A Machine Learning-Based Digital Twin Model for Pressure Prediction in the Fuel Injection System

Edwin P. Duarte*†, Eduardo K. Viegas*‡, Altair O. Santin*
†Robert Bosch GmbH, Engineering Systems, Curitiba, Brazil
edwin.duarte@bosch.com
‡Secure Systems Research Center, Technology Innovation Institute (TII), Abu Dhabi
eduardo@ssrc.tii.ae
*Graduate Program in Computer Science – PPGIa — Pontificia Universidade Catolica do Parana – PUCPR, Brazil
santin@ppgia.pucpr.br

IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society | 978-1-6654-8025-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/IECON49645.2022.9968945

***Abstract*—Over the last years, the engine calibration task has mostly been conducted based on the engineers' knowledge. As a result, considering the complexity of modern engines, finding the most suitable configuration for each situation has become an impractical and expensive task. Apart from causing engines to be produced with inadequate calibration configuration, it can also decrease the lifespan of their components, degrading their efficiency. This paper proposes a machine learning-based digital twin model for pressure prediction in a fuel injection system, split into two steps. First, we extract statistical engine features based on a predefined time window to represent the engine behavior over time. Second, a digital twin implemented through a machine learning model is used to predict pressure levels in the fuel injection system. As a result, the predicted values can be used to assist the engine common rail system module in avoiding undesired engine states. Experiments performed on a new dataset, built over a real diesel-based engine, consisting of 208 features and over 1.3 million instances, have shown the feasibility of our proposal. The proposed scheme can predict in an advance time of 0.1 seconds the pressure levels for a fuel injection system with only 0.057 RMSE. Moreover, it increases its error rate by only 10.6% if a 0.5-second time advance is required.**

***Index Terms*—Digital Twin, Engine, Machine Learning**

I. Introduction

Developing new vehicle engines, especially those based on diesel fuel, is a challenging task that demands manufacturers to meet government legislation while ensuring engine efficiency [1]. To fulfill such a task, manufacturers must properly calibrate the fuel injection system, which is directly related to the exhaust gas system, such as the Common Rail System (CRS). In general terms, the common rail system calibration is based on a proportional–integral–derivative (PID) controller embedded in the Electronic Control Unit (ECU). The controller's goal is to fine-tune the injection system's real pressure based on the ECU's required pressure signal. Engine manufacturers have been performing the calibration task based solely on engineers' expertise following a previously settled guideline [2]. Therefore, the calibration is only mutable during the development phase, and several quality levels must be met before the engine production can start.

As each system calibration comes out uniquely, finding possible failures and their root causes has become increasingly challenging considering the rising number of projects and their complexity [3]. As an example of failure, a not adequately tuned controller may generate *overshoots* and *undershoots* that the CRS might face during its usage. *Overshoots* occur when the real pressure of the system exceeds the required pressure limits, reaching higher than desired levels. *Undershoots* lowers the system pressure beyond its intended operation level. Such situations may significantly decrease the engine lifespan while degrading its efficiency. In such a context, over the last years, several works have been proposed to improve the industry engine development process, wherein approaches based on digital twin (DT) have yielded promising results [4].

A DT aims to accurately reflect the physical object properties (e.g., engine), as represented by their sensors' values. The values depict a series of the physical object conditions, which are then used to create the DT. The DT can then be used to run simulations, investigate performance issues, and generate possible improvements and insights within the digital domain, which can then be applied back to the original physical object [5]. Several approaches have been proposed to create DT, wherein authors typically resort to machine learning (ML) techniques [6].

A behavioral model can be built by evaluating a training dataset composed of vast amounts of the physical object collected sensor values. As a result, the built model, acting as the DT, will be able to portray the data used during the training phase. Unfortunately, the building of a realistic DT training dataset is a challenging task. A realistic DT must collect training data under a variety of conditions. This should include faulty and normal events often not easily achieved during the engine development. The DT can often only portray the relationship between the collected sensor values even if a realistic training dataset is available. Therefore, the way they can be used for improvement purposes is neglected, for example, to improve system calibration in an engine fuel injection.

This paper proposes a DT model based on ML techniques to assist the engine calibration task, implemented through two strategies. First, we extract statistical features based on a predefined time window of the collected engine sensor values. The insight of such an approach is that historical statistical features can be used to represent the engine behavior in detecting future engine failures. Second, the extracted feature values are used as input by an ML model, which acts as the engine DT. The model is used to predict pressure levels in the fuel injection system. The main insight of such an approach is to use a DT to assist the engine calibration reliability while also predicting the system behavior and engine failures so that counteractions can be performed accordingly.

The main paper contributions are as follows:

- A new DT dataset built through the collection of 208 sensor values from real diesel-based engines. It is composed by over 1.3 million of samples, including 57 and 52 thousand *undershoot* and *overshoot* failures respectively.
- A new DT model based on ML techniques for the prediction of pressure levels in the fuel injection system. The proposed scheme can predict in advance of 0.1 seconds the pressure levels of the fuel injection system with only 0.057 of RMSE.

II. Preliminaries

A. Common Rail Injection System Layout

Our work consider a common rail injection system (CRS) layout, as shown in Figure 1, which is commonly used by traditional diesel-based engines, such as the one used by a commercial vehicle with a 2.0 liters diesel engine 4×4 model. In the CRS, the pressure generation and the fuel injection are independent. The pressurization of the fuel takes place in the Common Rail, as the high-pressure pump supplies a continuous flow of diesel to it, ensuring the fuel is under the ideal pressure and ready to be injected.

To enable its proper operation, it is a must that the manufacturer ensures the appropriate calibration of the CRS, which is typically achieved by a proportional–integral–derivative (PID) controller embedded in the Electronic Control Unit (ECU). The main calibration goal is to ensure that the pressure of the injection system reflects the required pressure signal from the ECU. Therefore, CRS operation can be typically described according to three main situations:

- *Normal*. Expected CRS injection system state wherein the system injects the expected fuel amount as computed by the PID controller. The CRS operates at the proper pressure, resulting in no waste of fuel and no degradation of the system components' lifespan.
- *Overshoot*. Failure state wherein the CRS injection system over-injects fuel. Fuel injection system pressure increases, causing fuel waste and degradation of the engine components' lifespan.
- *Undershoot*. Failure state wherein the CRS injection system under-injects fuel. Fuel injection system pressure decreases beyond expected, affecting the engine reliability and driving comfort.

B. Digital Twin

A digital twin (DT) aims to reproduce the behavior of a complex physical product using a probabilistic function, which

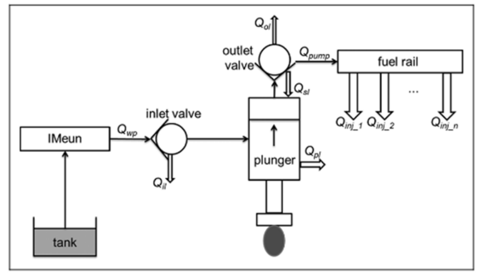


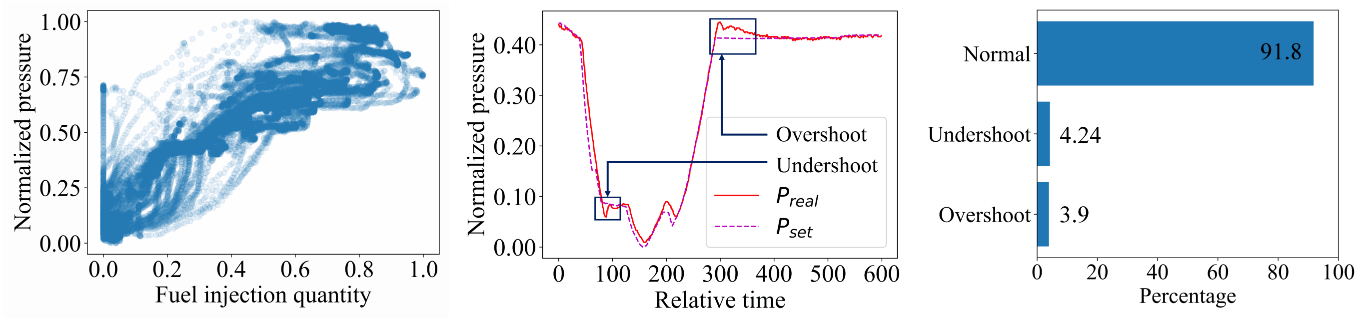
Fig. 1: Common rail injection system (CRS) layout is considered in our work. The layout depicts a CRS in a real vehicle as used in a commercial vehicle with a 2.0 liters diesel engine.

is used to mirror the behavior of its corresponding twin [7]. Its main goal is to act as a digital copy of a physical asset. As a result, a DT can give assessments of how a system will perform under production. Thus, it can be used to identify and easily pave the way to efficiency improvements. A DT is typically utilized at the initial phases of development and design. It enables the precise reproduction of how specific systems and subsystems will perform in a set of predefined circumstances.

Several approaches have been proposed for the creation of a DT in a variety of fields, including industry, medical, and even for monitoring purposes [8], [9]. Authors generally resort to machine learning (ML) techniques, yielding promising reported results [10]. In such a case, an ML model is built through a computationally expensive model training process, which evaluates the data available in a training dataset. The dataset must be composed of huge amounts of samples from the environment, given that the ML model will be built accordingly. As a result, the building of a realistic DT through ML means becomes a challenging task. This is because the data collected from the physical object sensors often cannot correctly depict the behavior of the to-be-digitalized object given that it must be monitored for a long time under a variety of environmental conditions. In such a case, the built ML model, acting as the DT, may present high accuracy in physical object representations. However, at the same time, it does not provide the expected level of realism to be used as a DT, e.g., to evaluate the physical object behavior under production or even pave the way to efficiency improvements.

III. Related Works

Many researchers have proposed different approaches for studying and implementing the digital twin concept. Ghanishtha B. *et al.* [4] proposed the main three stages of DT development in vehicle as *archetype modeling*, *virtual sensors modeling* and *parameter update*. Focusing specifically on the first stage, which contemplates the activities of standalone model, many researches can be found in the literature on applying artificial intelligence concepts to engine calibration and fault detection [11]. For instance, Airamadan A. *et al.* [12] showcased the strength of machine learning models in imitat-



(a) *Fuel injection quantity* $vs.$ P_{set} (b) *Undershoot* and *Overshoot* failures (c) DTDED states in percentage

Fig. 2: Data distribution of the *Digital-Twin Diesel Engine Dataset* (DTDED).

ing the operation of an advanced engine concept - the gasoline compression ignition at low loads.

Similarly, M. Hinrichs [13] used a traditional model-based approach to detect faults of a heavy-duty diesel engine based on the injected fuel. The author presented that despite achieving satisfactory results, there are limitations in such traditional models. For example, the results can be biased considering the difficulty of applying the model in different scenarios of engine functionality. A. di Gaeta *et al.* [14] proposed a model-based gain scheduling approach for controlling the common-rail system for gasoline direct injection engines, aiming for best performance concerning emissions, fuel economy, driveability, and diagnostics. J. Zheng *et al.* [15] studied the possibility of using a classification algorithm to diagnose faults of the injector used in a diesel engine CRS, with promising results but not focusing on the calibration process itself. W. Chatlatanagulchai *et al.* [16] proposed a quantitative-feedback theory controller designed and applied to a CRS of a diesel-dual-fuel engine. The resulting controller is robust to model uncertainties and external disturbances. A quantitative measure of the achieved robustness is also provided and confirmed in simulation via experiments but without considering the effects of cylinder interactions. As a result, the possibilities of applying machine learning and digital twin concepts into engine development stages are evidenced. Yet, despite the promising prediction capabilities of the presented works, these models do not support the calibration process and do not anticipate possible faults.

IV. Problem Statement

This section further investigates the calibration challenges of a CRS from a real diesel engine (see Section II-A). More specifically, we first introduce the dataset built on our work, and then the failures evidenced due to a miss-calibration on the CRS.

A. Dataset

Current DT datasets used in the literature often do not depict the complexities of the DT domain. This is because as a DT aim at building a digitalized copy of a physical object, the representation of a realistic behavior of the to-be-digitalized object requires the data collection to be made for a significant time, ensuring that even failures can be adequately collected. In contrast, in general, current datasets in the literature either generate the data in a simulated environment or monitor the to-be-digitalized object in unrealistic settings, thus, without data related to failure states.

This paper presents a new DT dataset namely *Digital-Twin Diesel Engine Dataset* (DTDED). One of the first of its kind, the dataset depicts the data collected from a real diesel-based engine. More specifically, DTDED was built through the data collected in a real commercial vehicle with a 2.0 liters diesel engine 4×4 model, still in the validation phase. The data collection took place from the CAN network (*controller area network*) responsible for managing all the information transmitted in the vehicle, from commands sent from the ECU, to the reading of data from sensors spread across the other vehicle systems.

Data collected in the automotive context follows the Association for the Standardization of Automation and Measurement Systems (ASAM) definitions. Therefore, the data is made available in the Measurement Data Format version 4 (MF4) format, capable of supporting the recording of a high volume of attributes with a high acquisition rate. The data consists of $1,347,340$ instances with 208 attributes each. Sample rates range from 1 second, 100 milliseconds, and 10 milliseconds. The collection of DTDED took place in a total period of 10 minutes of vehicle operation.

In practice, DTDED showcases a miscalibration in a diesel-based engine. The miscalibration occurs due to bad operation by the Common Rail System (CRS), which can result in a *Normal*, or *Overshoot/Undershoot* failure situation (see Section II-A). Undershoot situations are characterized when the fuel injection system's real pressure (P_{real}) is at least 5% less than the fuel injection system's desired pressure P_{set}. Overshoot situations are characterized when the P_{real} is at least 5% higher than the engine P_{set}. The fuel injection system P_{set} is computed according to Equation 4.

$$K_p = a * K_{pcrit} \tag{1}$$

$$K_i = b * K_p * 2\pi * f_{crit} \tag{2}$$

$$K_d = K_{pcrit} * (c - a) \quad (3)$$

$$P_{real} = P_{set} + (K_p + K_i + K_d) \quad (4)$$

Where:

K_p = proportional gain
a = factor in avoiding $K_p critical$
K_{pcrit} = proportional gain value that excites the system at a resonant frequency
K_i = integral gain
b = factor in maintaining correlation-ship between gains
f_{crit} = frequency equal or close to natural frequency
K_d = derivative gain
c = factor in maintaining correlation-ship between gains
P_{real} = real fuel injection system pressure
P_{set} = desired fuel injection system pressure

B. *How frequent are failures observed in DTDED dataset?*

In this section, we investigate the data distribution of DTDED dataset. More specifically, we first evaluate how the amount of fuel injection from the engine relates to the pressure of the fuel injection system (P_{set}, Eq. 4), as shown in Figure 2a. In practice, the engine fuel injection amount is highly correlated with the fuel injection system pressure, reaching a correlation value of 0.94. However, it is possible to note a significant dispersion in the collected values caused by the collection of real values from a real diesel-based engine.

We further investigate how the failures in DTDED dataset occur. Figure 2b shows a data fraction sample collected from our dataset, showing the occurrence of *Undershoot* and *Overshoot* failures as time passes. It is possible to note that such a failure continues to occur for a period in time, given the time needed for the P_{real} to reach the P_{set}. Figure 2c shows the data distribution of our DTDED dataset according to the considered failure states (see Section II-A). Considering both *Overshoot* and *Undershoot*, failure states account for only 8.14% of the total number of samples.

Such data distribution disparity is expected from real diesel engines, since failures must occur rarely to ensure proper engine operation. Unfortunately, the rarity of failures affects the construction of realistic DTs, since, in general, prior works relies on collecting data over a small time window, or even making use of simulated environments. As a result, the DTDED dataset allows operators to construct realistic DTs, given that the engine faults are properly represented in the original dataset.

V. A Machine Learning Based Digital Twin Model for Engine Calibration

This paper proposes a new machine learning-based digital twin model to assist operators with engine calibration and predicting the pressure in the fuel injection system to help the common rail injection system (CRS). The overall proposal is shown in Figure 3.

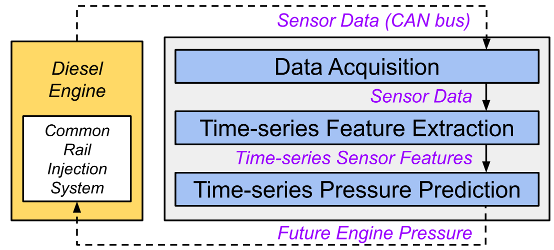


Fig. 3: A digital twin model based on machine learning to predict the pressure in the fuel injection system, helping in the common rail injection system (CRS).

The proposal considers a diesel engine with a CRS module that manages the fuel injection system pressure. The CRS module uses a digital engine twin that continuously evaluates the engine sensors' data to predict pressure level in the fuel injection system. The main insight of such an approach is that our proposal, based on a digital twin implementation, can predict pressure failure states in the fuel injection system (*Overshoot* and *Undershoot*) to assist the CRS module in proactively taking countermeasures. As a result, based on the digital engine twin, the proposal enables the CRS module to avoid situations wherein the fuel injection system pressure may affect the engine's reliability.

A. *Engine Digital Twin*

The proposal assumes that a diesel engine should not exceed or fall below a preset pressure in the fuel injection system. The engine manufacturer defines the pressure set-point to ensure the reliability of the engine over time. The CRS module must be calibrated to meet the manufacturer's standards, ensuring that the intended pressure levels in the fuel injection system are met. However, the calibration task requires a lot of engineers' time to achieve such a goal. As a result, CRS modules often go to the production line without proper calibration, generating under- and over-pressures in the fuel injection system over time (see Section IV).

Our proposal aims to integrate a digital twin model to forecast undesired pressure levels in the fuel injection system. The digital twin aims to replicate the diesel engine behavior as time passes, providing the CRS module with an indicator of undesired pressure levels. The operation of our proposed scheme starts with the data collection by a *Data Acquisition* module (Figure 3). The module continuously collects engine sensor data through a CAN (*Controller Area Network*) bus. The collected sensors data values are used as a representation of the diesel engine's current state. The data is used as input by a *Time-series Feature Extraction* module, which aims to compound a feature vector that depicts the historical behavior of the diesel engine. The module builds a feature vector through a sliding window of events rationale (further described in Section VI). As a result, the built feature vector depicts the diesel engine state in a given time window, thus, representing the engine behavior as time passes. The built time-series sensor

TABLE I: Features values for the diesel engine used by our digital twin, after applying the feature selection technique.

Feature Groups	Description	Number of features
Sensors and actuators of high pressure pump	Data collected from sensor and signals sent from ECU to control the high pressure pump	28
Sensors and actuators of CRS	Data collected from sensor and signals sent from ECU to control the Common Rail System	22
Vehicle drivability	Data collected from sensors regarding driving behavior of the running vehicle	9
Environmental conditions	Data from external sensors in the vehicle. (e.g. external temperature, altitude, etc)	4
Injection strategy	Data collected from sensor and signals sent from ECU to control injectors and their strategy. (e.g. moment and duration of fuel injection)	7

values are used as input by a *Time-series Pressure Prediction* module. The module, in turn, acts as the digital engine twin to forecast pressure levels in the fuel injection system. To achieve such a goal, it applies a machine learning model, which predicts, within milliseconds in advance, the pressure levels in the fuel injection system. Finally, the CRS module can use the prediction outcome to take counteractions before the pressure level in the fuel injection system reaches undesired states.

Consequently, the proposed machine learning model for building a diesel engine digital twin enables operators to improve the operation of CRS modules. Nevertheless, CRS calibration can be facilitated during the engine development process since calibration inadequacy can be identified and fixed during CRS operations

VI. Prototype

A proposal prototype was implemented considering the previously described DTDED dataset (see Section IV). A traditional machine learning process is considered for building our proposed digital twin (Figure 3). The DTDED dataset was split into training, validation, and testing datasets, each composed of 40%, 30%, and 30%, respectively, of the original dataset. Each dataset depicts the diesel engine operation in a given time window. The dataset sensor values were normalized through a *min-max* normalization procedure.

Before building our DT model, a linear interpolation procedure was used, given that the dataset was made of 208 feature values collected from a variety of sensors. Each with a sample rate ranging from 1 second (e.g., *fuel temperature*), to 10 milliseconds (e.g., *current amount of injected fuel*). The resulted dataset depicts the engine values in a 10 millisecond interval.

We perform feature selection based on information gain on the collected sensor values. Thus, we select the features with a strong correlation to the current fuel injection system pressure for the building of our DT (*information gain* ≥ 0.8), resulting in 70 selected features, as shown in Table I. The DT *Time-series Pressure Prediction* module was implemented through a linear regression model. The model receives as input a feature vector built by the *Time-series Feature Extraction* module (see Figure 3). The feature vector was built by concatenating the previously selected 70 features considering a 3 sample window.

TABLE II: Regression performance for predicting P_{real}.

Prediction Time (s)	RMSE	R^2	MAE
0.1	0.057	0.949	0.037
0.2	0.058	0.948	0.038
0.3	0.059	0.947	0.039
0.4	0.062	0.941	0.040
0.5	0.063	0.940	0.041

The model and the previously described data preprocessing were implemented through *scikit-learn* API $v.1.1.1$, and *pandas* API $1.4.2$. The model was evaluated through the Root Mean Square Error (RMSE), Adjusted R Square (R^2), and Mean Absolute Error (MAE), as usually made in related works.

VII. Evaluation

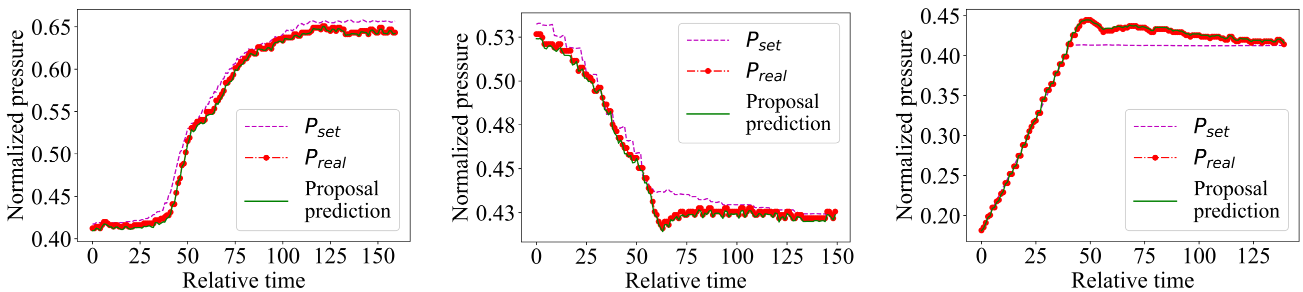
The evaluation aims at answering the following research questions (RQ): (***RQ1***) *How does our proposed DT model work for predicting pressure levels in the fuel injection system?* (***RQ2***) *What is the prediction performance of our model for a longer prediction time for the pressure levels?*

A. A Digital Twin Model

The first experiment aims at answering RQ1 and evaluates the prediction performance of our proposed DT model for pressure levels in the fuel injection system. We consider a DT model that aims to predict the pressure level in the fuel injection system for the engine CRS 0.1 seconds before it occurs. This prediction time was defined based on calibration expert experience. Other prediction times will also be evaluated (see Section VII-B).

Table II shows the proposal error rates for predicting pressure level 0.1 seconds ahead. In such a case, our proposal reached an RMSE of only 0.057, thus, enabling the application of the proposed DT model to assist the CRS module. Figure 4 shows the performance of our proposal in a variety of DTDED dataset settings.

Our proposed scheme was able to provide similar prediction performance when utilized under *Overshoot* and above *Undershoot* to the *Normal* situation. For instance, in an *Overshoot* setting, our proposed scheme was able to properly detect the future undesired pressure level in the fuel injection system,



(a) *Normal* settings (b) *Undershoot* settings (c) *Overshoot* settings

Fig. 4: Proposal performance under different *Digital-Twin Diesel Engine Dataset* (DTDED) settings. The goal is to detect ahead (0.1 seconds) the pressure levels in the fuel injection system (P_{real}).

with a normalized pressure error of only 0.005 (Figure 4c, at ≈ 50 seconds). Similarly, our scheme can detect an *Undershoot* setting with a pressure error of only 0.005 (Figure 4b, at ≈ 60 seconds).

B. Prediction of Pressure Levels in the Fuel Injection System

To answer RQ2, we further investigate how the fuel injection system pressure's prediction time impacts our proposed scheme's prediction performance. Specifically, we vary our model's pressure level prediction time from 0.1 to 0.5 seconds for the fuel injection system. This is because the prediction time for pressure, in the fuel injection system, must be defined according to the expert's needs and may vary according to the used engine configuration. Table II shows the prediction performance of our model according to the future prediction time. It is possible to note that the prediction time directly relates to our proposal measured error rates. For instance, the RMSE is increased by 0.006 ($+10.6\%$) when the future prediction time increases from 0.1 to a 0.5 second setting. As a result, the proposed scheme can be by the CRS module to assist in the pressure management for the fuel injection system even if a higher future pressure time is needed.

VIII. Conclusion and Future Work

Digital twin is the critical technology to fully fusion physical and virtual models. This paper has proposed a digital twin model based on machine learning techniques to assist the vehicle engine development task. The proposed scheme predicts pressure levels in the fuel injection system of a real diesel-based vehicle engine. As a result, it can be used to assist the Common Rail System (CRS) module in preventing unwanted pressure levels in the fuel injection system, avoiding the engine from premature wear. As future works, we plan to extend the proposed dataset to include additional vehicles and incorporate the proposal in an actual engine.

Acknowledgment

This work was partially supported by the CNPq (National Council for Scientific and Technological Development) grant 304990/2021-3.

References

[1] W. Knecht, "Diesel engine development in view of reduced emission standards," *Energy*, vol. 33, no. 2, pp. 264–271, Feb. 2008.

[2] R. J. Lygoe, M. Cary, and P. J. Fleming, "A many-objective optimisation decision-making process applied to automotive diesel engine calibration," in *Lecture Notes in Computer Science*, 2010, pp. 638–646.

[3] W. Xi, Z. Li, Z. Tian, and Z. Duan, "A feature extraction and visualization method for fault detection of marine diesel engines," *Measurement*, vol. 116, pp. 429–437, Feb. 2018.

[4] G. Bhatti, H. Mohan, and R. R. Singh, "Towards the future of smart electric vehicles: Digital twin technology," *Renewable and Sustainable Energy Reviews*, vol. 141, p. 110801, May 2021.

[5] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twins—streamlining simulation-based systems engineering for industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1722–1731, Apr. 2018.

[6] P. Horchulhack, E. K. Viegas, and A. O. Santin, "Toward feasible machine learning model updates in network-based intrusion detection," *Computer Networks*, vol. 202, p. 108618, Jan. 2022.

[7] S. Haag and R. Anderl, "Digital twin – proof of concept," *Manufacturing Letters*, vol. 15, pp. 64–66, 2018, industry 4.0 and Smart Manufacturing.

[8] F. Ramos, E. Viegas, A. Santin, P. Horchulhack, R. R. dos Santos, and A. Espindola, "A machine learning model for detection of docker-based APP overbooking on kubernetes," in *ICC 2021 - IEEE International Conference on Communications*. IEEE, Jun. 2021.

[9] B. B. Bulle, A. O. Santin, E. K. Viegas, and R. R. dos Santos, "A host-based intrusion detection model based on OS diversity for SCADA," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Oct. 2020.

[10] E. K. Viegas, A. O. Santin, V. V. Cogo, and V. Abreu, "A reliable semi-supervised intrusion detection model: One year of network traffic anomalies," in *IEEE Int. Conf. on Comm. (ICC)*, 2020, pp. 1–6.

[11] V. Abreu, A. O. Santin, E. K. Viegas, and V. V. Cogo, "Identity and access management for IoT in smart grid," in *Advanced Inf. Net. and App.* Springer International Publishing, 2020, pp. 1215–1226.

[12] A. S. Airamadan, Z. A. Ibrahim, B. Mohan, and J. Badra, "Machine learning model for spark-assisted gasoline compression ignition engine," in *SAE Technical Paper Series*. SAE International, Mar. 2022.

[13] M. Hinrichs, "Online fault detection of a heavy duty diesel engine with model-based methods," 2021. [Online]. Available: https://tuprints.ulb.tu-darmstadt.de/id/eprint/18886

[14] A. di Gaeta, U. Montanaro, G. Fiengo, A. Palladino, and V. Giglio, "A model-based gain scheduling approach for controlling the common-rail system for GDI engines," *International Journal of Control*, vol. 85, no. 4, pp. 419–436, Apr. 2012.

[15] J. Zheng, M. Zhao, W. Yang, J. Sun, and C. Liu, "Fault diagnosis of injector in high pressure common rail system of diesel engine," in *2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)*. IEEE, Oct. 2020.

[16] W. Chatlatanagulchai, K. Yaovaja, S. Rhienprayoon, and K. Wannatong, "Gain-scheduling integrator-augmented sliding-mode control of common-rail pressure in diesel-dual-fuel engine," in *SAE Technical Paper Series*. SAE International, May 2010.