that the T-KAN model with 10 lookbacks effectively models the system.

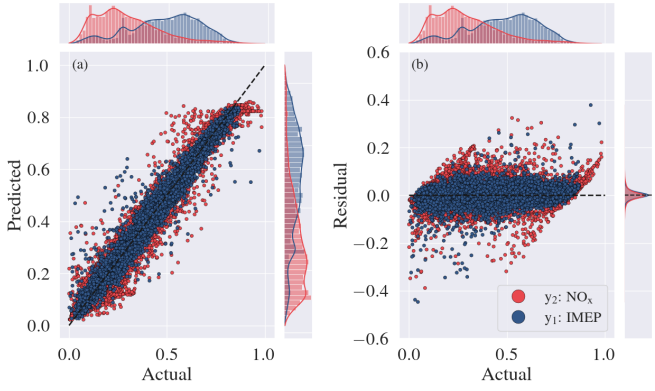

Figure. 6. Final predictions of the temporal Kolmogorov-Arnold network for the test data: (a) predicted versus actual values, (b) residuals versus actual values, with histograms on the sides

The practical application of this study is developing a more efficient model for real-time engine control systems. One key

aspect is reducing model size through the KAN model, which decreases computational complexity. This modification makes the model more suitable for real-time applications where fast prediction over a short horizon is essential. By optimizing the trade-off between model accuracy and computational cost, this approach can contribute to more efficient engine control strategies, ultimately improving performance and responsiveness under dynamic operating conditions.

VI. Conclusions and Future Work

To investigate the control-oriented modeling of IMEP and NO_x with hydrogen and diesel injection timings a data driven approach was used. Seven deep learning models were trained using various lookback periods to evaluate the prediction performance of different modeling approaches. The degree of dependence between input and output variables was quantified using MI, revealing that longer lookback periods generally improved predictive accuracy by capturing extended temporal dependencies but required higher computational effort. The T-KAN model using a single hidden layer structure to approximate nonlinear relationships was used. The results highlighted the trade-offs between model complexity, predictive accuracy, and computational cost. The result for 10 as lookback was R^2 0.956, MAE 0.025, and MSE 0.0013 for this dataset. Comparison of model parameters and computation emphasized the importance of balancing model size and efficiency over the seven ML models tested. The findings highlight the importance of lookback selection, feature dependency analysis, and model complexity trade-offs in predictive modeling. Future work will concentrate on refining model architectures with hyperparameter optimization. Computational efficiency will be enhanced by replacing B-splines with simpler functions for the KAN model, making these more computationally efficient for real-time applications in engine control systems.

References

[1] H. Köse and M. Ciniviz, "An experimental investigation of effect on diesel engine performance and exhaust emissions of addition at dual fuel mode of hydrogen," Fuel Processing Technology, vol. 114, pp. 26–34, Oct. 2013.

[2] N. Saravanan and G. Nagarajan, "An experimental investigation of hydrogen-enriched air induction in a diesel engine system," International Journal of Hydrogen Energy, vol. 33, no. 6, pp. 1769–1775, Mar. 2008.

[3] P. Dimitriou, M. Kumar, T. Tsujimura, and Y. Suzuki, "Combustion and emission characteristics of a hydrogen-diesel dual-fuel engine," International Journal of Hydrogen Energy, vol. 43, no. 29, pp. 13 605–13 617, Jul. 2018.

[4] P. Dimitriou and T. Tsujimura, "A review of hydrogen as a compression ignition engine fuel," International Journal of Hydrogen Energy, vol. 42, no. 38, pp. 24 470–24 486, Sep. 2017.

[5] M. Aliramezani, C. R. Koch, and M. Shahbakhti, "Modeling, diagnostics, optimization, and control of internal combustion engines via modern machine learning techniques: A review and future directions," Progress in Energy and Combustion Science, vol. 88, p. 100967, Jan. 2022.

[6] V. Ahire, M. Shewale, and A. Razban, "A review of the state-of-the-art emission control strategies in modern diesel engines," Archives of Computational Methods in Engineering, vol. 28, no. 7, pp. 4897–4915, Mar. 2021.

[7] C. Ramsay, K. R. Dinesh, W. Fairney, and N. Vaughan, "A numerical study on the effects of constant volume combustion phase on performance and emissions characteristics of a diesel-hydrogen dual-fuel engine," International Journal of Hydrogen Energy, vol. 45, no. 56, pp. 32 598–32 618, Nov. 2020.

[8] A. I. Jabbr and U. O. Koylu, "Influence of operating parameters on performance and emissions for a compression-ignition engine fueled by hydrogen/diesel mixtures," International Journal of Hydrogen Energy, vol. 44, no. 26, pp. 13 964–13 973, May 2019.

[9] M. C. Cameretti, R. De Robbio, E. Mancaruso, and M. Palomba, "CFD study of dual fuel combustion in a research diesel engine fueled by hydrogen," Energies, vol. 15, no. 15, p. 5521, Jul. 2022.

[10] S. Shahpouri, D. Gordon, M. Shahbakhti, and C. R. Koch, "Transient NO_x emission modeling of a hydrogen-diesel engine using hybrid machine learning methods," International Journal of Engine Research, vol. 25, no. 12, pp. 2249–2266, Sep. 2024.

[11] S. Shin, Y. Lee, J. Park, M. Kim, S. Lee, and K. Min, "Predicting transient diesel engine NO_x emissions using time-series data preprocessing with deep-learning models," Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, vol. 235, no. 12, pp. 3170–3184, Mar. 2021.

[12] G. Kushwaha and S. Saraswati, "Air path identification of turbocharged diesel engine using RNN," in 2015 International Conference on Industrial Instrumentation and Control (ICIC). IEEE, May 2015, pp. 1328–1332.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[14] X. Zhang, H. Li, M. Sekar, M. Elgendi, N. Krishnamoorthy, C. Xia, and D. Priya Matharasi, "Machine learning algorithms for a diesel engine fuelled with biodiesel blends and hydrogen using LSTM networks," Fuel, vol. 333, p. 126292, Feb. 2023.

[15] Y. Yu, Y. Wang, J. Li, M. Fu, A. N. Shah, and C. He, "A novel deep learning approach to predict the instantaneous NO_x emissions from diesel engine," IEEE Access, vol. 9, pp. 11 002–11 013, 2021.

[16] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014.

[17] J. Liao, J. Hu, P. Chen, H. Wu, M. Wang, Y. Shao, and Z. Li, "Prediction of transient emission characteristic from diesel engines based on CNN-GRU model optimized by PSO algorithm," Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, vol. 46, no. 1, pp. 1800–1818, Jan. 2024.

[18] M. Tofigh, A. Kharazmi, D. J. Smith, C. R. Koch, and M. Shahbakhti, "Temporal dilated convolution and nonlinear autoregressive network for predicting solid oxide fuel cell performance," Engineering Applications of Artificial Intelligence, vol. 136, p. 108994, Oct. 2024.

[19] H. Mehnatkesh, S. M. J. Jalali, A. Khosravi, S. Nahavandi, An intelligent driven deep residual learning framework for brain tumor classification using MRI images, Expert Systems with Applications 213 (2023) 119087.

[20] H. Mehnatkesh, A. Alasty, M. Boroushaki, M. H. Khodsiani, M. R. Hasheminasab, and M. J. Kermani, "Estimation of water coverage ratio in low temperature PEM-fuel cell using deep neural network," IEEE Sensors Journal, vol. 20, no. 18, pp. 10 679–10 686, 2020.

[21] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov-arnold networks," 2024.

[22] J. B. Heywood, Internal combustion engine fundamentals. Mcgraw-hill, 1988.

[23] D. Gordon, C. Wouters, M. Wick, F. Xia, B. Lehrheuer, J. Andert, C. R. Koch, and S. Pischinger, "Development and experimental validation of a real-time capable field programmable gate array–based gas exchange model for negative valve overlap," International Journal of Engine Research, vol. 21, no. 3, pp. 421–436, Jul. 2018.

[24] McNally, Jakub T, "Hydrogen-diesel dual fuel combustion characterization for an internal combustion engine," Master's thesis, University of Alberta, 2024.

[25] D. C. Gordon, A. Norouzi, A. Winkler, J. McNally, E. Nuss, D. Abel, M. Shahbakhti, J. Andert, and C. R. Koch, "End-to-end deep neural network based nonlinear model predictive control: experimental implementation on diesel engine emission control," Energies, vol. 15, no. 24, p. 9335, 2022.

[26] A. N. Kolmogorov, "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition," Dokl. Akad. Nauk SSSR, vol. 114, pp. 953–956, 1957.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.