Clemson University
TigerPrints

All Dissertations | Dissertations

8-2024

Hybrid Physics-Infused Machine Learning Framework For Fault Diagnostics and Prognostics in Cyber-Physical System Of Diesel Engine

Shubhendu Kumar Singh
shubhes@clemson.edu

Follow this and additional works at: https://open.clemson.edu/all_dissertations

 Part of the Navigation, Guidance, Control, and Dynamics Commons, and the Other Mechanical Engineering Commons

Recommended Citation

Singh, Shubhendu Kumar, "Hybrid Physics-Infused Machine Learning Framework For Fault Diagnostics and Prognostics in Cyber-Physical System Of Diesel Engine" (2024). *All Dissertations*. 3732.
https://open.clemson.edu/all_dissertations/3732

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

Hybrid Physics-Infused Machine Learning Framework For Fault Diagnostics and Prognostics in Cyber-Physical System Of Diesel Engine

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Automotive Engineering

---

by
Shubhendu Kumar Singh
August 2024

Accepted by:
Dr. Rahul Rai, Committee Chair
Dr. Venkat Krovi
Dr. Srikanth Pilla
Dr. Zhen Li

Abstract

Fault diagnosis is required to ensure the safe operation of various equipment and enables real-time monitoring of associated components. As a result, the demand for new cognitive fault diagnosis algorithms is the need of the hour. Existing deep learning algorithms can detect, classify, and isolate faults. Still, most depend solely on data availability and do not incorporate the system's underlying physics into their prediction. Therefore, the results generated by these fault-detecting algorithms sometimes need to make more sense and deliver when tested in actual operating conditions.

Similar to diagnosis, the fault prognosis of diesel engines is paramount in numerous industries. Unexpected diesel engine failures can lead to significant operational disruptions and maintenance costs. Accurate Remaining Useful Life (RUL) estimation is crucial for proactive maintenance. Fault prognosis methods are essential for accurately estimating RUL. While current deep learning algorithms excel at identifying patterns in data, they often rely solely on data availability, neglecting to integrate the fundamental physics of the system into their predictions. Consequently, these algorithms fall short when subjected to real-world operational challenges.

The presented dissertation addresses the aforementioned issues. The specific contributions of this dissertation are: (1) Deploying a novel physics-infused one-dimensional Convolutional Neural Network (1D-CNN) based deep learning framework for diesel engine fault detection. (2) A hybrid ensemble learning framework that

integrates physics principles into a 1D-CNN-based ensemble learning model to detect faults in diesel engines. (3) A physics-informed 1D CNN-based prognostics framework underpinned by a 1D CNN to estimate the RUL of the fuel injector.

Dedication

To my wonderful parents, who have selflessly supported me in pursuing my dreams.

Thank you!

Acknowledgments

Throughout the course of composing this dissertation, I have been the fortunate recipient of extensive aid and support. Above all, I extend my profound appreciation to my esteemed advisor, Dr. Rahul Rai, whose guidance and unwavering encouragement sustained me throughout my doctoral studies. His vast expertise and wealth of experience served as a beacon of inspiration for me, both academically and personally. I am grateful to him for his patience and invaluable assistance, particularly during uncertainty and self-doubt. Furthermore, I thank Dr. Venkat Krovi, Dr. Srikant Pilla, and Dr. Zhen Li for their valuable technical assistance, which significantly impacted the development of my research techniques and enhanced the caliber of my dissertation. I also extend my gratitude to Dr. Benjamin Lawler and his team from Advanced Combustion and Renewable Fuels Lab for helping me with the experimental data collection, which was crucial in the successful completion of the dissertation research work.

I am grateful for the generous assistance provided by my colleagues at the Geometric Reasoning and Artificial Intelligence Lab (GRAIL). It was a pleasure collaborating with them, and I extend my appreciation for their valuable contributions in enhancing my research. Moreover, I would like to express my gratitude to all my friends for being there for me during the good and challenging times and for their encouragement and support.

In conclusion, I want to express my gratitude to my family, and especially my parents, for their invaluable guidance, unwavering support, and motivation during my research journey. Their constant presence and encouragement have been instrumental in my success, and I dedicate this dissertation to them as a gesture of gratitude.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation for Fault Diagnosis

Diesel engines are ubiquitous as mobile drives, power generators, and automobile engines. The vast canvas of their application extends across industries and happens to be the driving force behind a myriad of things we come by daily. Due to the high thermal efficiency, low operating cost, and high reliability, diesel engines find various applications in automobiles, locomotives, construction, mining, and marine vessels. In summary, diesel engines are still prevalent and form the backbone of different industrial sectors, especially the automotive ones. Extreme environments and harsh working conditions render these complex machines prone to various faults. Among various faults, the fuel injector system's faults are more prominent. Clogging, coil winding, electro-mechanical defects are a few of the leading causes of the anomalous behavior of an injector system. These abnormalities in injectors may lead to severe consequences. For instance, clogging in an injector nozzle may cause irregular fuel supply, leading to an engine misfire. Misfiring can cause pre-ignition, overheating, and reduced fuel efficiency. Wear and tear of injector coil windings, and

other electro-mechanical problems result in overheating, leakage, and malfunctioning injectors. An aberration in injector behavior can inflict serious damage to the entire engine system thus eventually degrading the vehicle performance.

At present, the automotive industry is undergoing a paradigm shift. A significant amount of effort is being focussed on developing more complex systems to suffice the need for more fuel and emission-efficient vehicles without compromising on drivability and performance. With modern-day chip technology and robust computing power, more mechanical systems are being replaced by sophisticated electronic versions. The technical advances in the automobile sector put a significant responsibility on the after-market services for the efficient continual operation of modern-day vehicles. The after-market services also determine the Original Equipment Manufacturers' (OEMs) profit margin. Consequently, detection, classification, and localization of faults are critical for smooth running, optimized performance, and safe operations of complex machines, culminating in higher productivity and improved process efficiency. Timely fault detection has its share of benefits, including longer operating life, lower maintenance cost, and reduced downtime. The automotive industry, in particular, relies heavily on robust fault detection and isolation techniques. One of the critical components of any automotive is its engine. Accurate diagnostics for engine fault is mandatory to maintain reasonable high up-time and reduce servicing costs. Besides, environment considerate regulations such as On-Board Diagnostic-II (OBD-II), as adopted by the United States Environmental Protection Agency (US EPA), specifies some stringent conditions for performance assessment of engine diagnostics tools. Legislations render the domain of engine-fault diagnosis one of the most complicated and sought after. Also, an efficient engine diagnosis method ensures easy repairability and better protection. Existing, On-Board Diagnostic (OBD) tools are doing their job just fine when it comes to aiding the technical team with

troubleshooting the faulty components of an automotive diesel engine. However, the rapidly evolving technical complexities of the new-age automobile bring several challenges to the extant diagnostic methods. The increased complexity of the system renders the fault diagnostics quite time-consuming, resulting in increased servicing and after-sales costs, which ultimately transpires into immense customer dissatisfaction. In light of the aforementioned arguments, it becomes essential to promptly conduct a proper diagnosis and health check of the diesel engine. Besides, there is a need for these diagnostic methods to be intelligent and easy-to-use so that to enable autonomous troubleshooting of automotive systems.

1.2 Motivation for Fault Prognosis

Prognostics, a branch of predictive maintenance, has garnered significant attention in recent years due to its potential to revolutionize how industries manage their assets. Traditional maintenance practices often rely on predetermined schedules or reactive approaches, leading to unnecessary maintenance costs or unplanned downtime. Prognostics, in contrast, seek to predict when a component or system will fail and how much useful life remains. This predictive capability allows for more efficient and cost-effective maintenance strategies, such as condition-based maintenance, where maintenance actions are performed only when needed, based on real-time health assessments. The idea of remaining useful life (RUL) is central to prognostics. RUL represents the time remaining until a system or component is expected to fail or reach a predefined degradation threshold. Accurate RUL estimation can enable proactive decision-making, optimizing maintenance schedules and reducing operational risks. In essence, it shifts maintenance from a reactive mode to a proactive and data-driven approach.

Diesel engines, renowned for their robustness and efficiency, are the workhorses of numerous industries, including transportation, power generation, and heavy machinery. Ensuring the reliable operation of these engines is of utmost importance, as downtime or unexpected failures can result in substantial financial losses and safety risks. Prognostics for diesel engines, therefore, present a compelling challenge and an area ripe for innovation. They are known for their durability and efficiency, making them indispensable in many industrial settings. However, their complex and nonlinear behavior and diverse operating conditions and environments make RUL estimation a formidable task. The challenges of prognostics in diesel engines can be summarized as follows: Firstly, diesel engines exhibit nonlinear behavior, which can change significantly based on load, speed, and operating conditions. Capturing these nonlinearities is essential for accurate RUL estimation. Second, data collected from diesel engines are often heterogeneous, originating from various sensors and sources. Combining and preprocessing these data sources effectively is crucial. Third, obtaining failure data for diesel engines is challenging due to their long lifetimes. Consequently, building prognostic models with limited failure cases is a common problem. Considering the factors above, it becomes imperative to undertake a comprehensive fault prognosis for diesel engines and ascertain the remaining useful life. Given the considerations discussed earlier, it is imperative to swiftly engage in a comprehensive diesel engine fault prognosis and estimate the remaining useful life. Additionally, there is also a need for these prognostic approaches to possess intelligence and user-friendliness, facilitating autonomous problem-solving within automotive systems. Over recent decades, extensive endeavors have explored and refined efficient tools and methodologies for fault prognosis. This area of research is garnering significant attention in academic and industrial circles.

This dissertation addresses these challenges by (1) Proposing a physics-infused

1D-CNN-based deep learning framework for diesel engine fault detection. (2) Developing an innovative hybrid ensemble learning framework that integrates physics principles into a 1D-CNN-based ensemble deep learning model to detect faults in diesel engines. (3) Developing an end-to-end hybrid physics-informed 1D CNN-based prognostics framework to estimate the remaining useful life of the fuel injector in the context of the increase in the amount of fuel it injects.

1.3 Research Challenges

To build computationally efficient frameworks for fault diagnosis and prognosis of cyber-physical systems, specifically diesel engines, the following research challenges need to be addressed:

Research Challenge 1: How to address issues related to the extant diagnostic methods that are inefficient when dealing with the rapidly evolving technical complexities of the new-age automobiles?

Research Challenge 2: The increased complexity of the automotive system renders conventional fault diagnostics quite time-consuming, resulting in increased servicing and after-sales costs, which ultimately transpires into immense customer dissatisfaction. Can methods be developed to address this prevalent issue?

Research Challenge 3: How to develop computationally efficient diagnostic methods that are intelligent and easy to use to enable autonomous troubleshooting of automotive systems?

Research Challenge 4: Can we develop methods that can address the challenges faced by the data-driven diagnostics and prognostics models when it comes to being domain agnostic and not considering the underlying governing physics of the system?

Research Challenge 5: Diesel engines exhibit nonlinear behavior, which can change significantly based on load, speed, and operating conditions. Can we develop prognostic methods to capture the nonlinearities essential for accurate RUL estimation?

1.4 Research Contributions

Despite significant advancements in diesel engine fault diagnostics and prognostics. Existing diagnostics and prognostics tools are prone to certain limitations that reduce their practical utility. Additional challenges arise due to the increasing complexities of the new-age automotive systems. The **major contributions of this dissertation** are as follows:

1. A novel state-of-the-art end-to-end physics-infused 1D-CNN-based deep learning framework for diesel engine fault detection is conceptualized. The outlined hybrid architecture is completely autonomous in processing the raw test bed data and diagnosing the corresponding fault scenarios.

2. An innovative hybrid ensemble learning framework that integrates physics principles into a 1D-CNN-based ensemble deep learning model to detect faults in diesel engines. Our approach combines domain knowledge with end-to-end 1D-CNN-based methods. The ensemble model follows a stacked generalization approach, featuring two learning levels: level-0 and level-1 (meta-level). At level-0, multiple physics-infused 1D-CNN-based submodels operate in tandem. At level-1, a deep neural network (DNN) aggregates the predictions of the level-0 submodels through an automatic weighted average mechanism, thereby enhancing the predictive accuracy of the final outcome. This hybrid architecture is capable of processing raw data from the test bed and autonomously diagnosing fault scenarios without external intervention.

3. To train and test the feasibility of the proposed hybrid diagnostics architectures, we curated a rare-of-its-kind large-scale diesel engine fault database, termed *NavicEngine* [1], under different operating conditions. The database is located

at (*NavicEngine*).

4. Validation tests of the proposed hybrid diagnostics architecture under different possible running scenarios have been conducted. The models showcased acceptable levels of accuracy on various evaluation metrics. Besides, the model demonstrated stability and robustness to noisy data samples.

5. An end-to-end hybrid physics-informed 1D CNN-based prognostics framework underpinned by physics principles and leveraging a one-dimensional Convolutional Neural Network (1D CNN) to estimate the remaining useful life of the fuel injector in the context of the increase in the amount of fuel injected by it. The delineated hybrid physics-informed 1D CNN-based model operates autonomously, is computationally efficient, and can preprocess unprocessed test bed data and accurately perform fuel injector fault prognostics.

6. To assess and validate the practicality of the proposed hybrid prognosis model, we curated an extensive dataset [2] focusing on the pattern of fault increase over time, defined as the increase in the quantity of fuel injected in diesel engines over time, denominated as the "ProgEngine" repository, and is located at (*ProgEngine*). Fellow researchers working in fault prognosis can benefit from the curated data repository.

1.5 Outline of the dissertation

The dissertation comprises five chapters. The present chapter, labeled as chapter 1, introduces the dissertation, outlining its motivation and the primary research gaps it seeks to address. Additionally, this chapter emphasizes notable research contributions. Subsequent sections of the dissertation are structured as follows:

Chapter 2 provides a comprehensive review of the literature concerning fault diagnosis, model-based fault diagnosis, knowledge-based fault diagnosis and data-driven diagnostics. It also examines research conducted on hybrid physics-informed machine learning models. Additionally, this chapter delves into the challenges present in the current research literature regarding the conventional model-based diagnostics methods and the data-driven fault diagnostics frameworks. Besides, the chapter provides a detailed literature review related to fault prognostics. The chapter also reviews extant literature pertaining to model-based and data-driven fault prognostics. Towards the end, the chapter highlights the challenges the existing conventional and data-driven prognostics approaches face.

In Chapter 3, a physics-infused 1D-CNN-based deep learning framework for diesel engine fault detection is introduced. In particular, we propose a physics-infused diagnostics model in a holistic framework, comprising a physics-based engine model and the deep learning modules, 1 Dimensional-Convolutional Neural Network (1D-CNN), in our case, that can detect and classify different fuel injector faults associated with an automobile diesel engine.

In Chapter 4, We propose an innovative hybrid ensemble learning framework that integrates physics principles into a 1D-CNN-based ensemble deep learning model to detect faults in diesel engines.

Chapter 5 describes a hybrid physics-informed 1D CNN-based prognostics

framework underpinned by physics principles and leveraging a one-dimensional Convolutional Neural Network (1D CNN) to estimate the remaining useful life of the fuel injector of the diesel engine.

Chapter 6 ponders over the conclusions derived from the study's outcomes and offers suggestions for future directions of this research.

Chapter 2

Literature Review

2.1 Fault Diagnostics

In light of the arguments presented in the previous chapters, it becomes essential to promptly conduct a proper diagnosis and health check of the diesel engine. Besides, there is a need for these diagnostic methods to be intelligent and easy-to-use so that to enable autonomous troubleshooting of automotive systems. Over the past few decades, significant effort has been made to research and develop effective diagnostics tools and techniques. Such research is gaining a lot of traction in both academia and industry. Over the years, the typical fault diagnosis has progressed into three major paradigms: model-based, knowledge-based, and data-driven.

2.2 Model-based diagnostics

he model-based approach builds on the data generated from simulated models under nominal and faulty operating conditions. These models require the knowledge of the underlying dynamics of the system in the form of either parameters or

mathematical equations [46, 47]. For instance, Wang et al. [119] used sliding mode deviation to estimate deviations in engine torque, which leads to the detection of engine misfires. Jung et al. [54] used a model-based misfire detection algorithm. A method using training data is used to auto-tune the model. Constantinescu et al. [19] deployed a model to represent the dynamics of an N-cylinder diesel engine. The model, along with cylinder pressure waveforms, aids in detecting the misfire in engine cylinders through efficient torque estimation. In [94], the author proposed process model-based and signal model methods for fuel system diagnosis in a heavy-duty diesel engine. Chen et al. [16] used a 0-D Selective Catalytic Reduction (SCR) nonlinear dynamic model for detecting and isolating NOx sensor and dosing faults. Nohra et al. [83] proposed a nonlinear model with four state variables for detecting and isolating six different engine system faults. The devised strategy is built on top of the theory of Gain Schedule Control as applied to the well-known Takagi-Sugeno diesel model. The method demonstrated satisfactory levels of efficiency in the presence of noisy data, taking the described model-based approach quite close to the actual working conditions of the physical system. Kiencke et al. [55] used a model-based approach, using Kalman filters, to estimate engine torque from the angular velocity, which ultimately helps detect engine misfires. The discrete state-space model works with angular steps instead of equidistant time-based sampling. Nohra et al. [84] developed an innovative model-based approach in combination with variable geometry turbocharger control for fault diagnosis in diesel engines. The overall diagnosis algorithm uses a mu analysis control over an LTI diesel engine model. The simulation results establish the efficacy of the proposed algorithm in the presence of uncertainties and noisy data. Ludwig et al. [68] used a model-based approach for fault detection in diesel engine turbochargers. The proposed parametric model encapsulates the underlying nonlinear dynamics of the diesel engine. The proposed Hammerstein model

was compared with a neural network. Then, the model-based approach proved to be faster than the neural network. Still, it relied heavily on the available static and dynamic data, thus rendering the fault identification process not so easy. Frisk et al. [29] used a model-based approach for the residual generation to diagnose faults. The proposed method addresses the issue of analyzing robust residual generation under the influence of deterministic disturbances and parametric uncertainties. However, owing to the complex dynamics of the involved system, developing accurate models that can enable robust failure diagnostics in diesel engines is challenging.

2.3 Knowledge-based Diagnostics

Knowledge-based models combine domain expertise and logical reasoning to diagnose system faults. In addition, relevant equipment and systematic fault diagnosing theory are put to work while performing complex system fault detection and isolation. Xiaojun et al. [123] devised a knowledge-based model for fault diagnosis in an automobile engine. The proposed model is based on Hierarchy Diagnostic Principle (HDP) and encompasses signal processing knowledge, domain expertise, and functional information related to the engine system to be checked for fault presence. Bonissone et al. [11] from General Electric have implemented an expert system based on prior knowledge of the system for fault detection. The proposed method caters to the issues of knowledge representation, knowledge acquisition, knowledge inference, control mechanisms, and uncertainties of the system. The devised technology was tested to demonstrate its feasibility of troubleshooting diesel locomotives. Xu et al. [124] proposed a belief rule-based (BRB) model for diagnosing abnormal wear in marine diesel engines. The initial BRB model was built using the knowledge of domain experts. The model is further optimized using the data samples gathered during the

actual operation of the diesel engine. A series of BRB models are combined to obtain the desired diagnostic results. The research results demonstrated the possibility of using the proposed model for engine fault diagnostics in a probabilistic manner and can identify concurrent faults in the system. Twiddle et al. [114] proposed a fuzzy-rule-based model for condition monitoring and diagnosing faults in the cooling system of diesel engines. The model outputs are compared with the measured data to generate residuals. The generated residuals are then analyzed to conclude the specific fuzzy class indicative of the potential anomaly in the engine system. Campoverde et al. [13] developed an expert system-based model deriving its strength from the knowledge base of the domain experts, thus taking into account the cause behind a specific condition and ultimately determining the appropriate mitigating action to be taken. The expert system can identify and isolate faults by incorporating data from Condition-Based Maintenance techniques. The Condition-Based Maintenance data includes information pertinent to engine specification, operating conditions related to faults, and the corresponding procedures for fault mitigation. Nabende et al. [79] used an expert system based on Bayesian networks for diagnosing faults in Heavy-Duty Diesel Engines (HDDEs). However, the reliance on domain expertise renders the knowledge-based models not so practically feasible in many real-world applications where the prior knowledge of the system might not be available beforehand.

2.4 Data-driven Diagnostics

Unlike their knowledge-based counterparts, data-driven models' [6, 18, 30, 32, 38, 45, 49, 58, 60, 61, 64, 67, 71, 87, 96, 99, 110, 113, 115, 117, 129, 133, 136] effectiveness depends on the quality of the available data and the characterization of the variations in the process data. With the advent of new computational hardware

tools and supporting software frameworks, data-driven methods are gaining ground in fault detection [3, 5, 12, 17, 41, 50, 59, 60, 81, 117, 128, 136] and isolation domains. Data-driven methodologies such as Machine Learning and Deep Learning are being seen as an alternative to the conventional fault diagnostics approaches owing to their cheaper deployment cost, sensitivity to faults, and high efficiency when fed with large datasets. Data-driven models process a large amount of observational data from the actual operations of the engine system while imbibing the process knowledge to establish a relationship between the fault conditions and their causes. Besides, the data-driven models can use the relationship between the fault modes and their respective causes to estimate the fault severity. The data-driven models also assist in deciding the appropriate course of action to follow once an abnormal condition is reported. For example, Liu et al. [63] devised an Artificial Neural Network (ANN) based model for detecting misfires in the diesel engine. The proposed model was trained and validated on the data coming out of a V6 turbocharged engine. Chen et al.[14] used ANNs, which input the torsional vibration signals from the crankshaft along with the acceleration signals from the engine block, to detect engine misfires in Internal Combustion Engine (ICE). Singh et al. [103] used a Support Vector Machine (SVM) classifier, fed with the radiated sound from the engine cylinder block and the exhaust tailpipe, to detect engine faults. Moshou et al. [76] developed a One-Class Support Vector Machine (OCSVM) and One-Class Self Organizing Map (OC-SOM) for detecting fuel injector faults. The proposed models, i.e., the one-class classifiers, used the acoustic signals from the fuel injectors under different operating conditions. Kowalski et al. [57] used an ensemble machine learning model to detect 14 fault classes in a 4-stroke marine diesel engine. For fault pattern classification, the ensemble model relies on various signals omitted by the engine specimen. Rogov et al. [93] used Deep Neural Network (DNN) in combination with the Quantitative Associated Rule

Mining Algorithm (QARMA) to diagnose four different fault conditions in a diesel engine. The author [85] used fully connected and convolutional network-based models to analyze the engine angular velocity pulse pattern to decipher fuel injector faults in a diesel engine. Tao et al. [111] proposed an extreme gradient-boosted algorithm to diagnose misfiring faults in diesel engines. The model used the engine's time-frequency vibration signals via triaxial accelerometers to diagnose misfire faults. Yan et al. [126] used a combination of an artificial neural network and an evolutionary algorithm to diagnose combustion faults in a diesel engine. The model used the maximum eruption pressure and exhaust temperature as inputs and differentiated the fault conditions into six categories. Jiang et al. [52] developed a novel adaptive sparse attention network (ASAN) to detect valve failure in a direct injection diesel engine using vibration angle data of the cylinder valves and envelope spectrum as model inputs. Zhan et al. [132] used a combination of Variational Mode Decomposition (VMD) and Convolutional Neural Network (CNN) to perform fault diagnosis in a six-cylinder diesel engine. The model could classify six fault classes: cylinder misfire, insufficient oil supply to the high-pressure pump, cylinder misfire, air filter clogging, and damage in the oil supply pipe. Shao et al. [97] used an ensemble learning approach for fault detection in a diesel engine. The proposed model was exposed to five different working conditions, used fifteen input features, and classified three abnormal behavioral patterns: compressor fault, air cooler fault, and defects related to delay in the injection valve timing. Wang et al. [118] used Auto Encoder (AE) and CNN models to diagnose air system faults. Specifically, the model uses twelve input features to detect three air systems faults: EGR closed-loop monitoring fault, low boost pressure fault, and the defect due to EGR valve communication timeout. Zhao et al. [134] proposed multi-branch convolutional neural networks (MBCNNs) to pinpoint six faulty scenarios related to valve clearance. The model was trained

using twelve different diesel engine valve clearance operating conditions. Zhong et al. [137] came up with an intelligent fault diagnostics method, comprising correlation analysis and Deep Belief Network (DBN), to diagnose faults in the marine diesel engine. The correlation analysis comes in handy in the dimensionality reduction of the input feature vector and filters out any unwanted and not-so-important feature. The DBN then takes the reduced feature vectors as input and segregates six fault conditions pertinent to the engine. Sharma et al. [98] used different combinations of a number of decision trees, processing vibration signals from engine block, to distinguish between faulty and non-faulty conditions. Szabo et al. [108] used a fuzzy deep learning algorithm, to process vibration signals, to detect misfire faults in eight cylinder large gas engine.

2.5 Drawbacks of model-based and data-driven diagnostics approaches

Despite all the aforementioned advantages, data-driven models come with their own set of shortcomings [130]. The data-driven models, such as machine learning and deep learning, have recently gained much traction in diesel engine fault diagnostics. However, we hardly see any effort in incorporating the systems' physics in the data-driven diagnostic algorithms. Typically, the data-driven methods rely on training black-box models, which are susceptible to a few restrictions. Firstly, the performance of these models is a direct function of the dataset size. On small and scanty data, the performance of the data-driven models, especially the deep learning ones, degrades considerably, and the models fail to generalize well beyond their seen environment. The machine/deep learning models show absolute disregard towards the

system's underlying physical laws and governing principles under consideration, in our case, a diesel engine. Even though the performance of the data-driven models improves with the increasing data size, the results fall short of physical interpretability and scientific consistency. As a result of their agnosticism towards the physics of the system, the data-driven models become victims of skeptical criticism at the hands of researchers across the domains, which limits the scope of using these models for subsequent scientific advancements. On the other hand, the knowledge-based and the physics-based models have their respective limitations. The physics-based models are grounded in core physics doctrines and imbibe knowledge from subject matter experts. In most cases, the foundational base of physics-based models pivots around the mathematical equations derived from understanding the complex dynamical system. However, the physics-based models' dependency on expert knowledge makes them difficult to build. Moreover, at times the physics-based knowledge models fail to imitate the exact system behavior they aim to represent, and this occurs due to the result of oversimplifying approximations or lumped parameter analysis performed during the development of these models. The modeling abstractions lead to low-fidelity models that are not apt for use in a highly dynamic system such as a diesel engine, where the computations are time-sensitive and iterative in nature, for example, diagnosing a fault in an engine and then performing suitable control maneuvers to mitigate the aftereffects.

2.6 Physics-Informed Machine Learning (PIML)

An upcoming concept to overcome the modeling limitations as mentioned above is the Hybrid or Physics-Infused Machine Learning (PIML) models. The PIML model combines physics-based and data-driven machine learning/deep learning mod-

els [10, 15, 23, 24, 33, 35, 37, 39, 48, 65, 66, 77, 80, 86, 90, 91, 109, 125, 135, 138, 139]. For instance, Cubillos et al. [21] used a Grey-box Neural Model (GNM), a combination of phenomenological models and feed-forward neural networks, to model the fish-meal rotary drying process. Due to the better representative capabilities of the GNM, an enhanced control performance was observed when the process variables were exposed to disturbances. Singh et al. [102] used a combination of physics-based models and Long Short-Term Memory (LSTM) networks, termed as Physics-Infused Long Short-Term Memory (PI-LSTM), to model highly dynamical systems of the cancer cell growth and inverted pendulum. The results show that the PI-LSTM model outperformed the pure data-driven LSTM networks. As evident from the recent work in the field of hybrid or physics-infused machine learning models, the PIML models showcased some promising results while replicating the dynamic behavior of a complex physical system. In recent years, the discipline of hybrid or physics-infused machine learning has primarily branched into three major categories: *Combination Approach*, *Constraint Approach*, and *Embedding Approach*. The combination approach employs the physics of the system either as an extra input to the machine learning block (parallel hybrid model) or as a single direct input to the machine learning model (sequential hybrid model) [4, 7, 51, 82, 89, 101]. In the constraint approach, people tend to include physics information as an extra constraint to the objective function by augmenting the loss term [22, 25, 44, 78, 100, 105]. Adding the physical knowledge as an added term to the loss function ensures that the physically constrained model achieves better generalization when the training data is small and not fully representational of the system dynamics. In the embedding approach, the methodology is to substitute the high-cost computational portion of the physics model with machine learning block [56, 62, 69, 92]. The approach intends to speed up the physics solver and enhance accuracy. The arguments as mentioned above emphasize the importance

of using the governing physics of the system along with the data-driven model as a new modeling paradigm for imitating the behavior of a complex dynamic system.

Similarly, in the sphere of fault diagnostics, contrary to the pure data-driven models, the PIML model can accentuate the features that contain the most vital information pertinent to the diesel engine fault. Especially when it comes down to fault classification, the PIML model, leveraging the underlying physical laws of the system, can help improve the accuracy compared to a pure data-driven or physics-derived fault detection model.

Chapter 3

Hybrid Physics-Infused 1D-CNN based Deep Learning Framework for Diesel Engine Fault Diagnostics

In this chapter, we try to address the limitations faced by conventional model-based approaches and their data-driven counterparts when applied to fault diagnostics by resorting to a hybrid physics-infused deep learning framework. In particular, we propose a physics-infused diagnostics model in a holistic framework, and the deep learning modules, 1 Dimensional-Convolutional Neural Network (1D-CNN) in our case, can detect and classify different fuel injector faults associated with an automobile diesel engine. Specifically, we use an Auto Encoder (AE) model to reduce the dimensionality of the data from the test bed (diesel engine). After that, the 1D-CNN module of the framework processes the combined feature vector, the output of the AE, and the data coming out of the physics-based engine model, to diagnose and classify three different fault categories along with the nominal working condition. The input of the hybrid diagnostics framework is the sensor data coming out of the engine test

bed, and the outputs are the four different categories: Nominal or no-fault case, fault due to injection pressure, fault due to a delay in injection duration, fault due to the change in the start of injection (SOI). The main contributions of this chapter are as follows:

1. A novel end-to-end physics-infused 1D-CNN-based deep learning framework for diesel engine fault detection is conceptualized. The outlined hybrid architecture is completely autonomous in processing the raw test bed data and diagnosing the corresponding fault scenarios.

2. To train and test the feasibility of the proposed hybrid architecture, we curated a rare-of-its-kind large-scale diesel engine fault database, termed *NavicEngine* [1], under different operating conditions. The database is located at (*NavicEngine*).

3. Validation tests, of the proposed hybrid architecture, under different possible running scenarios, have been conducted. The model showcased acceptable levels of accuracy on various evaluation metrics. Besides, the model demonstrated stability and robustness to noisy data samples.

In essence, the use of physics along with the 1D-CNN module ensured better exploitation of the underlying patterns in the data and resulted in a satisfactory performance level for a highly dynamic diesel engine system. We demonstrated that by using the direct raw data from the engine; we could diagnose faulty diesel engine conditions in a completely autonomous manner.

Chapter Outline: We begin with the diagnostics-dataset collection setup details, followed by the strides made toward collecting nominal and faulty data samples. After that, we continue with a detailed description of the proposed hybrid fault diagnosis architecture. Subsequently, we present an analysis of the performance of the

Table 3.1: Engine geometry and specifications

Displacement	[L]	7.6
Bore	[mm]	116.6
Stroke	[mm]	118.9
Compression Ratio	[-]	16.9

physics-infused 1D-CNN model across various evaluation metrics. In conclusion, we provide a comprehensive discussion of the empirical results obtained and explore the potential future avenues for this research.

3.1 Experimental Setup and Data Collection

3.1.1 Experimental Setup

Experiments were conducted on an inline 6-cylinder Navistar production DT diesel engine (Figure 3.1) donated by Pure Power Technologies, Inc. The engine is a production engine with overhead valves, electro-hydraulic fuel injectors, and a dual-staged turbocharger. Table 3.1 describes the engine geometry. The engine was instrumented with sensors to measure airflow rate, fuel flow rate, fuel rail pressure, intake and exhaust pressure, and various temperatures along the intake and exhaust flow paths. The air flow-rate was measured using a laminar flow element and the fuel flow-rate was measured using an AVL fuel balance. Individual exhaust temperature measurements in each port serve as an indicator of the combustion process in each individual cylinder. The temperature is critical in sensing potential faults on individual cylinders in conjunction with other sensor signals.

The engine speed was controlled using an active Dynamometer via a test bed automation system, AVL PUMA. A CCP (Can Calibration Protocol)was established

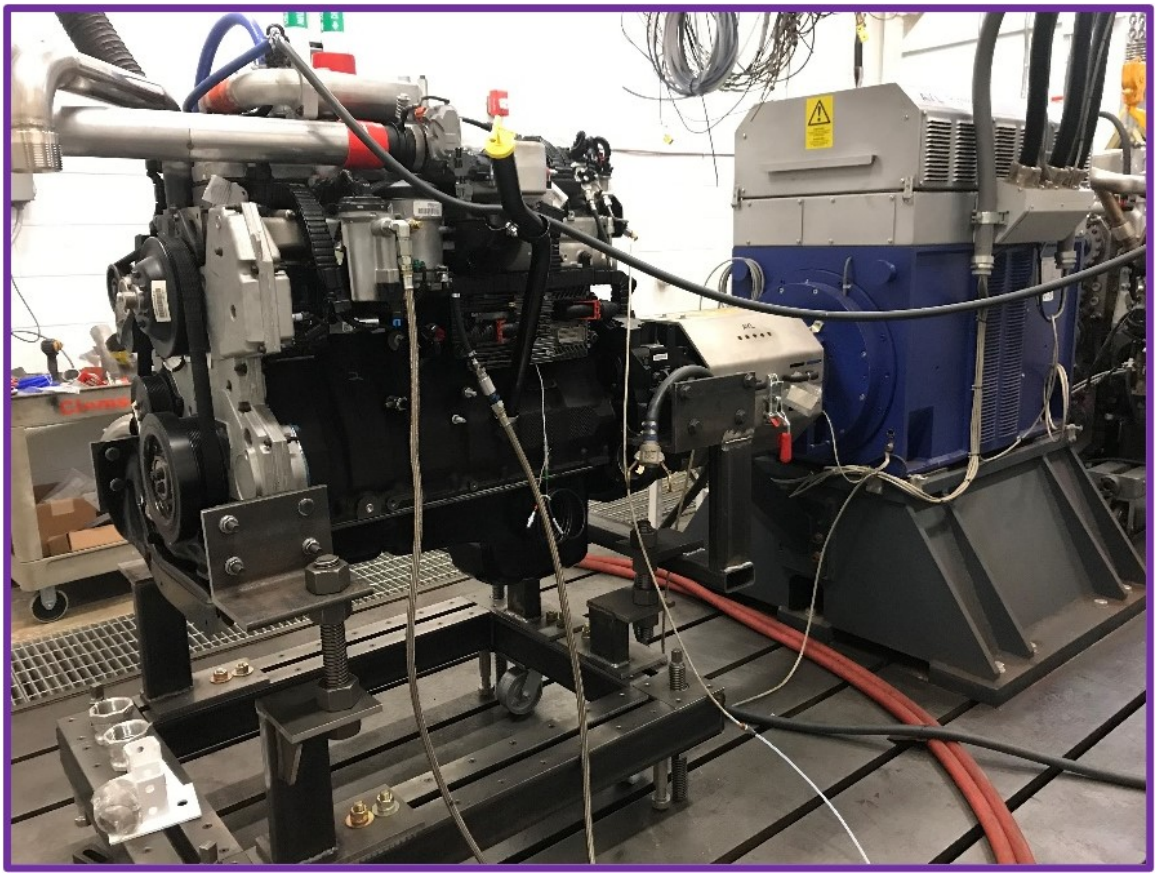

Figure 3.1: Navistar Engine Test Bed

using INCA to connect the ECU to the data acquisition and controls systems in the test cell. The injection parameters were controlled using an open engine control unit (ECU) interface through INCA software (Figure 3.2).

3.1.2 Data collection

The scope of data collection was base-lining the engine first by operating on the pre-calibrated ECU maps. Then faults were simulated by controlling the actuators demand values individually and recording the response. The data set consists of two different files for each operating conditions: an ECU save-file, and a PUMA test cell automation save-file. The ECU file contains data from approximately 100 real and

Figure 3.2: INCA, PUMA and Indicom setup for data acquisition

Table 3.2: Baseline Parameters

Baseline				
Engine Speed (RPM)	1000	1200	1400	1600
Injection Duration (ms)	1.2	1.2	1.2	1.2
Start of Injection (deg aTDC)	0.64	1.95	2.84	2.60
Fuel Injection Pressure (MPa)	6.0	8.0	8.0	15.0
Injection Faults (Unit change from baseline)				
Injection Duration (ms)	-0.4,+0.4,+0.8			
Start of Injection (deg aTDC)	-5,+5,+8	-7,-12,+9	-7,-11,+5	-5,+5,+8
Fuel Injection Pressure (MPa)	-2,+2,+4			

virtual sensors monitored by the ECU. This data was collected at a rate of once per engine cycle. The PUMA file has the test cell conditioning information, including temperatures, pressures, and flow rates, at a sample rate of 10 Hz. Beyond the baselining data for model validation and calibration, several sample fault data sets were collected.

3.1.3 Design of Experiments

At different engine speeds (1000,1200,1400, and 1600), three different types of potential faults associated with the fuel injector or injection system were introduced. First, the injection duration on one cylinder individually was increased, or decreased, from the nominal ECU value. As evident from table-2 the injection duration for base calibration is 1.2 ms at all speeds but for the fault cases the duration was first decreased by 0.4 ms and then for the next two faults increased by 0.4 and 0.8 ms respectively in the 1st cylinder. This was to simulate a failure with the injector including injector deposits that reduce the fuel flow or eroded injector nozzle holes that increase the amount of fuel flow. This failure could also be associated with a particular injector's solenoid coil or command signal which may cause the injector to open for more or less time than commanded by the ECU.

A second injector fault simulated was achieved by altering the injection timing of one of the cylinders slightly compared to the ECU-commanded signal. The base map has different start of injection timing event based on the different speed. Thus the baseline timings are different for all speeds. For example, at 1000 RPM the start of injection was retarded by 5 crank angle degrees for the first fault then advanced by 5 and 8 crank angle degree for the next set of faults. However, the magnitude of this advanced and retarded timing event varied as the baseline value varied across the speed range. Finally, the fuel pressure set-point was also perturbed from the ECU command and the induced fault trend followed the same trend as the aforementioned parameters. Both the nominal and the simulated faults data-sets were recorded with a time-based recorder to equalize the sample size across different data sets and have been briefly summed up in the Table 3.2. The faults were introduced in either one or all cylinders depending upon the response in the exhaust temperatures and the brake torque. Only the injection duration fault was simulated for one injector (1st cylinder), the remaining faults were induced case-wise in all 6 cylinders simultaneously.

3.2 Data Preprocessing

Experimental data collected from real-world setups come with different data modeling challenges. Often there are issues of arbitrary missing data, erroneous sensor readings, and/or mismatched sampling frequency rates for other data streams. Preparing well-trained machine-learning models to address such irregularities associated with experimental datasets is crucial. We undertook the following key data preprocessing steps to prepare our engine data for the subsequent modeling phases:

- **Variable Selection:** We investigated all the variables from the INCA and PUMA data acquisition setups and discarded several irrelevant sensor readings.

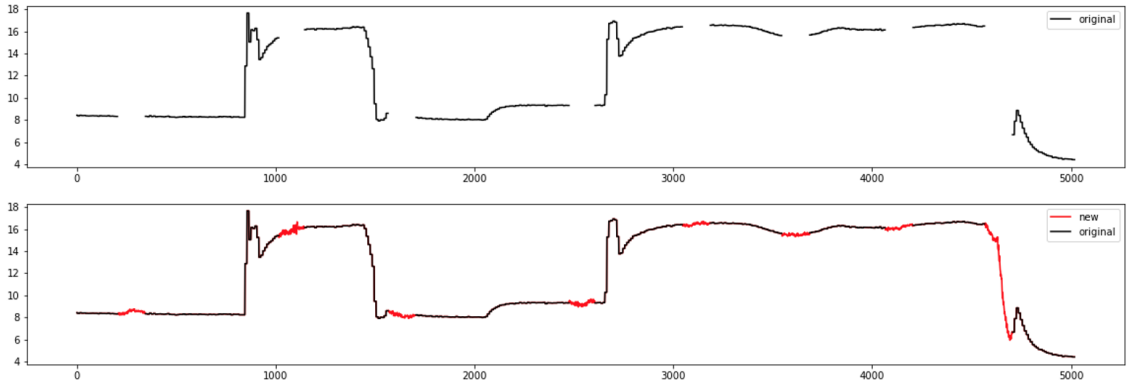

Figure 3.3: Data imputation example. Filling in the missing fuel flow rate values (in red) for rpm 1600 with MLP model.

We selected 30 out of the total 40 PUMA variables and 59 out of 628 INCA variables. The variable selection is driven by the relative relevance of the sensors pertinent to the injector faults. The channels chosen for PUMA logged information related to the pressure, temperature, and volume flow rates of coolant, engine oil, fuel, intake, and exhaust air. Controlling oil and coolant temperatures are vital to maintaining a controlled ambient condition for baselining the engine across different engine speeds. While baselining the engine, the intake and exhaust air measurements were used to validate how an engine would operate in normal conditions and react to controlled inputs simulated as faults. INCA channels comprise the override values given as a command to the actuators, the feedback signal received from the actuators, and the measurement value read after the changes have been made. These channels also capture the magnitude of the faults introduced based on their physical unit of measurement.

- **Data Imputation:** A few sensor readings (variables) had missing data entries in the initial raw engine data files. As a preprocessing step, we replaced the missing values using neural network-based regression models. For instance,

when the engine data was collected at different speeds, the variable *fuel flow rate* for the PUMA setup had several inconsistent and missing values. We trained separate multilayer perceptron (MLP) models for each speed level by training them on the data points where a sensor reading was available. These models were then used to fill the data points where sensor readings were corrupted or missing. The input to the model was 29 different variables out of the 30 total INCA variables, and the output was the variable with the missing values. The MLP models had two hidden layers with 48 and 16 neurons, respectively. Rectified Linear Unit (ReLU) was used as the activation function in both these layers to help model the non-linear dynamics in the variable relationship. Table 3.3 summarizes the performance of the MLP models used to fill the missing values of *fuel flow rate* variable for the different speed levels. Figure 3.3 visualizes the result of data imputation for an instance of rpm 1600, where the red points are the missing values as substituted in by the corresponding MLP model for that speed.

- **Data Resampling:** Another challenge while processing the experimental engine data was that different subsets of variables were measured at different frequencies and for different time duration. To have a consistent data representation, we resampled the variables to match the same frequencies such that the subsequent hybrid machine learning models can conveniently ingest the engine data at a constant rate. We employed a simple bilinear interpolation scheme to resample subsets of sensor readings to match the data collection timestamp of all the data points.

Table 3.3: Data imputation performance of the MLP models at different speed levels

Speed (rpm)	fraction of missing data	MLP model RMSE
1000	789/9591	0.007
1200	820/7065	0.008
1400	1220/7534	0.011
1600	1100/5019	0.019

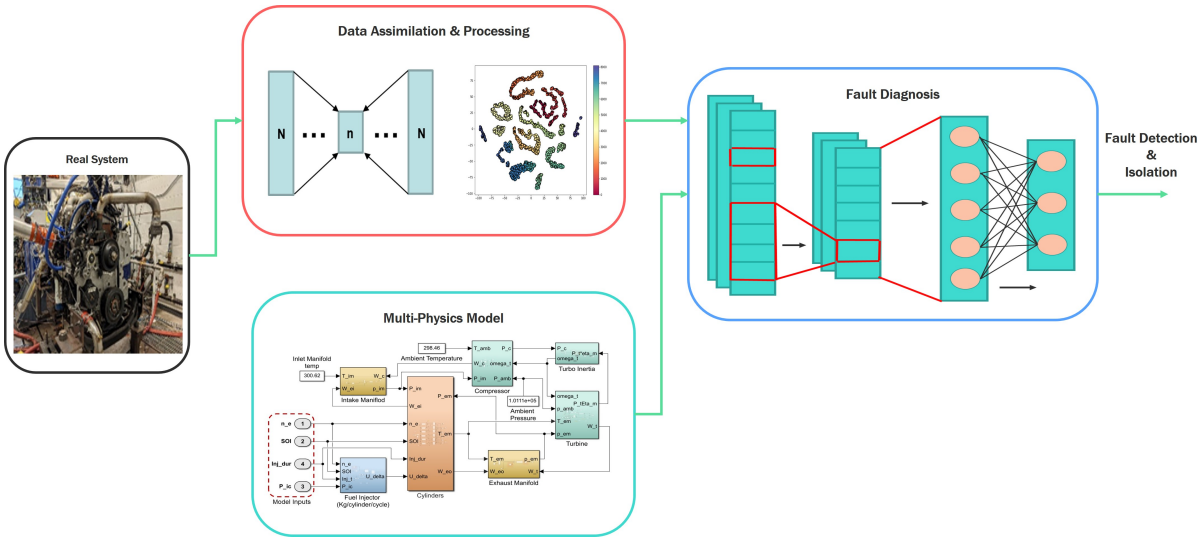

Figure 3.4: Hybrid Physics-Infused 1D CNN-based Fault Detection and Isolation Pipeline

3.3 Hybrid Fault Diagnostics Architecture

The proposed architecture is a concoction of the autoencoder (AE), the physics (Simulink-based) model of the engine, and the 1D-CNN module. The AE network processes the signals from the engine test-bed sensors, creating a latent vector representation of the entire feature space. The Simulink-based engine model takes in the input control parameter and generates specific outputs. The combined feature vector, outcomes of the AE and physics-based engine models, is fed to a series of convolutional blocks, where each block, in turn, is a combination of convolutional and max pooling layers. Figure 5.3 depicts the overall fault diagnostics computational pipeline.

3.3.1 Autoencoder model

As the first component of our hybrid model architecture, we developed autoencoder models to represent the high-dimensional space of sensor readings for different experimental scenarios. An autoencoder is a neural network used to learn a representation (encoding) of a high-dimensional dataset, usually for dimensionality reduction or data compression. An autoencoder consists of two parts: an encoder network and a decoder network. The encoder maps the input data to a lower-dimensional representation, and the decoder maps the encoded representation to the original input data. The goal of an autoencoder is to minimize the difference between the original input and the reconstructed output so that the encoded representation can capture the essential features of the input data. Compared to popular dimensionality reduction methods like principal component analysis, an autoencoder is much more flexible in its modeling capabilities and can model non-linear variable relationships in data.

3.3.1.1 Model Architecture and Training

For our preprocessed engine data, we trained autoencoders to take the combined PUMA and INCA variables and reduce them to a low-dimensional latent space, capturing the essential sensor data properties. Figure 4.2 gives a high-level overview of the autoencoder architecture used in our work. It reduces N (in our case 89) sensor reading to a low dimension of n. The network configuration of the encoder and the decoder are identically opposite, with two hidden layers of 64 and 32 neurons each. To model the non-linear relationships, we used the ReLU activation function. The overall model was trained using a combined dataset from all speed levels under a nominal engine run scenario. During the training phase, the model was tuned to accurately reconstruct the original high-dimensional sensor data. Table 4.1 shows the

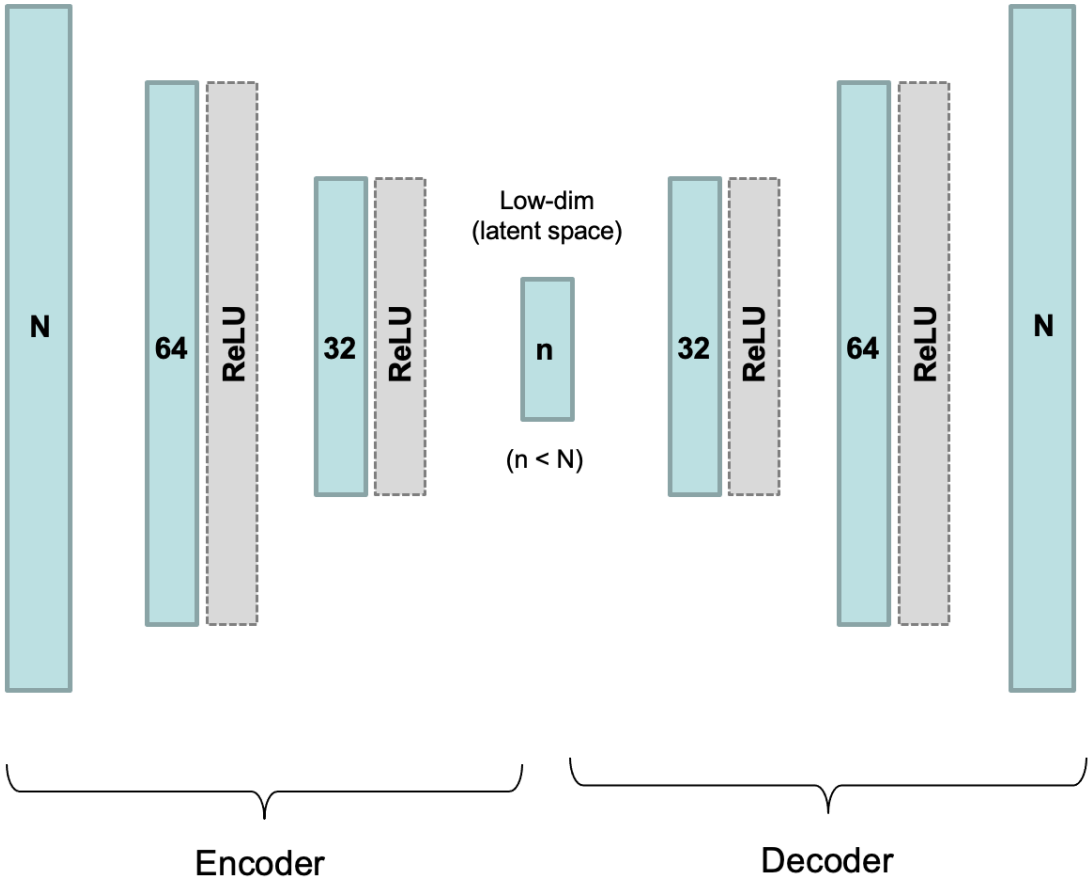

Figure 3.5: Overview of the autoencoder architecture.

reconstruction performance of the trained autoencoder for different latent dimensions (i.e., n) in terms of root mean squared error (RMSE). We tested with different latent dimensions by training three different autoencoders with latent-dimensions 16, 8, and 4, respectively. As seen in the table, the reconstruction performance of the autoencoder decreases with a decrease in the latent-dimension, which is expected as any dimension reduction technique will suffer from information loss with lesser dimensions.

Table 3.4: Autoencoder reconstruction results for different latent dimensions

latent dimension	reconstruction RMSE
16	0.005
8	0.006
4	0.011

3.3.1.2 Latent Dimensions

Once the autoencoder is trained, we pass on the latent dimensional representation of the sensor readings to the next stage of the hybrid diagnosis model. The autoencoder learns to capture the critical data properties in the low-dimensional latent space, a rich data source for subsequent modeling. For visualizing the features captured in the latent space, a common approach is to use another standard dimensionality reduction method on top of the latent dimensions to visualize the data properties. For this, we used t-SNE (t-distributed stochastic neighbor embedding), which helps visualize high-dimensional spaces by projecting them to lower dimensions that retain the high-dimensional properties and relationships among the data points. Figure 4.3 shows the t-SNE projection results of 16-dimensional latent variables for an rpm 1600 data. The continuous color scale maps to the duration of the engine run. We explored three distinct clusters shown in the projection (highlighted in the figure with orange, green, and blue circles) and found them to correspond to different operating states of the engine.

3.3.2 Physics-based model

3.3.2.1 Model Introduction

The performance of a pure data-driven model for engine fault diagnostics primarily depends on the data size. Particularly the data collected on several different

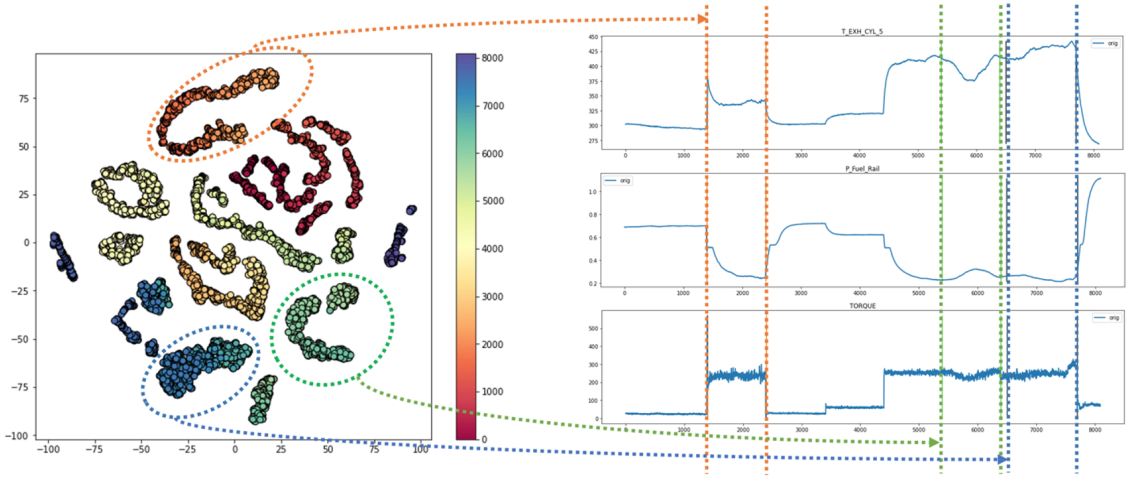

Figure 3.6: Visualization of the latent dimensional representation of the high-dimensional sensor readings for rpm 1600 instance.

operating conditions and the severity of the faults. Collecting experimental data for the faulty behavior is challenging, as it can permanently damage the engines' components if we induce faults for all possible scenarios and run the engine. The data collected for this paper was in a controlled environment; it monitored every engine output and ensured that the induced faults should not damage any engine component. Safely inducing faults restricted us with lower fault severity levels in generating the data. However, in a real-life scenario, if any fault occurs in the system, it doesn't necessarily happen in a limited manner. Most of the time, faults significantly impact the system, which is impossible to incorporate in the test cell for data collection. Thus, in such cases, the role of a physics-based model is crucial, as it can operate on the desired condition, unlike the test cell engine. Therefore, infusing a well-studied physics-based model, incorporating domain expertise, combine with a data-driven model can overcome the issues of generalizability and sparse datasets.

Various physics-based diesel engine models are available in the literature, for example, 0D, 1D, 2D, and 3D Computational Fluid Dynamics (CFD), to name a few [20, 43, 70, 88]. 0D models don't have a spatial dependency; those models are

only time-dependent. The 0D model comprises algebraic and ordinary differential equations, as it's a function of only one variable, time. On the other hand, 1D, 2D, or 3D models include one, two, or three spatial dimensions within the chosen coordinate system. Such models are tough to build and rely heavily on domain expertise. Generally, computational time and accuracy are directly proportional for such models and are considered a primary deciding factor in choosing which model to use. We need a model that can simulate faster than a wall-clock time on a regular windows machine for engine fault diagnosis and can still give an intuition of the actual system behavior (diesel engine in this case). To strike a balance between computational efficiency and the level of accuracy, we decided to use the 0D model and develop a few of the subsystems of interest with high fidelity.

The physics-based engine model's structural diagram that shows the flow direction and connection between the inputs and outputs of all subsystems. The control inputs and final outputs of the engine model are represented in red dashed rectangular box and ellipse, respectively.

3.3.2.2 Model Details

This paper proposes a 0D high-fidelity physics-based engine model developed for a 7.6-liter Navistar engine. This Navistar engine is established at Clemson University's Powertrain Laboratory. To develop the 0D model, we used a Mean Value Engine Model (MVEM) technique, which neglects the discrete cycle of the engine and uses an effective value from the combustion cycle [40]. Several mean value models on diesel engines are available in the literature [42, 73, 104, 120]. Specifically, we considered the model developed by Wahlstrom and Eriksson [116] as a baseline model and further improvised the fidelity of a few of the subsystems. We develop this model in the MATLAB and Simulink environments. The overview of the engine model is

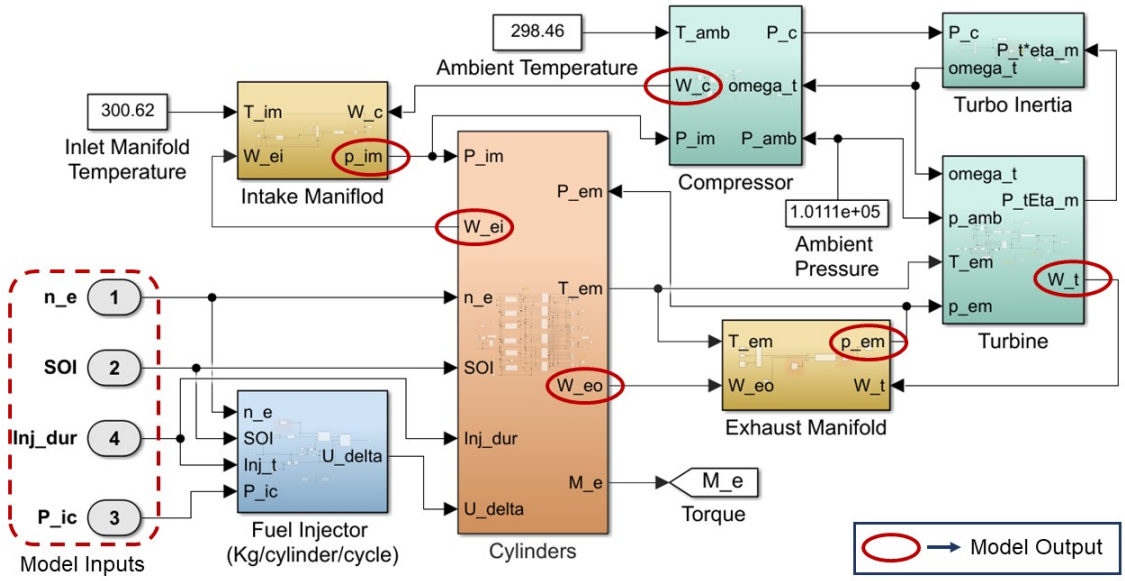

Figure 3.7: The physics-based engine model's structural diagram which shows the connection and flow direction between inputs and outputs of all subsystems. The control input parameters and output measurements of the engine model is represented in red dashed line rectangular box and ellipse, respectively.

depicted in Figure 5.4. The model captures mass flow, temperature, pressure, and total torque from multiple engine subsystems using fourteen state variables (shown in Figure 5.4).

The input/control parameters of physics-based diesel engine model are mentioned below:

1. Engine speed (rpm) - n_e
2. Start of injection (Crank angle degree - CAD) - SOI
3. Injection duration (ms) - inj_dur
4. Injection pressure (MPa) - P_ic

The modeling details are elaborated in the following two sub headings.

Cylinder and fuel injector subsystems

The process of combustion is the essential process of engine operation, and the overall engine model's accuracy is highly dependent on the fidelity of the combustion model. Here, we use a high-fidelity cylinder model, which is primarily based on thermodynamic principles. This cylinder model simulates inside cylinder processes for all CAD. This model gives several outputs for an entire cycle, such as torque, temperature of the cylinder exhaust, mass flows of intake and exhaust manifolds, and in-cylinder pressure. We extend the thermodynamic cylinder model with a multi-cylinder concept to simulate the effect of a single faulty injector/cylinder behavior on the entire system. The multi-cylinder model has six separate cylinders and fuel injectors. The fuel injection model mimics the behavior of a high-pressure common rail fuel injection system from the experimental setup. This injection model outputs the injector fuel flow rate using the concept of conservation of mass.

Turbocharger and manifolds subsystems

The intake and exhaust manifolds are modeled with two states: intake and exhaust manifold pressure. Fundamentally the principle of mass conservation and the ideal-gas law are used for modeling manifolds pressure. The differential equation for calculating the manifolds pressure is described below,

$$\frac{d}{dt}p_{im} = \frac{R_a T_{im}}{V_{im}}(W_c - W_{ei}), \tag{3.1}$$

$$\frac{d}{dt}p_{em} = \frac{R_e T_{em}}{V_{em}}(W_{eo} - W_t) \tag{3.2}$$

All the parameters are mentioned in the appendix from Equation 5.5, Equation 5.6, and Figure 5.4. The subsystems of the turbocharger comprise of compressor,

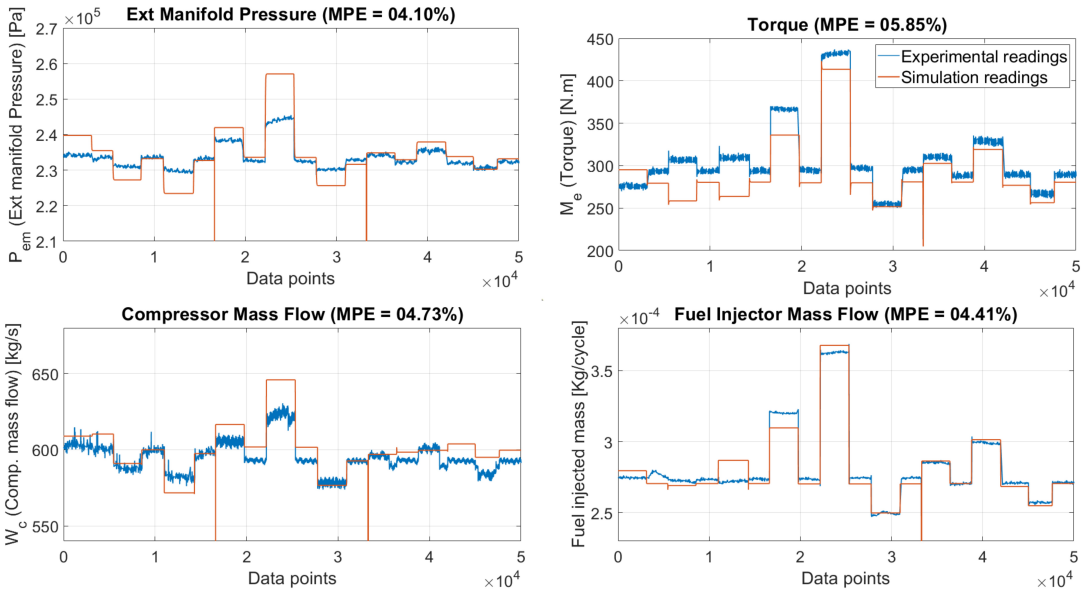

Figure 3.8: Physics-based model validation plots for the engine speed of 1600 rpm with different operating points. Blue and orange lines denote experimental and simulation readings, respectively. The Mean Percent Error (MPE) of each model's output is mentioned in their title.

turbine, and turbo inertial model. We extract the model of a turbocharger from Wahlstrom and Eriksson's model [116].

The physics-based model has been calibrated for the range of 1000-1800 RPM of engine speed and 100-600 Nm of torque. The experimental data is used for the model calibration. The error between experimental and modeled values is minimized for model calibration using a well-known approach, the least square optimization method [75]. This model is validated over experimental data and shows a maximum error of 5-6 %. The validation plots for some of the outputs of the physics-based model are demonstrated in Figure 5.5. We choose the outputs from the physics-based model that overlap with the experimental data. Thus, six outputs from the physics-based model and sixteen from the autoencoder model are sent to one dimensional-convolutional neural network.

3.3.3 1 Dimensional-Convolutional Neural Network (1D-CNN)

Recently, Convolutional Neural Networks (CNNs) have established themselves as the go-to solution for various computer vision tasks, including object detection, face recognition, and numerous other machine learning applications [127]. The widespread adoption of CNNs has made them the default choice in the field, serving as the de facto tool for these tasks. CNNs are a type of deep learning network that serves as a go-to choice for analyzing image or spatial data. The CNNs may seem feasible for various applications like 2D or 3D image processing. Yet, they may need to be more handy when dealing with signals or time-variant data samples in 1D. The 1D-CNNs came as a powerful solution to address the abovementioned issue. They immediately garnered significant support from researchers across the domains such as early health condition diagnosis, structural health monitoring, and anomaly detection owing to their state-of-the-art performance credentials. Besides, the 1D-CNN has the upper hand as low-cost hardware implementation and fast real-time onboard execution are possible. The combined feature vector of size twenty-two, comprising the outputs of the auto-encoder model (sixteen outputs) and the outputs from the physics-based engine model (six outputs), is traversed through the 1D-CNN network for the purpose of fault detection and isolation. Our proposed 1D-CNN module consists of several convolutional blocks, each composed of combinations of 1D-convolutional and 1D max-pooling layers. The attributes influencing the efficiency of the 1D-CNN layer are as follows: feature map, the number of filters, kernel size, and strides. The output of the convolutional layer is processed through a Rectified Linear Unit (ReLU) activation function to decipher the curvilinear correlation between the inputs and the respective outcomes. The mathematical formulation for ReLU activation is described as follows:

$$ReLU(x) = max(x, 0) \tag{3.3}$$

After the convolutional layer, the max-pooling (or the down-sampling layer) aids in the fast convergence of the overall 1D-CNN model and enhances its generalizability. The max-pooling layer helps discover the most prominent feature from the vector output of the 1D convolutional layer. The attributes affecting the efficacy of the pooling layers are the filter size and the strides. In the proposed 1D-CNN model, some convolutional block consists of convolutional and 1D max-pooling layers, while others contain just the convolutional layer per the demand of the data, which the hyperparameter tuning process determines. After the convolutional blocks come the flatten layer which helps to reshape the tensor into a vector that matches the number of elements in the output tensor generated by the convolutional block. Subsequently, the flatten layer's output is then passed through a sequence of dense layers. These dense layers play a crucial role in diagnosing the fault and identifying the corresponding fault class. The dense layers contribute to the final detection and classification of faults through a series of computations and learned patterns. In essence, the 1D-CNN model frames the fault diagnostics problem as a four-class classification problem, where the respective classes are namely: Nominal (no fault) condition, fault due to injection pressure (Inj_Prs), fault due to a delay in injection duration (Inj_Dur), and the fault due to the change in the start of injection (SOI). Figure 5.6 portrays the architecture of the 1D-CNN model for fault diagnostics.

3.3.4 Computational Resources

The fault diagnostics model is trained using Tensorflow-Keras (Tensorflow GPU 2.9.1) in a machine of Lenovo ThinkPad X1 with Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, six cores with 12 logical processors, 16 GB RAM, and NVIDIA

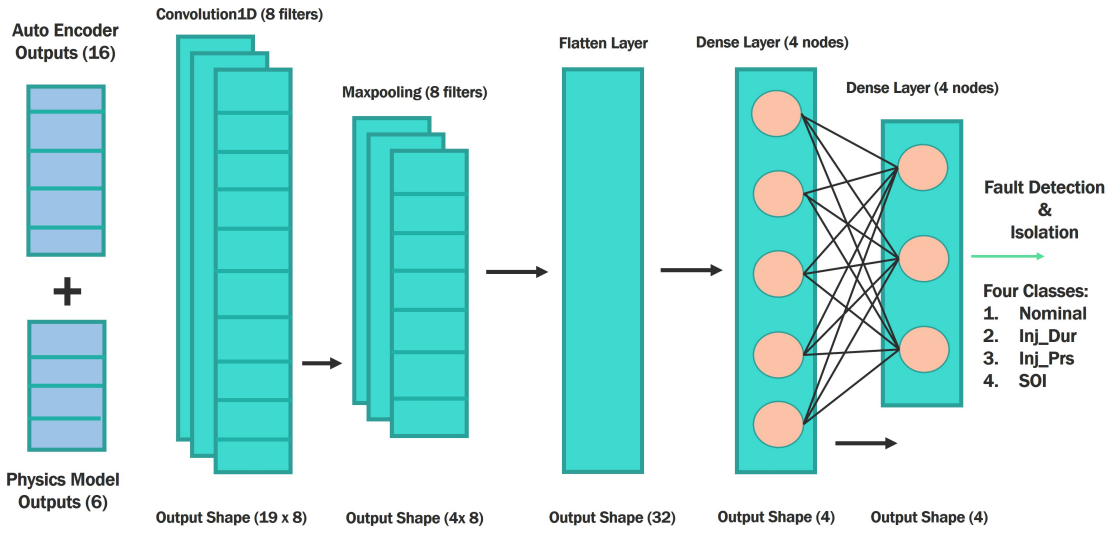

Figure 3.9: Hybrid 1D CNN-based Fault Diagnosis Model

GeForce GTX 1650Ti with Max-Q Design GPU.

3.4 Hybrid Model Evaluation

3.4.1 Evaluation Metrics

The fault detection and isolation problem can be framed as a multi-class classification problem. For this research work, we posed the problem statement as a four-class (or working conditions) classification problem, where the respective categories are as follows: Nominal (no fault) condition, fault due to injection pressure, fault due to a delay in injection duration, fault due to the change in the start of injection (SOI). In this paper, we used categorical cross-entropy as the loss function for classifying the four working conditions as discussed above. Equation 4.5 expresses the loss function for categorical cross-entropy:

$$-\sum_{i=1}^{C} y_{o,i} \log(p_{o,i}) \tag{3.4}$$

Where 'C' is the number of classes, 'log' is the natural logarithm, 'y' is the ground truth label (0 or 1) if the class label 'i' is the true prediction for the observation 'o', and 'p' is the prediction made by the model pertinent to class 'i'. To evaluate the performance of the proposed hybrid diagnostics model, we choose several evaluation metrics, which are enumerated as follows:

3.4.1.1 Confusion Matrix

The confusion matrix is a way of visualizing the classification performance of a model in a tabular format. It's a straightforward yet one of the most efficient tools to measure the performance of a classification model. The values in the matrix denote the number of accurate predictions made by the model when correctly or incorrectly classifying the respective classes. Figure 3.10 showcases the confusion matrix for a binary classification problem. Each column represents an instance in the actual class, while the entries in the rows depict the predictions made for the corresponding classes. Next, we list some of the crucial metrics as derived from the confusion matrix:

True Positive (TP): True positive refers to the number of correct predictions made by the model when the expected label was positive.

True Negative (TN): True negative refers to the number of correct predictions made by the model when the expected label was negative.

False Positive (FP): False positive refers to the number of instances where the model predicts the negative class as positive.

False Negative (FN): False negative refers to the number of instances where the model predicts the positive class as negative.

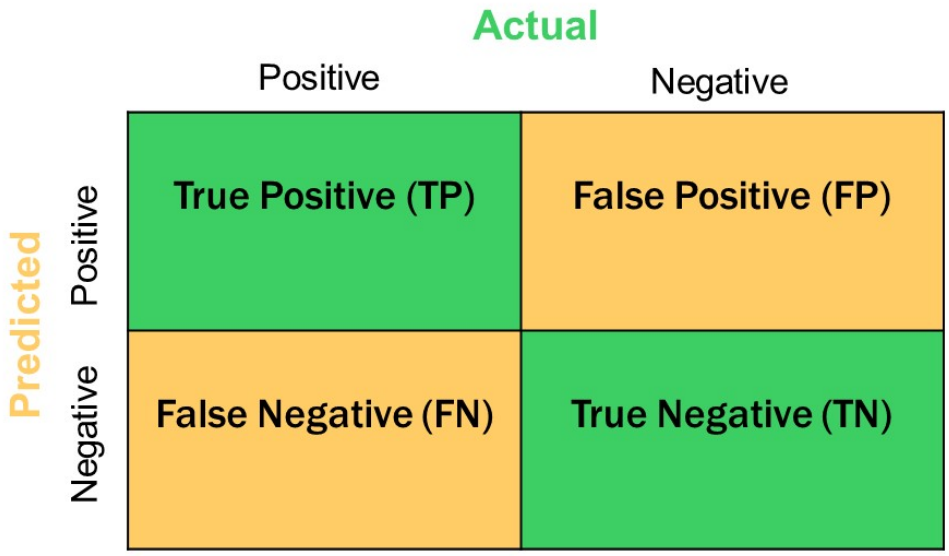

Figure 3.10: Confusion Matrix for Binary Classification.

3.4.1.2 Accuracy

Accuracy is one of the most widely accepted metrics for the performance quantification of a classification model. It measures the number of correct predictions made by the model out of the total available samples. Equation 3.5 expresses the accuracy in terms of entries of the confusion matrix. However, accuracy works best when the data has a uniform sample size for each class. In most practical scenarios, the data size distribution won't be uniform; that was the case with the diagnostics experiment data we collected. To address the issues mentioned above, we resort to other performance evaluation criteria, such as precision and recall, which are less affected by the imbalance in the training data samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.5}$$

3.4.1.3 Precision

Precision is the ratio of True Positive (TP) to the sum of True Positive (TP) and False Positive (FP). It alludes to the notion that the fraction of labels predicted as positive by the model is actually positive. Equation 3.6 expresses the precision as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.6}$$

3.4.1.4 Recall

Recall refers to the fraction of total actual positives which were predicted as positives by the model. Equation 3.7 expresses the mathematical formulation of recall. The recall is also known as the Probability of classification, True Positive Rate (TPR), or Sensitivity.

$$Recall = \frac{TP}{TP + FN} \tag{3.7}$$

3.4.1.5 Specificity

Specificity refers to the fraction of total actual negatives which were predicted as negatives by the model. Equation 3.8 expresses the mathematical formulation of specificity. The specificity is also referred to as the True Negative Rate (TNR).

$$Specificity = \frac{TN}{TN + FP} \tag{3.8}$$

3.4.1.6 F-1 Score

It gets tricky to correctly analyze a model's performance based on high recall and low precision or vice-versa. The key conceptual idea behind the conceptualization of the F-1 score is to define a proper quantification measure that can strike a balance between the two metrics, i.e., precision and recall, while bringing the best of both. The high value of the F-1 score not only elucidates the high classification capabilities of a model but also expounds the models' ability to give unbiased equal importance to different class labels under consideration. The F-1 score combines both precision and recall by calculating their harmonic mean and can be expressed via Equation 3.9 as follows:

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.9}$$

3.4.1.7 Micro F-1 Score

For our multi-class classification problem of identifying different faults, we used the micro F-1 score and calculated the total TP, FP, and FN using the various classes available. This metric does not consider individual classes. It estimates the values globally. After calculating the cumulative TP, FP, and FN, global precision and recall values are calculated, which ultimately leads to the micro F-1 Score. Since we calculate the values globally, we observe that all the metrics become equal, which implies that:

$$Recall = Precision = MicroF1 = Accuracy \tag{3.10}$$

3.4.1.8 Macro F-1 Score

For this metric, we calculated the values of TP, FP, and FN individually for each of the four conditions (1 nominal and three fault types). After that, we estimate the precision, recall, and F-1 metrics for the individual classes and then use an unweighted mean of the F-1 metric for the four classification labels to yield a macro F-1 score. Equation 3.11 represents the macro F-1 score in it's mathematical form:

$$\frac{\sum_{i=1}^{C} F1_i}{\sum_{i=1}^{C}} \tag{3.11}$$

In Equation 3.11, the numerator represents the summation of F-1 scores over the total number of classes 'C', and the denominator is the sum of the number of various classes present in the problem, which in our case was four.

3.4.1.9 Weighted F-1 Score

Unlike the macro F-1 score, the weighted one is calculated using weighted mean of the individual F-1 scores of the respective classification labels and can be expressed as follows:

$$\frac{\sum_{i=1}^{C} F1_i * (DataSamples)_i}{\sum_{i=1}^{C} (DataSamples)_i} \tag{3.12}$$

For Equation 3.12, the numerator is a summation of the multiplication between the F-1 scores of the respective classes and the corresponding number of samples for each class. The denominator is the sum of the number of data samples in each category.

3.4.2 Robustness to Noisy Data

During the actual operations of an automobile, the onboard engine sensors might get exposed to different types and levels of noise. The noise creeps into the sensor measurements from the electronics or the data-acquisition systems and is the leading cause of measurement errors. To simulate the working conditions of a production-class diesel engine, we added different noise levels to the data from the various test-bed (engine) sensors. Specifically, we added multiple levels of Additive White Gaussian Noise (AWGN) to the data samples. The noisy synthetic data samples are then passed through the pre-trained 1D-CNN diagnostics module. We added noise samples ranging from a five-percent standard deviation, of the respective signal, to three times the standard deviation. Specifically, we used five different noise levels: five percent of the standard deviation of the signal, fifty percent of the standard deviation, standard deviation, two times the deviation, and three times the standard deviation, respectively. The exercise helps appraise the model's robustness when subjected to noisy data, which is the norm in actual working conditions.

3.4.3 Hyperparameter Tuning

The process of hyperparameter tuning [26, 131] involves finding the optimal combination of hyperparameters for a learning algorithm. This quest aims to discover the best-performing model, which minimizes the predefined cost function on a training dataset. By systematically exploring different hyperparameter settings, hyperparameter tuning enables the identification of the most effective configuration that yields superior results. The number of layers, number of neurons in a specific layer, filter size, batch size, and learning rate are a few of the prominent hyperparameters to be considered when striving for the top-notch version of the model for a given

dataset. We used different hyperparameter combinations for the pure data-driven and hybrid fault diagnostics models. Besides, while training the model, we used regularization techniques, such as L1 and L2, to account for any overfitting. The details are elaborated on in the following subsections:

3.4.3.1 Tuning for Fault detection and isolation

We used the grid-search hyperparameter optimization technique for the hybrid physics-infused 1D-CNN-based fault diagnostics model. We have explored 216 different hyperparameter combinations employing a five-fold cross-validation, resulting in 1080 fits of the overall training dataset. The accuracy values, over the validation set, for a few of the sample hyperparameter configurations, are displayed in Figure 5.7. The model corresponding to the best hyperparameter combination is saved, and the same was later used to analyze the performance on noisy datasets. We used the Adam optimizer to train the model, with a learning rate of 0.01. For the 1D-convolutional and dense layers, we have used $ReLU$ as the activation function, while *Softmax* has been used as the activation for the outermost dense layer. Table 3.6 showcases the network topology of the best hybrid fault diagnostics model.

Similar hyperparameter optimization was performed for the pure data-driven 1D-CNN-based diagnostics model (without the physics-based engine model input), used to benchmark the hybrid model's performance. The hyperparameters considered are the batch size, epochs, learning rate, number of convolutional filters, kernel size, and the pool size for the max pooling layer. Out of the 216 combinations, box plots for a small sample set's accuracies are displayed in Figure 3.12. Here also, a five-fold cross-validation strategy was put to execution, which culminated in 1080 fits over the training dataset. We have used $ReLU$ as an activation function for the 1D-convolutional and dense layers. The outermost dense layer uses *Softmax* activation

Table 3.5: Network details of the 1D CNN module of the pure data-driven diagnostics framework

Layer Type	**Output Shape**	**#Parameters**
Conv1D	(None,13,16)	80
MaxPooling1D	(None,3,16)	0
Flatten	(None,48)	0
Dense	(None,4)	196
Dense	(None,4)	20
Total Parameters: 296 Trainable Parameters: 296 Non-Trainable Parameters: 0		

Table 3.6: Network details of the 1D CNN module of the hybrid diagnostics framework

Layer Type	**Output Shape**	**#Parameters**
Conv1D	(None,19,8)	80
MaxPooling1D	(None,4,8)	0
Flatten	(None,32)	0
Dense	(None,4)	132
Dense	(None,4)	20
Total Parameters: 192 Trainable Parameters: 192 Non-Trainable Parameters: 0		

function for the multiclass classification. The model used an Adam optimizer with a learning rate of 0.01. The network details of the best data-driven model are displayed in Table 5.1.

Table 3.7: Evaluation Metrics: Data-driven model

Micro Metrics		**Macro Metrics**		**Weighted Metrics**	
Metrics	**Values**	**Metrics**	**Values**	**Metrics**	**Values**
Micro Precision	0.93	Macro Precision	0.92	Weighted Precision	0.93
Micro Recall	0.93	Macro Recall	0.92	Weighted Recall	0.93
Micro F1-Score	0.93	Macro F1-Score	0.92	Weighted F1-Score	0.93

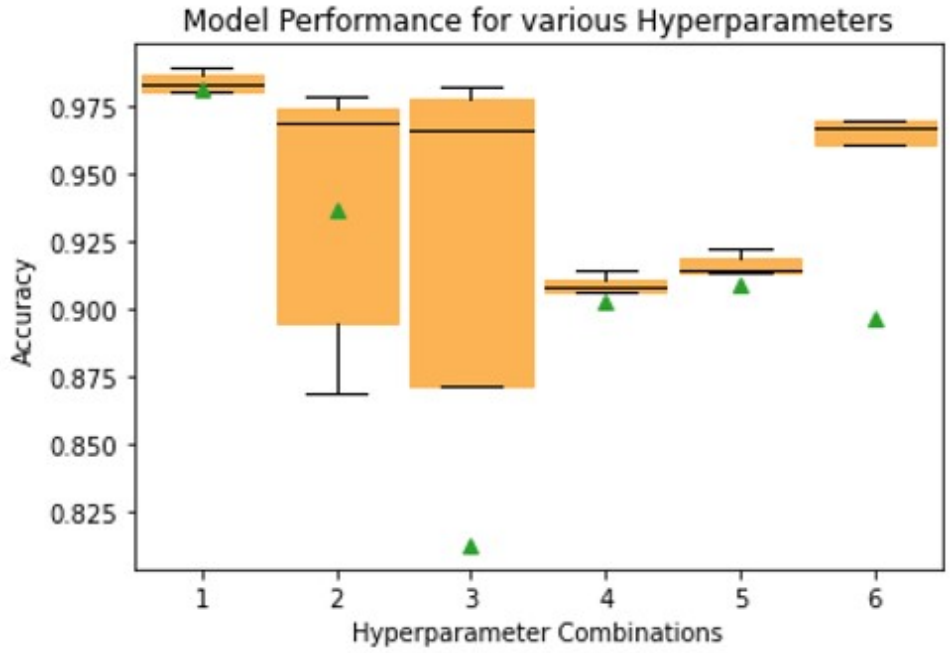

Figure 3.11: Hyperparameter tuning results for the hybrid model

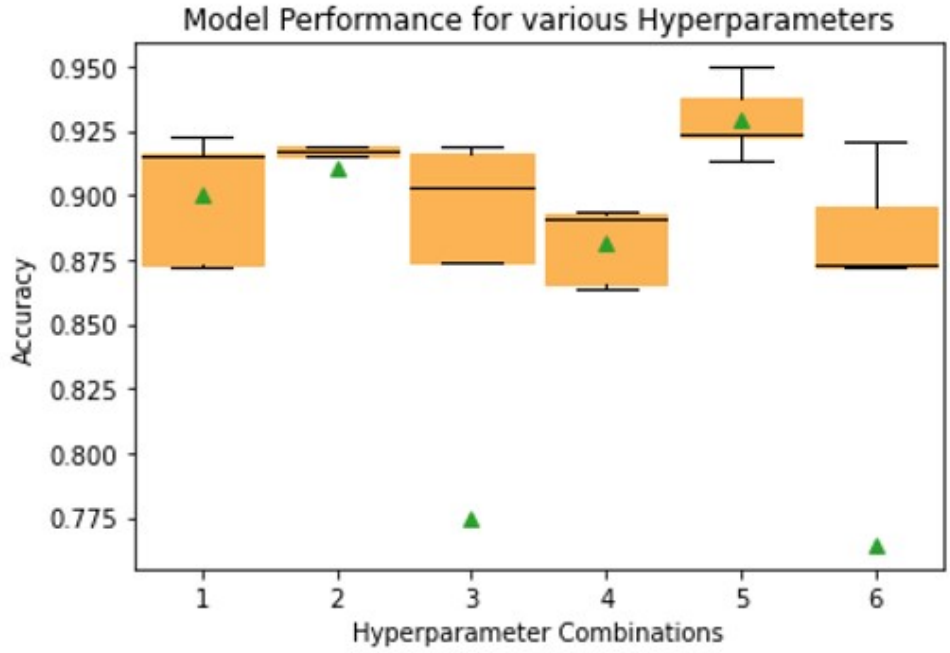

Figure 3.12: Hyperparameter tuning results for the pure data-driven

Table 3.8: Evaluation Metrics: Hybrid model

Micro Metrics		**Macro Metrics**		**Weighted Metrics**	
Metrics	**Values**	**Metrics**	**Values**	**Metrics**	**Values**
Micro Precision	0.98	Macro Precision	0.98	Weighted Precision	0.98
Micro Recall	0.98	Macro Recall	0.98	Weighted Recall	0.98
Micro F1-Score	0.98	Macro F1-Score	0.98	Weighted F1-Score	0.98

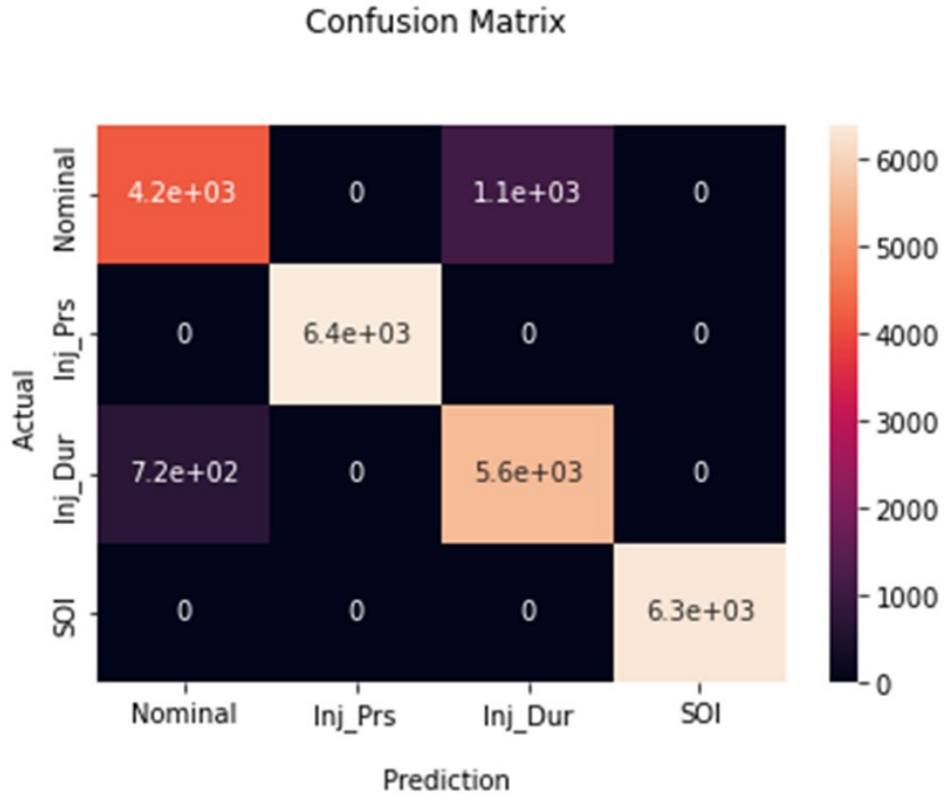

Figure 3.13: Confusion Matrix: Pure data-driven diagnostics model

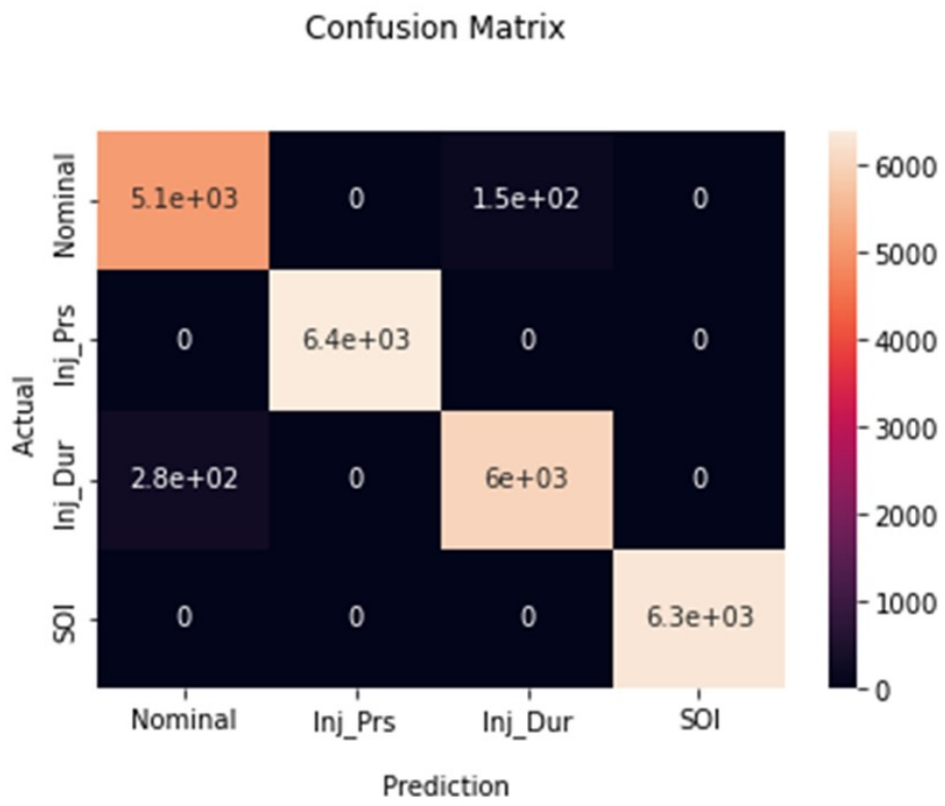

Figure 3.14: Confusion Matrix: Hybrid diagnostics model

Table 3.9: Comprehensive Diagnosis Report: Data-driven model

	Precision	**Recall**	**F1-Score**	**Number of Samples**
Nominal	**0.85**	**0.80**	**0.82**	**5262**
Injection Pressure	**1.00**	**1.00**	**1.00**	**6399**
Injection Duration	**0.84**	**0.89**	**0.86**	**6300**
SOI	**1.00**	**1.00**	**1.00**	**6318**
Accuracy			**0.93**	**24279**
Macro Average	**0.92**	**0.92**	**0.92**	**24279**
Weighted Average	**0.93**	**0.93**	**0.93**	**24279**
Accuracy: 0.93				

Table 3.10: Comprehensive Diagnosis Report: Hybrid model

	Precision	Recall	F1-Score	Number of Samples
Nominal	**0.95**	**0.97**	**0.96**	**5262**
Injection Pressure	**1.00**	**1.00**	**1.00**	**6399**
Injection Duration	**0.98**	**0.96**	**0.97**	**6300**
SOI	**1.00**	**1.00**	**1.00**	**6318**
Accuracy			**0.98**	**24279**
Macro Average	**0.98**	**0.98**	**0.98**	**24279**
Weighted Average	**0.98**	**0.98**	**0.98**	**24279**
Accuracy: 0.98				

3.5 Results and Discussions

3.5.1 Fault Detection and Isolation

When comparing Table 3.9 and Table 3.10, it is pretty evident that the physics-infused hybrid diagnostics model outperforms the pure data-driven diagnostics model on the accuracy front. Compared to the 93 percent accuracy of the data-driven model, the hybrid model achieves an overall accuracy of 98 percent on fault detection and isolation tasks. Not only accuracy, from the readings of Table 3.7 and Table 3.8, we can observe that on the various other metrics such as precision, recall, and F1-Score, and their respective variations such as micro, macro, and weighted, the hybrid model supersedes the data-driven one.

Besides, Figures 3.13 and 4.9 offer a more in-depth insight into the fault detection and isolation capabilities of the data-driven, and hybrid models, respectively. From the two confusion plots as mentioned above, we can discern that the hybrid model has a clear edge over the data-driven model while accurately classifying the four classes: nominal (no fault) condition, fault due to injection pressure (Inj_Prs), fault due to a delay in injection duration (Inj_Dur), and the fault due to the change

in the start of injection (SOI). More diagonal elements in the confusion matrix for the hybrid model stand witness to the superior diagnosis ability of the hybrid model compared to its data-driven counterpart. Also, Tables 3.9, 3.10 showcase that even for the individual fault classes, namely the nominal, fault due to injection pressure(Inj_Prs), fault due to injection duration (Inj_Dur), and the fault due to start of injection (SOI), the recall, precision, and F1-Scores are much better for the hybrid model as compared to the pure data-driven model. The better fault segregation capability of the hybrid model can be attributed to the physics model, which guides the parameter construction and tuning of the overall hybrid framework.

3.5.2 Robustness to noisy data

Figures 3.15, 3.16, 4.11, and 4.12 showcase the confusion matrices for the performance of the physics-infused model, as compared to the pure data-driven model, over datasets consisting of various noise levels ranging from five percent of standard deviation to two times the standard deviation of the original signal. It is patent from the confusion matrices that the proposed physics-infused hybrid diagnostics framework remains relatively stable even in noisy unprecedented input signals compared to the pure data-driven diagnostics model. The high robustness of the model connotes the ability to perform even when it's tough to get precise measurements from the onboard sensors, which deviates the readings from the expected values.

Figure 3.15 represents the confusion plot for the best performance of our pretrained hybrid model for the noise level of five percent of the standard deviation of the original signal. From the plot, we can observe that the number of misclassifications is still relatively low compared to the pure data-driven model, and the hybrid model registers an accuracy of 98 percent. Also, the model performed reasonably

well on the precision, recall, and F1-scores, as shown in Table 3.11, which presents a comparison between the comprehensive diagnosis reports of both the data-driven and the hybrid models for a noise level of five percent of the standard deviation. Both the data-driven and hybrid models show a decline in accuracy with the increasing noise levels, and models have the worst accuracy for the case where the noise level was equal to three times the standard deviation, as quite evident from the Figures 3.19, 4.14. However, the hybrid model demonstrates better fault detection and isolation capability for the same noise level compared to the pure data-driven model. From the Tables 3.11, 4.3, 4.4, 4.5, we can observe that even on the other metrics such as precision, recall, and F1-score, the hybrid model outperforms its data-driven counterpart, when subjected to varying noise levels. The improved ability of the hybrid model to handle noise can be attributed to its greater robustness, which stems from the incorporation of physics-based information that enhances its feature-capturing capabilities. Physics-based engine model aids the overall hybrid diagnostics framework to become sentient. Moreover, domain knowledge improves the generalizability of the hybrid fault diagnostics model. The enhanced generalizability comes to the rescue of the hybrid model when tested against out-of-distribution noisy data samples. Besides, the involvement of physics helps decipher underlying dependencies in the data, and it gives the hybrid physics-infused model an edge over the data-driven model when classifying the appropriate fault classes, even in the presence of noise.

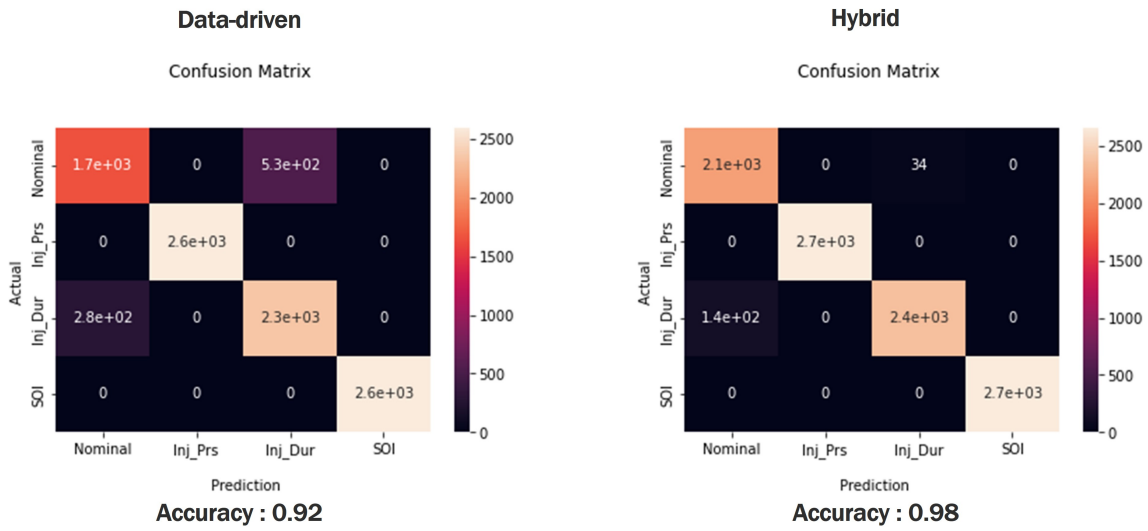

Figure 3.15: Comparative confusion matrix: Noise level 5 percent of standard deviation

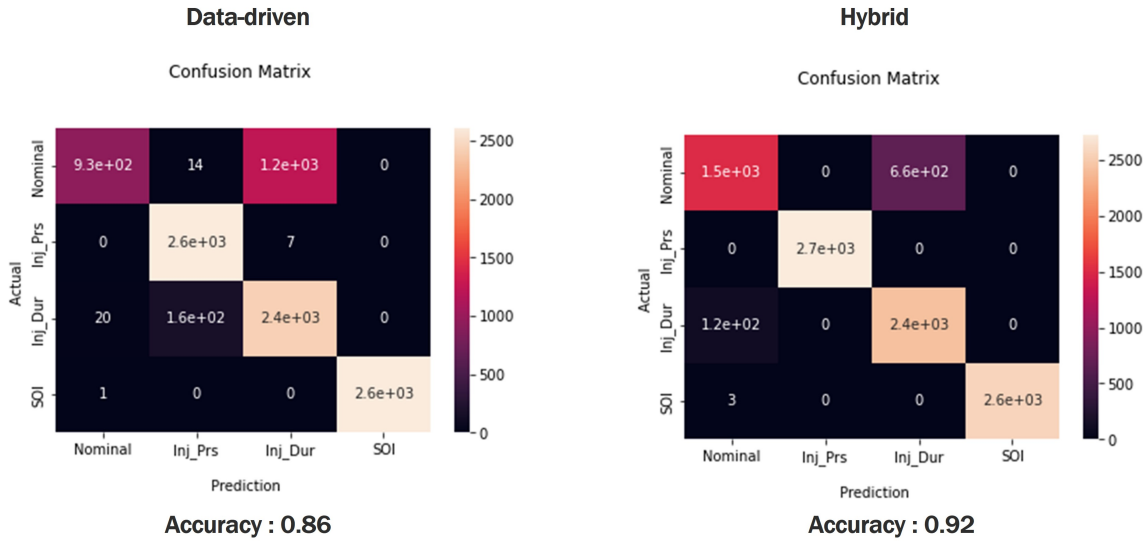

Figure 3.16: Comparative confusion matrix: Noise level 50 percent of standard deviation

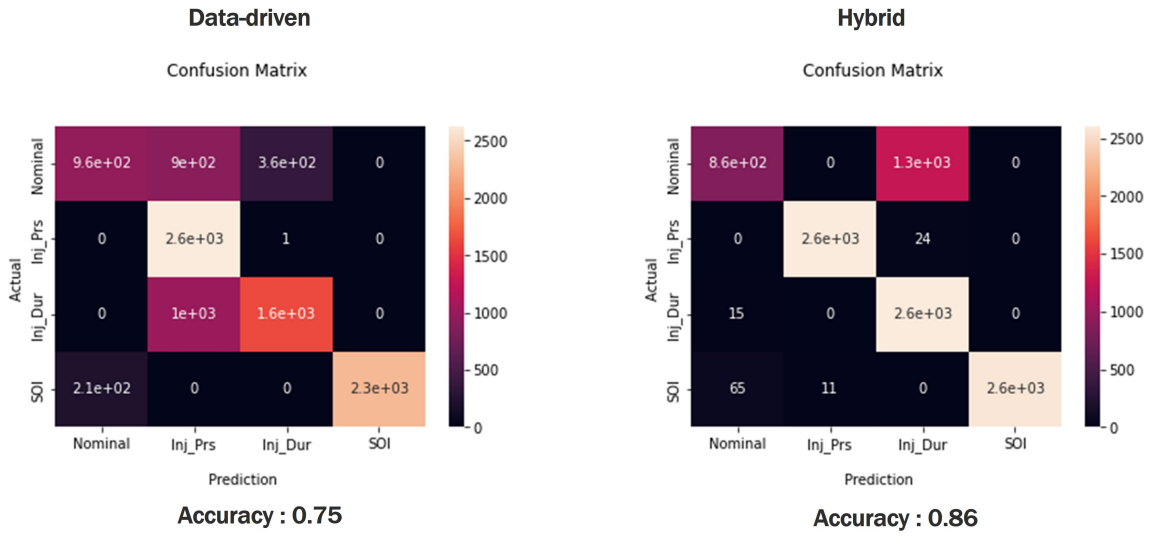

Figure 3.17: Comparative confusion matrix: Noise level equal to standard deviation

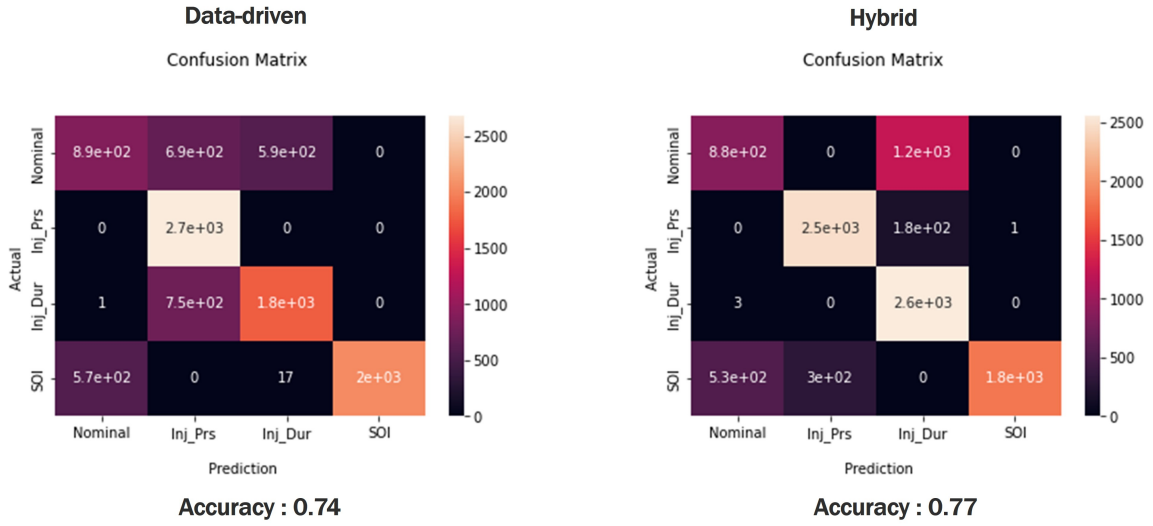

Figure 3.18: Comparative confusion matrix: Noise level-2 times of standard deviation

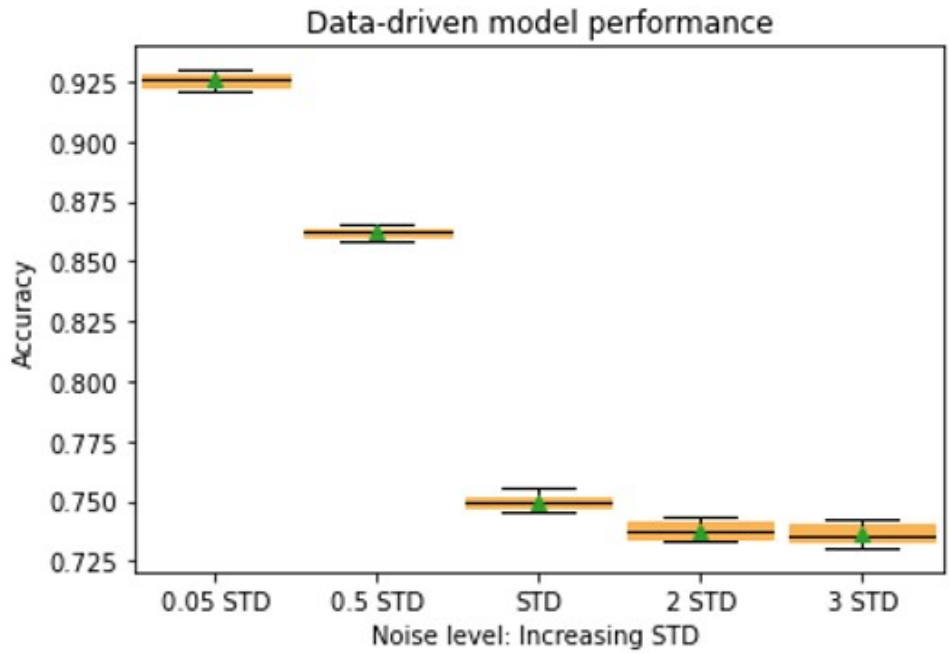

Figure 3.19: Comparative performance over different noise levels: Data-driven model

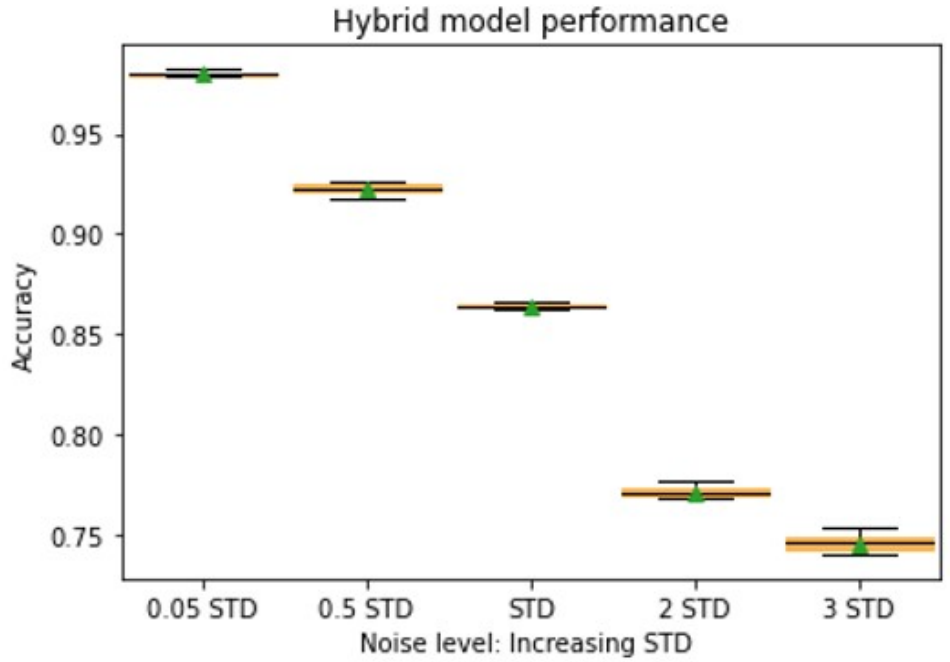

Figure 3.20: Comparative performance over different noise levels: Hybrid model

Table 3.11: Comparative comprehensive diagnosis report: Noise level 5 percent of standard deviation

	Data-driven Model			Hybrid Model		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Nominal	**0.86**	**0.76**	**0.80**	**0.94**	**0.98**	**0.96**
Injection Pressure	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Injection Duration	**0.81**	**0.89**	**0.85**	**0.99**	**0.94**	**0.96**
SOI	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Accuracy			**0.92**			**0.98**
Macro Average	**0.92**	**0.91**	**0.91**	**0.98**	**0.98**	**0.98**
Weighted Average	**0.92**	**0.92**	**0.92**	**0.98**	**0.98**	**0.98**
	Accuracy: 0.92			**Accuracy: 0.98**		

Table 3.12: Comparative comprehensive diagnosis report: Noise level 50 percent of standard deviation

	Data-driven Model			Hybrid Model		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Nominal	**0.91**	**0.43**	**0.60**	**0.92**	**0.69**	**0.79**
Injection Pressure	**0.94**	**1.00**	**0.97**	**1.00**	**1.00**	**1.00**
Injection Duration	**0.66**	**0.93**	**0.77**	**0.79**	**0.95**	**0.86**
SOI	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Accuracy			**0.86**			**0.92**
Macro Average	**0.89**	**0.84**	**0.83**	**0.93**	**0.91**	**0.91**
Weighted Average	**0.89**	**0.86**	**0.84**	**0.93**	**0.92**	**0.92**
	Accuracy: 0.86			**Accuracy: 0.92**		

Table 3.13: Comparative comprehensive diagnosis report: Noise level equal to standard deviation

	Data-driven Model			Hybrid Model		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Nominal	**0.82**	**0.43**	**0.57**	**0.91**	**0.41**	**0.56**
Injection Pressure	**0.58**	**1.00**	**0.73**	**1.00**	**0.99**	**0.99**
Injection Duration	**0.82**	**0.62**	**0.70**	**0.67**	**0.99**	**0.80**
SOI	**1.00**	**0.92**	**0.96**	**1.00**	**0.97**	**0.99**
Accuracy			**0.75**			**0.86**
Macro Average	**0.80**	**0.74**	**0.74**	**0.90**	**0.84**	**0.84**
Weighted Average	**0.80**	**0.75**	**0.74**	**0.89**	**0.86**	**0.85**
	Accuracy: 0.75			**Accuracy: 0.86**		

Table 3.14: Comparative comprehensive diagnosis report: Noise level equal to two times of standard deviation

	Data-driven Model			**Hybrid Model**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Nominal	**0.61**	**0.41**	**0.49**	**0.62**	**0.41**	**0.50**
Injection Pressure	**0.65**	**1.00**	**0.79**	**0.89**	**0.93**	**0.91**
Injection Duration	**0.74**	**0.70**	**0.72**	**0.64**	**1.00**	**0.78**
SOI	**1.00**	**0.78**	**0.87**	**1.00**	**0.69**	**0.81**
Accuracy			**0.74**			**0.77**
Macro Average	**0.75**	**0.72**	**0.72**	**0.79**	**0.76**	**0.75**
Weighted Average	**0.76**	**0.74**	**0.73**	**0.80**	**0.77**	**0.76**
	Accuracy: 0.74			**Accuracy: 0.77**		

Chapter 4

Hybrid Physics-Infused 1 Dimensional-Convolutional Neural Network (1D-CNN) based Ensemble Learning Framework for Diesel Engine Fault Diagnostics

This chapter presents a novel diesel engine fault diagnostics approach utilizing a physics-infused hybrid stacked generalization ensemble model. The model consists of various hybrid physics-infused submodels, each using an amalgamation of 1 Dimensional-Convolutional Neural Networks(1D-CNN) and a physics-based low-fidelity engine model. Each of the physics-infused 1D-CNN-based hybrid sub-models is fed with a combination of data from the engine test bed and the output from the physics-based engine model. The model adopts a weighted model averaging technique using a deep neural network (DNN) that assigns appropriate weights to each

hybrid physics-infused sub-model contribution to enhance the final outcome. The hybrid model can achieve better generalizability and performance on extrapolated datasets by integrating physics while efficiently handling non-linearity. Moreover, the proposed model can identify the optimal bias-variance trade-off, avoiding overfitting or underfitting scenarios. Empirical results demonstrate that our model is highly robust in the presence of noisy data, which is commonly encountered in the dynamic operations of a diesel engine system. The hybrid ensemble model combines data from diverse sensors and the corresponding outputs of the physics model in various ways to detect the presence of a fault. Additionally, it can isolate (localize) three distinct fault categories: '*Inj_prs*' fault resulting from changes in injection pressure, '*SOI*' fault caused by alterations in the start of injection, and '*Inj_dur*' fault caused by shifts in injection duration. The main contributions of this chapter are enumerated as follows:

1. We propose an innovative hybrid framework that integrates physics principles into a 1D-CNN-based ensemble deep learning model to detect faults in diesel engines. Our approach combines domain knowledge with end-to-end 1D-CNN-based methods. The ensemble model follows a stacked generalization approach, featuring two learning levels: level-0 and level-1 (meta-level). At level-0, multiple physics-infused 1D-CNN-based submodels operate in tandem. At level-1, a deep neural network (DNN) aggregates the predictions of the level-0 submodels through an automatic weighted average mechanism, thereby enhancing the predictive accuracy of the final outcome. This hybrid architecture is capable of processing raw data from the test bed and autonomously diagnosing fault scenarios without external intervention.

2. In order to assess the viability of the proposed hybrid ensemble model, an

extensive database [1] of diesel engine faults, named "NavicEngine," was meticulously constructed. Notably, this database encompasses a diverse spectrum of operational conditions. The NavicEngine database was employed for both the training and testing phases of the hybrid architecture, thereby facilitating a comprehensive evaluation of its performance.

3. The validity of the proposed hybrid architecture was established through a series of tests conducted across diverse operational scenarios. Notably, the model demonstrated commendable levels of accuracy as measured by a spectrum of evaluation metrics. Furthermore, the architecture exhibited robust stability and resilience, effectively mitigating the influence of noisy data samples on its performance.

4.1 Hybrid Ensemble Learning Fault Diagnostics Architecture

The proposed hybrid physics-infused 1D-CNN-based ensemble deep learning fault diagnostics framework, consisting of dual learning levels (level-0 and level-1 or Meta level), imbibes several components, including an autoencoder (AE), a physics model of the engine based on Simulink, a stack of few different variations of the 1D-CNN module constituting the level-0 learning, and a deep neural network (DNN) which forms the level-1 learning (Meta Level). The Auto Encoder network analyzes the signals gathered from the engine test-bed sensors and produces a condensed representation of the entire feature space known as a latent vector. The Simulink/MATLAB-based physics model processes the engine control parameters and produces relevant outputs. The resulting combined feature vector from the physics

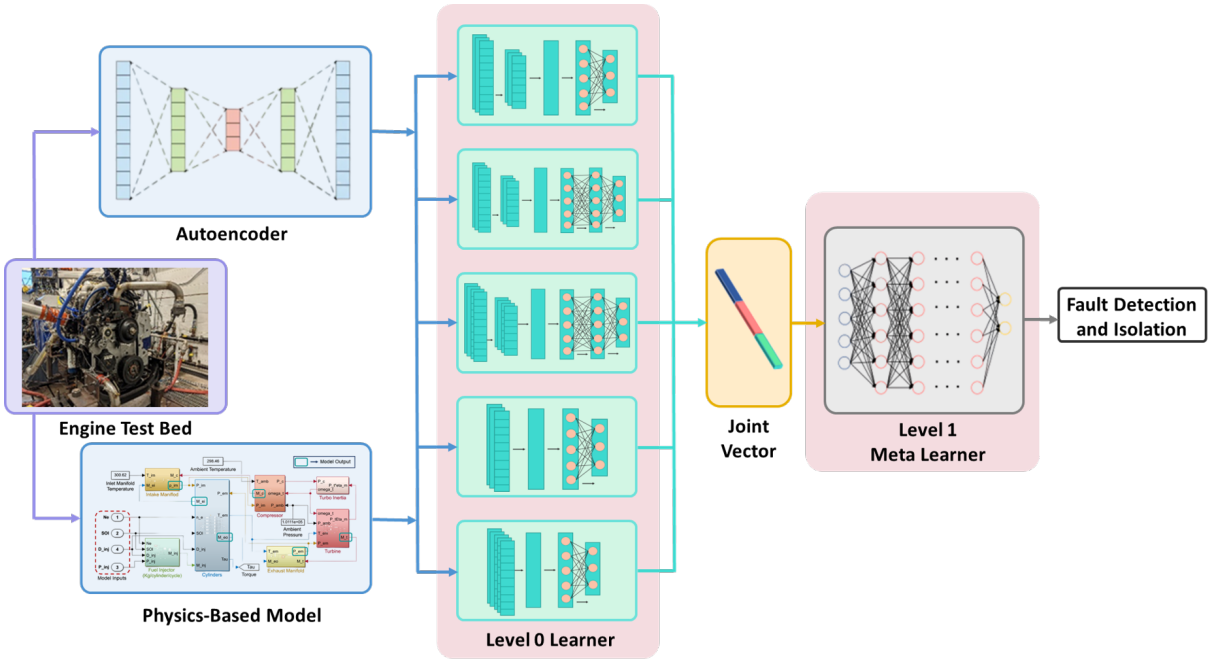

Figure 4.1: Hybrid Physics-Infused 1D-CNN-based Ensemble Learning Pipeline Fault Detection and Isolation

models and Auto Encoder is then passed through a sequence of 1D-convolutional blocks, each comprised of 1D-convolutional and 1D-max pooling layers as a part of level-0 of learning. The level-0 outputs are then combined as a vector and processed through the level-1 or Meta learner, the deep neural network (DNN), which ultimately results in the detection and isolation (classification of the fault type) of the fault. Figure 5.3 illustrates the comprehensive computational pipeline for fault diagnosis.

4.1.1 Autoencoder model

The autoencoder module of our hybrid architecture helps encode the high-dimensional space of sensor outputs from the experimental test bed to low-dimensional vectors apt for the following stacked 1D-CNN-based ensemble deep learning model to process information efficiently. Essentially, autoencoders are neural networks that

can learn the representation of high-dimensional data for the purpose of dimensionality reduction. An autoencoder comprises two components: an encoder module and a decoder module. The encoder module maps high-dimensional data to a lower-dimensional latent space while the decoder reconstructs the original input from the latent representation. An autoencoder aims to minimize the reconstruction loss between the initial intake data and the recreated output, enabling the latent representation to capture the crucial characteristics of the input data without significant information loss. Compared to conventional dimensionality reduction methods such as principal component analysis, autoencoders are more adaptable in their modeling capabilities and can model non-linear relationships among variables in data. One of the main advantages of using autoencoders is that unlike their conventional dimensionality reduction counterparts, such as Principal Component Analysis (PCA), autoencoders are more efficient in modeling the non-linear relationships in the data.

4.1.1.1 Model Architecture and Training

Autoencoders were trained on the preprocessed sensor data from the engine test bed to compress the combined INCA and PUMA variables into a low-dimensional vector space called the latent space, capturing the sensor data's essential properties. Figure 4.2 presents the architecture of the autoencoder network used in this research. This architecture maps 89 sensor readings (N) to a latent dimension space n. The encoder and decoder network configurations are mirror images of each other. The encoder is a neural network with the first hidden layer of 64 neurons and the second hidden layer consisting of 32 neurons. On the other hand, the decoder attempts to reconstruct the original input from the latent representation using a sequential network of two hidden layers with 32 and 64 neurons each. The autoencoder deploys the ReLU activation function to capture the underlying non-linear dynamics of the

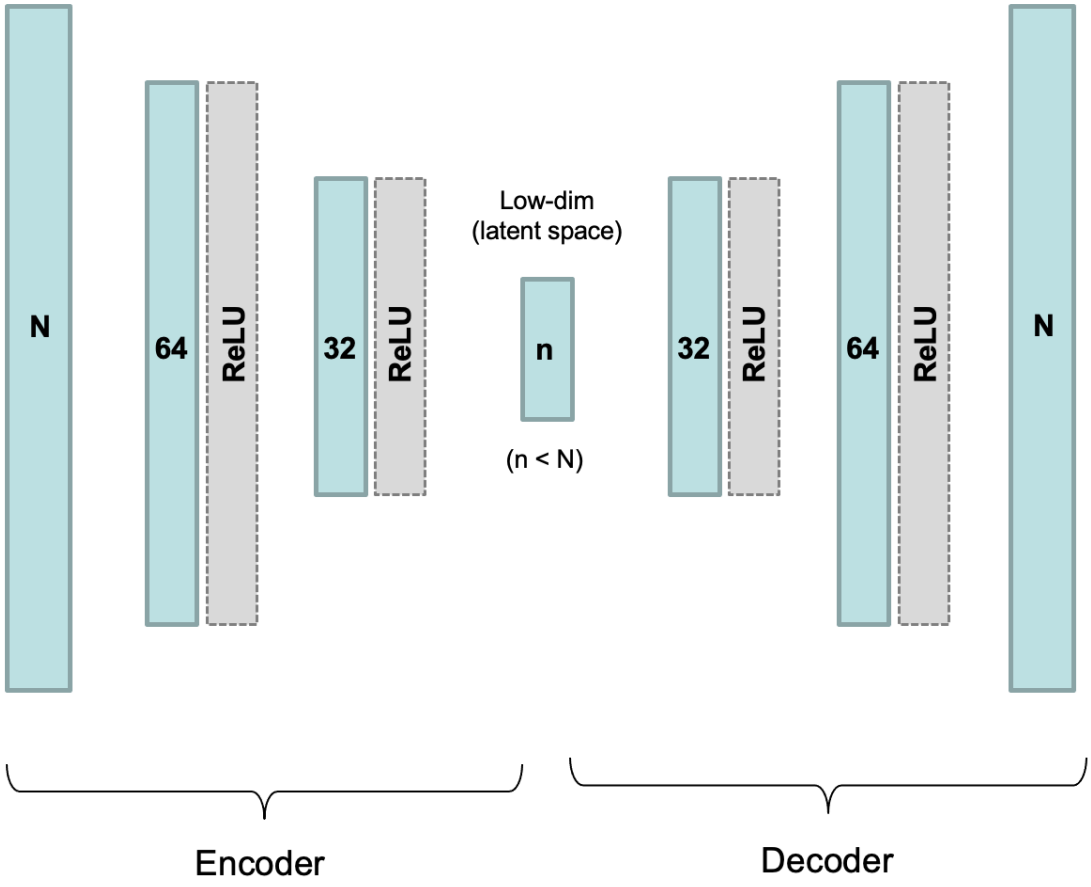

Figure 4.2: Overview of the autoencoder architecture.

data. We trained the model using a dataset comprising outputs corresponding to various speed levels under a nominal operational run of the engine to reconstruct the original input data accurately. Table 4.1 displays the performance of the trained autoencoder when reconstructing the original input data for various latent dimensions, where three versions, consisting of different sizes of the latent dimensions such as 16, 8, and 4, of the autoencoders were explored. As expected, the reconstruction performance of the autoencoder decreased with a decrease in the latent dimension, as information loss is an inherent issue in any dimensionality reduction technique. We used the root mean square error (RMSE) to measure the reconstruction performance.

latent dimension	reconstruction RMSE
16	0.005
8	0.006
4	0.011

Table 4.1: Autoencoder reconstruction results for different latent dimensions

4.1.1.2 Latent Dimensions

After training the autoencoder, we utilize the low-dimensional latent representation of the sensor readings in the subsequent stages of our hybrid diagnosis architecture to train the following 1D-CNN-based ensemble model. The autoencoder can capture the essential data characteristics in the latent space, a valuable data source for subsequent data-driven models. To visualize the properties of the latent space, we utilized t-distributed stochastic neighbor embedding (t-SNE), which allows us to visualize the high-dimensional input space by projecting it into lower dimensions while maintaining the inherent dataset properties and the interrelationship among the individual data samples. Figure 4.3 presents the visualization patterns corresponding to 1600 rpm, obtained using t-SNE, for the 16-dimensional latent space. The color bar on the side depicts the time duration for which the engine operates. In Figure 4.3, the green, orange, and blue clusters correspond to three distinct diesel engine operating conditions.

4.1.2 Physics-based model

4.1.2.1 Model Introduction

The wide range of a dataset plays a crucial role in the performance of the data-driven model for engine fault diagnostics, specifically in the dataset collected on various severity levels of the faults and operating conditions. Things get challenging

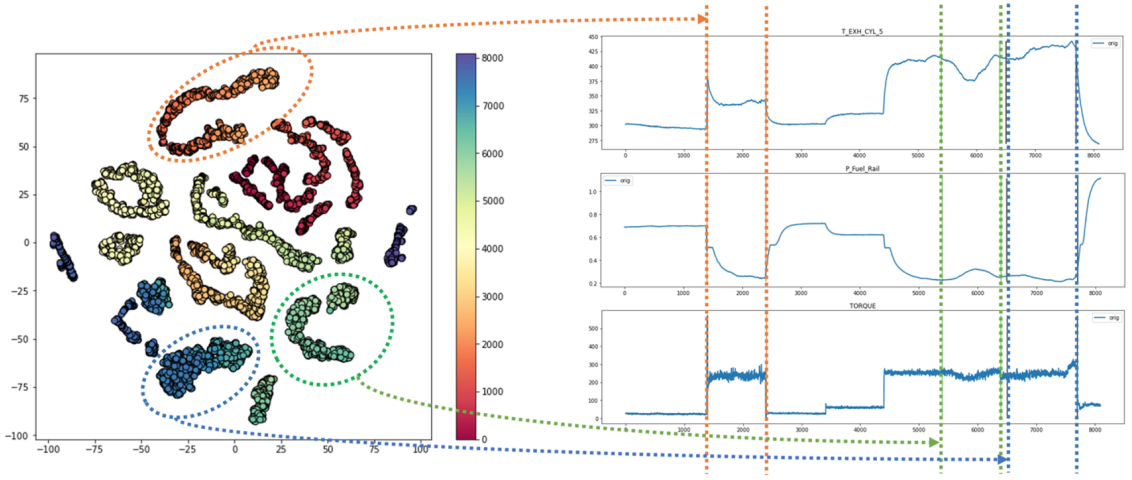

Figure 4.3: Visualization of the latent dimensional representation of the high-dimensional sensor readings for rpm 1600 instance.

while collecting the experimental data for faulty behavior. This is mainly due to running the engine in faulty conditions for all possible scenarios can cause serious harm to the engine's components. We collected the data for this paper in a very controlled manner and carefully observed every engine output to ensure all the components of the engine were in a healthy condition. However, this limited us to lower fault severity levels when collecting data. In practical situations, faults can occur without such restrictions, often significantly impacting the system, making it impossible to replicate in a test cell for data generation.

Due to this limitation, the inclusion of a physics-based model in a fault diagnosis system becomes important. Unlike the test bed engine, a physics-based model can function under critical and desired conditions. Combining a thoroughly analyzed physics-based model with a data-driven model makes it possible to resolve issues of sparse datasets and generalizability. This hybrid approach offers a more comprehensive and effective solution to engine fault diagnostics.

There are several physics-based models for the diesel engine that can be found in the literature, including Computational Fluid Dynamics (CFD), zero, one, and

two-dimensional models, to mention a few [8, 9, 74, 88, 107]. 0D models just have a time dependence, and do not take into account special dependencies. These models consist of ordinary differential and algebraic equations, and those equations are the function of only one variable, time. In contrast, models with one, two, or three spatial dimensions within the chosen coordinate system, referred to as 1D, 2D, or 3D models, are challenging to construct and require extensive domain expertise. These models typically exhibit a direct relationship between computational time and accuracy, making them a crucial factor in model selection. Ideally, for engine fault diagnosis, a model that can simulate faster than real-time on a regular Windows machine while providing insights into the behavior of the actual system (such as a diesel engine, in this scenario) would be desirable. This would require a balance between simulation speed and accuracy. To achieve a trade-off between accuracy and computational efficiency, our approach involves utilizing a 0D model while selectively developing high-fidelity subsystems of interest.

4.1.2.2 Model Details

In this paper, a 0D physics-based diesel engine model is proposed, which is developed for a 7.6-liter Navistar engine placed at Powertrain Laboratory, Clemson University. We develop this model for the engine specification mentioned in Table 3.1. The proposed model is developed using the Mean Value Engine Model (MVEM) technique, which involves using an effective value from the combustion cycle of the engine and neglecting its crank angle basis information [106]. There are numerous diesel engine models published in the literature for various applications [27, 28]. Particularly the model established by Wahlstrom and Eriksson [116] is considered a baseline model for our work and further improves the fidelity of certain subsystems. Wahlstrom constructed the mean value model that primarily describes the gas flow dynamics in a

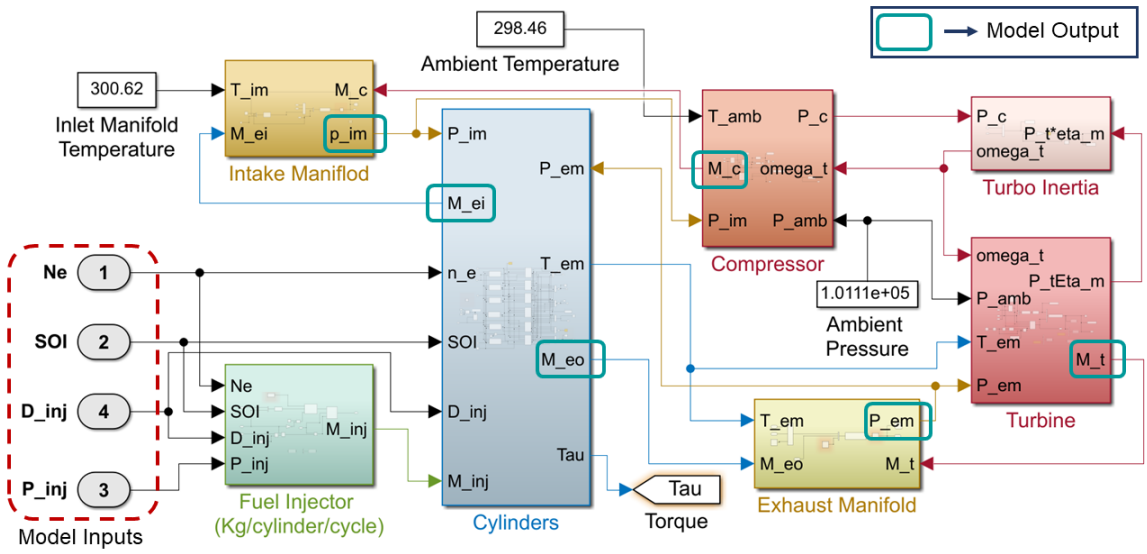

Figure 4.4: Structural diagram for the physics-based engine model. The green solid and red dashed rectangular box denotes output measurements and control input parameters, respectively.

computationally efficient manner. Figure 5.4 depicts an overview of the model, which captures temperature, pressure, and mass flow from several subsystems, as well as the total torque from the cylinder subsystem with fourteen parameters.

The inputs for the physics-based diesel engine model are mentioned below:

1. Injection pressure (MPa) - P_{inj}
2. Injection duration (ms) - D_{inj}
3. Start of injection (Crank angle degree - CAD) - SOI
4. Engine speed (rpm) - Ne

The details for individual subsystems are described in the following subheadings.

Fuel injection subsystem

The test cell engine has a high-pressure common-rail fuel injection system, and we

model this fuel injection system using the principle of conservation of mass. The equation for calculating the injection flow rate is mentioned below,

$$M_{inj} = \alpha.sign(P_{inj} - P_{cyl}).c_d.A_{flow}\sqrt{\frac{2}{\rho}|P_{inj} - P_{cyl}|} \tag{4.1}$$

We add α in the above equation to take care of the injection duration. Here, α is a binary value, and it is effectively activating the equation whenever we want to inject fuel into the cylinder. All the remaining parameters from Equation 4.1 are mentioned in the appendix.

Cylinder subsystem

The accuracy of the engine model greatly depends on the fidelity of the combustion model, which is a crucial aspect of any engine. To achieve high fidelity, we employ a thermodynamic cylinder model that simulates the in-cylinder processes for crank angle degrees over an entire engine cycle. This thermodynamic model captures various parameters, including exhaust temperature, in-cylinder pressure, intake and exhaust mass flows, and torque for a complete cycle. As it is a mean value model, we are taking the averages for all the parameters in each cycle. The test cell has a six-cylinder engine, and to mimic the same setup in the model, we use the multi-cylinder concept. This multi-cylinder model features six distinct cylinders and fuel injectors. This allows us to incorporate and control the faults in individual cylinders.

Compressor, turbine, and manifolds subsystems

The modeling of the intake and exhaust manifolds involves the intake manifold pressure and exhaust manifold pressure states. The modeling of manifold pressure relies on the core concepts of mass conservation and the ideal-gas law. The following differential equations are employed to compute the pressure within the manifolds.

$$\frac{d}{dt}P_{im} = \frac{R_a T_{im}}{V_{im}}(M_c - M_{ei}), \tag{4.2}$$

$$\frac{d}{dt}P_{em} = \frac{R_e T_{em}}{V_{em}}(M_{eo} - M_t) \tag{4.3}$$

All the parameters from Equation 5.5, Equation 5.6, and Figure 5.4 are mentioned in the appendix. The turbocharger comprises of compressor, turbine, and turbo inertial model. This turbocharger model is extracted from Wahlstrom and Eriksson model [116].

We calibrated the physics-based model on the experimental data as described in Table 3.2. The range that the model is calibrated is between the load and engine speed of 100-550 Nm and 1000-1800 RPM, respectively. The model is calibrated by minimizing the error between the experimental data and the physics-based model data using the least-square optimization method [72]. The validation results are mentioned in Figure 5.5. These validation plots show the comparison between physics-based model data and experimental data for the engine speed of 1600 RPM. The physics-based data is represented by the blue line, while the experimental data is represented by the red line. The Mean Percent Error (MPE) for each plot is stated in their title. The physics-based model shows an error of less than 7% with the experimental data. We use six outputs from the physics-based model for the 1D-CNN, as these six outputs overlap with the experimental data. Thus, a total of twenty-two outputs are fed to the 1D-CNN, where sixteen outputs are from the autoencoder model, and six are from the physics-based model.

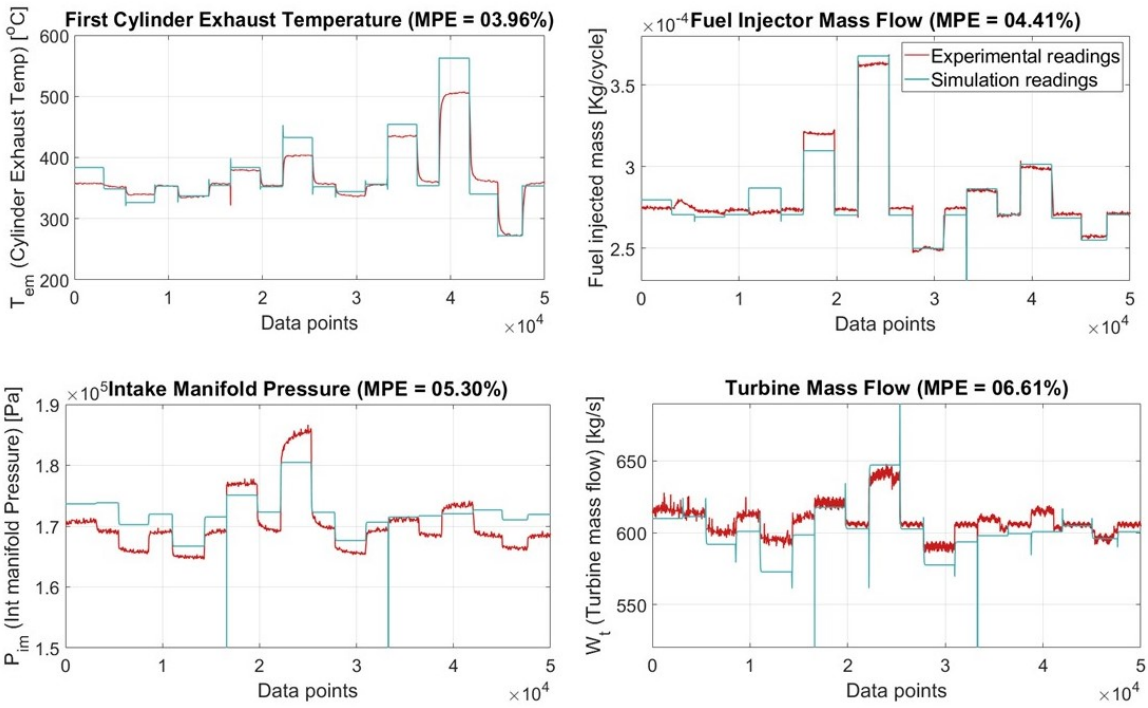

Figure 4.5: Physics-based model validation plots

4.1.3 1D-CNN-based Ensemble Deep Learning Model

The proposed method is a hybrid 1D-CNN-based ensemble deep learning model incorporating physics and following a stacked generalization technique. The model employs a weighted model averaging strategy, assigning suitable weights to the contribution of each sub-model, thereby enhancing the outcome. The proposed hybrid ensemble model, showcased in Figure 5.6, follows a stacked generalization approach and consists of the following levels of learning: Level-0 and Level-1 (Meta Level). The concept of stacked generalization [122] involves identifying and correcting the biases of generalizers based on a given set of data. This process involves creating a secondary space (meta level of learning) in which the generalizers' (level-0 learners) guesses are fed as inputs based on their training with a portion of the learning set, and their output is evaluated against the correct answer. By analyzing the discrepan-

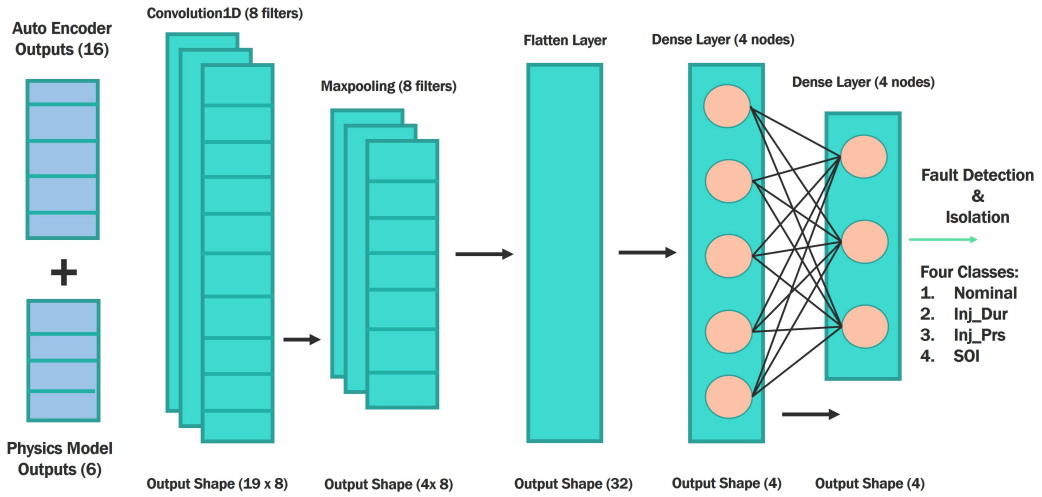

Figure 4.6: One of the Hybrid 1D CNN-based Fault Diagnosis Sub Model in Level-0 of Learning

cies between the two data sets, the original generalizers' biases (level-0 learners) can be identified and corrected. Each level of learning consists of neural network-based submodels, and the details of each level of learners are described as follows:

4.1.3.1 *Level-0 Learners:*

The various submodels at this level deal directly with the input data and make predictions. Level 0 of the ensemble architecture consists of five hybrid physics-infused 1D-CNN models. Each of the five models receives the sensor data from the engine test bed and the corresponding outputs from the multiphysics-engine model. To improve the processing capabilities of the level-0 1D-CNN network models, we utilized an autoencoder (AE) block as a preprocessing step to decrease the dimensionality of the test bed data, owing to its high dimensionality and numerous sensor outputs. The AE model's latent space outputs and the physics model outputs are used to create a combined feature vector traversed through the layers of the 1D-CNN-based submodels of level-0 of the ensemble learning model to generate the desired outcome. By guid-

ing the construction and parameter tuning of the neural network, the physics-based model enhances its ability to handle out-of-distribution data samples and makes it more robust and adaptable. Conversely, the submodel's data component aims to compensate for any approximations or lumped parameter assumptions made during the development of the partial physics-based engine model. The interdependent nature of the physics and data-driven components leads to the best of both worlds and can outperform conventional data-driven and model-based techniques when diagnosing diesel engine faults.

The individual hybrid physics-infused sub-models consist of 1D-CNNs, and the reason behind choosing 1D-CNNs as the data-driven component is as follows. 1D-CNNs have gained enormous popularity recently due to their ability to process sequential data effectively. This includes time-series data, natural language processing (NLP), and DNA sequences. Besides, the 1D CNNs have several advantages over other types of neural networks. They are particularly good at capturing local dependencies within the input data. Additionally, they require fewer parameters than other types of neural networks, which can make them more computationally efficient. As a component of the hybrid model, it helps extract essential features from the data, such as peaks, oscillations, or other patterns that may indicate physical phenomena. These features of the 1D-CNNs make it possible to create more accurate and physically meaningful hybrid physics-infused machine learning models, even with limited data. We utilize a consolidated feature vector of size twenty-two to detect and isolate faults, including sixteen outputs from the auto-encoder model and six from the physics-based engine model. This feature vector is fed into our proposed 1D-CNN module, which is comprised of multiple convolutional blocks. Each block is made up of 1D-convolutional and 1D max-pooling layers in various combinations. To optimize the performance of the 1D-CNN layer, numerous attributes are taken into account,

such as the number of filters, feature map, length of strides, and kernel size. These attributes have a significant impact on the efficiency of the layer. After the convolutional layer generates an output, it undergoes further processing through a Rectified Linear Unit (ReLU) activation function. This function is utilized to construe the non-linear relationship that exists between the input values and their corresponding outputs. The ReLU activation function can be expressed as follows:

$$ReLU(x) = max(x, 0) \tag{4.4}$$

Following the convolutional layer, the down-sampling layer, also known as the max-pooling layer, plays a crucial role in the rapid convergence of the entire 1D-CNN model while improving its generalization capability. By processing the vector output of the convolutional layer, the 1D max-pooling layer helps identify the most significant feature. The filter size and strides influence the effectiveness of the pooling layers. In the proposed 1D-CNN model, some convolutional blocks comprise both convolutional and max-pooling layers, whereas others contain only the convolutional layer as per the data requirements, which are determined by the hyperparameter tuning process. Once the convolutional blocks have been processed, the flatten layer converts the tensor into a vector containing the same number of elements as the output tensor obtained from the convolutional block. The output of the flatten layer is then fed through a sequence of dense layers that ultimately help diagnose the fault and identify the appropriate fault category. The 1D-CNN sub-models are designed to address the fault diagnostics problem by framing it as a four-class classification task. The four classes are as follows: Nominal (which represents the absence of any fault), Inj_Prs (which indicates a fault caused by injection pressure), Inj_Dur (which represents a fault resulting from a delay in injection duration), and SOI (which denotes a fault

caused by a change in the start of injection). To aid in understanding the architecture of the 1D-CNN model for fault diagnostics, Figure 5.6 is presented below, which shows the architecture of one of the five submodels at level-0 of learning. The figure illustrates the various components of the model and their interconnections. The network details of the five different physics-infused 1D-CNN models, in the level-0 of learning, are enumerated in Figure 4.7. The entire dataset is divided into three parts: *part A*, which is used to train the five different physics-infused 1D-CNN-based models, *part B*, which is used to train the meta or level-1 learner; and *part C*, which also serves as the test set to evaluate the performance of the hybrid physics-infused 1D-CNN-based ensemble learning model. The various physics-infused sub models in level-0 are trained using the *part A* data via a five-fold cross validation strategy.

4.1.3.2 *Level-1 (Meta) Learner:*

In our proposed ensemble architecture, the meta learner is a deep neural network (DNN), which is trained on the predictions made by the submodels (five different variations of the physics-infused 1D-CNN-based model). The trained submodels from level-0 of the ensemble architecture are used to make predictions on the part B dataset, which is separate from the ones these models were trained on. These predictions are stacked as a vector fed to the DNN model (meta learner). The meta learner is trained on this data against the corresponding labels. The trained ensemble model (level-0 and level-1 learners combined) is then used to make predictions on the test set (*part C data*). The output of the ensemble model is the four fault classes, including the Nominal (no-fault), Inj_Prs (fault associated with injection pressure), Inj_Dur (fault associated with injection duration), and SOI (fault associated with the start-of-injection). The unique training process of the meta learner helps prevent overfitting issues in the overall ensemble architecture. As a result, the proposed hy-

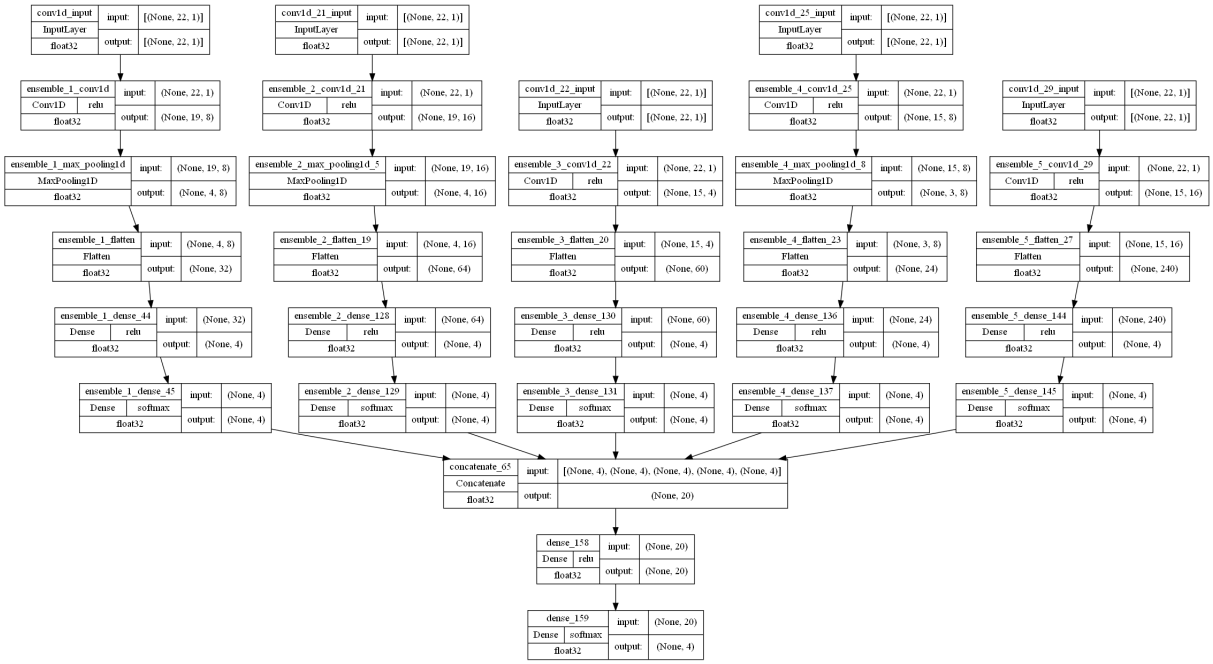

Figure 4.7: Network Structure of the Ensemble Model (including the level-0 submodels and the level-1 meta learner)

brid ensemble model can achieve better performance efficiency than the conventional data-driven and model-based approaches, especially when handling noisy sensor data and large unstructured state space. The dual level of feature learning and error mitigation aids the hybrid physics-infused 1D-CNN-based ensemble model helps to detect (the presence of fault) and isolate (classify the type of fault) the engine faults in an efficient manner. The network details of the meta learner are also outlined in Figure 4.7.

4.1.4 Computational Resources

The model for fault diagnostics has been trained on a Lenovo ThinkPad X1 with an Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, equipped with six cores and 12 logical processors, coupled with 16 GB of RAM and a NVIDIA GeForce GTX 1650Ti GPU featuring Max-Q Design, utilizing Tensorflow-Keras (Tensorflow GPU

2.9.1).

4.2 Evaluation Metrics

This research tackled the fault diagnosis issue by dividing it into four distinct working conditions. These conditions are Nominal (no-fault), Injection Pressure fault (Inj_Prs), Injection Duration fault (Inj_Dur), and the fault resulting from a modification in the initiation of injection (SOI). In multiclass classification problems, the goal is to predict the class of a given input from a set of multiple possible classes. The evaluation metrics for multiclass classification problems are used to assess the accuracy and performance of the classification model. To classify the four fault conditions mentioned above, we utilized categorical cross-entropy as our loss function. Equation 4.5 depicts the expression for this loss function. By using this method, we were able to effectively categorize the different working conditions and identify any faults that may be present.

$$-\sum_{i=1}^{C} y_{o,i} \log(p_{o,i}) \tag{4.5}$$

Where 'C' is our classification model's total number of classes. To calculate the loss function, we use the natural logarithm (log) function in conjunction with the ground truth label 'y' and the predicted label 'p' for a particular observation 'o'. Specifically, 'y' will be either 0 or 1 depending on whether the predicted class 'i' is correct for the observation 'o'. There are several evaluation metrics that can be used for multiclass classification problems, including confusion matrix, accuracy, precision, recall, and F1 score.

4.2.1 Noisy Data Generation

The onboard engine sensors can be exposed to varying forms and noise levels during diesel engine operations. This noise infiltrates the sensor measurements through electronic components or data acquisition systems, emerging as the primary source of inaccuracies. To replicate the operational environment of a typical production-grade diesel engine, diverse noise levels were introduced to the data obtained from different test-bed engine sensors. Specifically, the data samples were infused with varying degrees of Additive White Gaussian Noise (AWGN). Subsequently, the resultant noisy synthetic data samples were channeled through a pre-trained hybrid physics-infused 1D-CNN-based ensemble learning model. The injected noise samples ranged from five percent of the corresponding signal's standard deviation to three times that standard deviation. Five distinct noise levels were employed: five percent of the signal's standard deviation, fifty percent of the standard deviation, the standard deviation itself, twice the standard deviation, and thrice the standard deviation. This endeavor evaluated the model's resilience when exposed to noisy data, a common scenario in actual operational conditions.

4.3 Hyperparameter Tuning

Hyperparameter tuning [53, 112, 121] is handpicking the optimal hyperparameters of a machine learning model to achieve the best performance on a given task. Hyperparameters are parameters set before the model training and influence the model's behavior, such as learning rate, regularization strength, number of layers, and number of hidden units. Hyperparameter tuning is vital in developing vigorous and precise machine learning models. The process typically involves selecting a range of values for each hyperparameter and then testing the model with different

combinations of hyperparameters.

In our study, we employed the grid-search hyperparameter optimization method to enhance the performance of the fault diagnostics model. The model under consideration is a hybrid physics-infused 1D-CNN-based stacked generalization ensemble learning framework. The ensemble model comprises two levels of learners: level-0 and level-1 or meta learner. Level-0 learner consists of different variations of the physics-infused 1D-CNN model. Level-1, or the meta learner, is a dense neural network model which learns to combine the predictions of the level-0 learners in the best possible manner. As there is no set rule for deciding the number and variations of submodels in the level-0 of the ensemble model, we tried four configurations for the level-0 learner with different numbers, ranging from three to six, of hybrid physics-infused 1D-CNN submodels. For each of the level-0 configurations, we explored different variations of the dense neural networks (DNNs) as the level-1 learners using hyperparameter tuning and a five-fold cross-validation technique. The optimal ensemble model came out to be the one with five different variations of the physics-infused 1D-CNN model in the level-0 and a dual-layer DNN in the level-1 or meta learner. The optimal hybrid physics-infused 1D-CNN-based ensemble model registered an accuracy of 99 percent on the test set when detecting and isolating the faults.

We thoroughly investigated 216 distinct hyperparameter combinations, for each of the four variations of the level-0 learner (with different numbers of 1D-CNN submodels), during our experimentation by employing a five-fold cross-validation method, resulting in 1080 fits on the training dataset. The hyperparameters considered are the number of epochs, batch size, learning rate, number of dense layers in the level-1 learner, and the type of optimizer. Figure 5.7 presents the accuracy values for various sample hyperparameter configurations from the tuning process of the optimal ensemble model with five 1D-CNN models in level-0 and a dual-layer DNN in

level-1 over the validation set. We saved the hybrid physics-infused 1D-CNN-based ensemble model corresponding to the optimal hyperparameter combination for further analysis. We then evaluated its performance on noisy datasets to gauge its robustness. The DNN model of the meta-learner was trained using the Adam optimizer, utilizing a learning rate of 0.001. For the 1D-convolutional and dense layers, we opted for the $ReLU$ activation function, while the outermost dense layers of the level-0 and the level-1 (meta-learners) employed the *Softmax* activation. The detailed network topology, along with the activations of each layer, of the optimal configuration ensemble model, including the level-0 and level-1 learners, is illustrated in Figure 4.7.

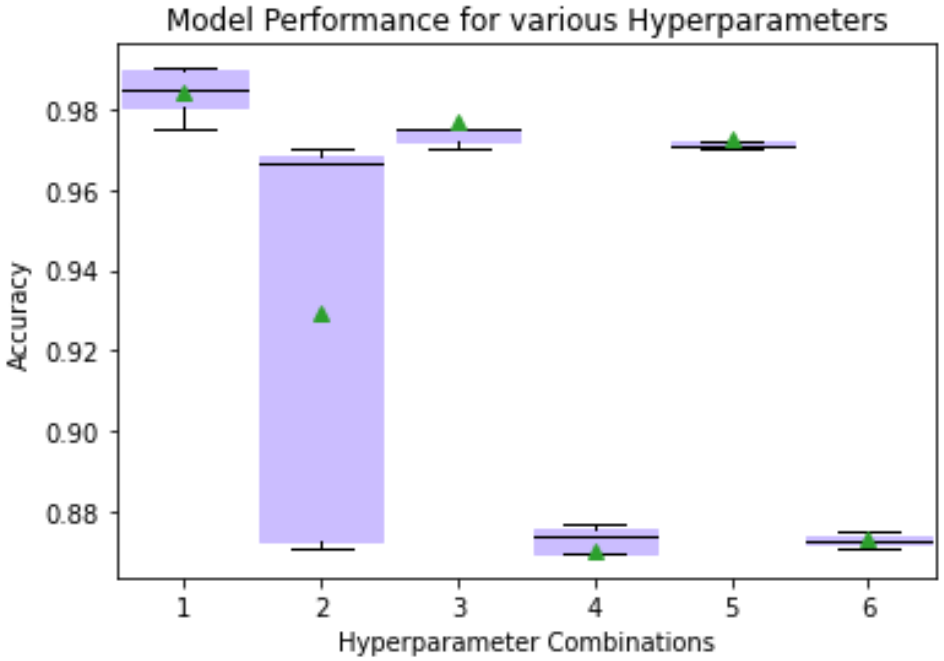

Figure 4.8: Hyperparameter tuning results for the hybrid ensemble model

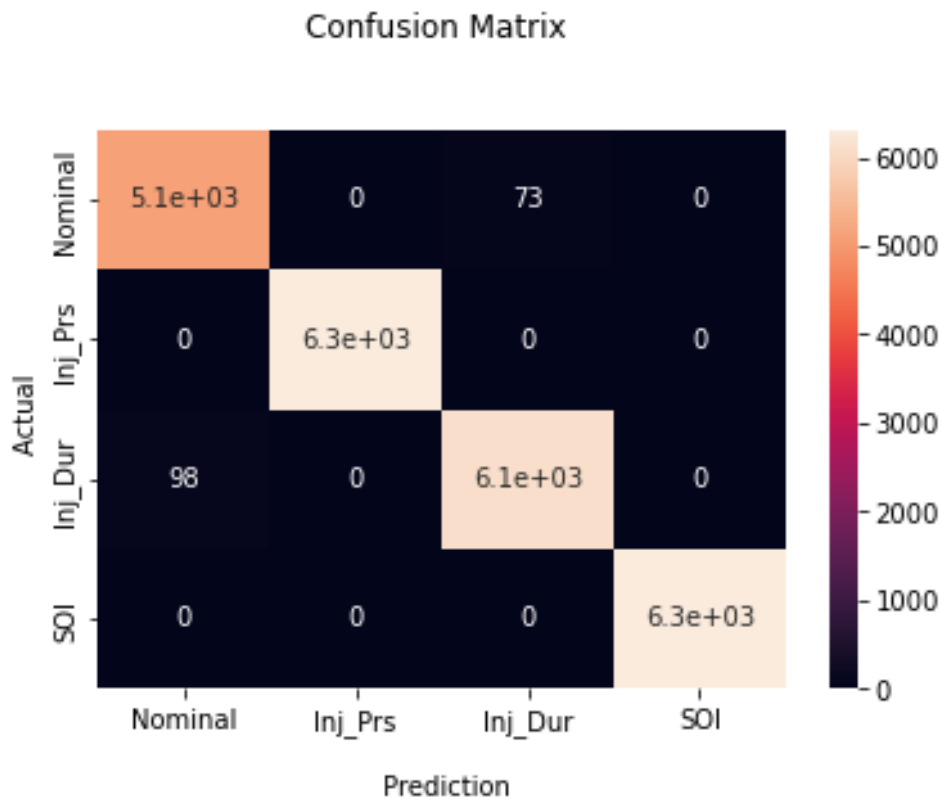

Figure 4.9: Confusion Matrix: Hybrid ensemble diagnostics model

Table 4.2: Comprehensive Diagnosis Report: Hybrid Ensemble model

	Precision	**Recall**	**F1-Score**	**Number of Samples**
Nominal	**0.98**	**0.99**	**0.98**	**5193**
Injection Pressure	**1.00**	**1.00**	**1.00**	**6312**
Injection Duration	**0.98**	**0.99**	**0.98**	**6192**
SOI	**1.00**	**1.00**	**1.00**	**6303**
Accuracy			**0.99**	**24000**
Macro Average	**0.99**	**0.99**	**0.99**	**24000**
Weighted Average	**0.99**	**0.99**	**0.99**	**24000**
Accuracy: 0.99				

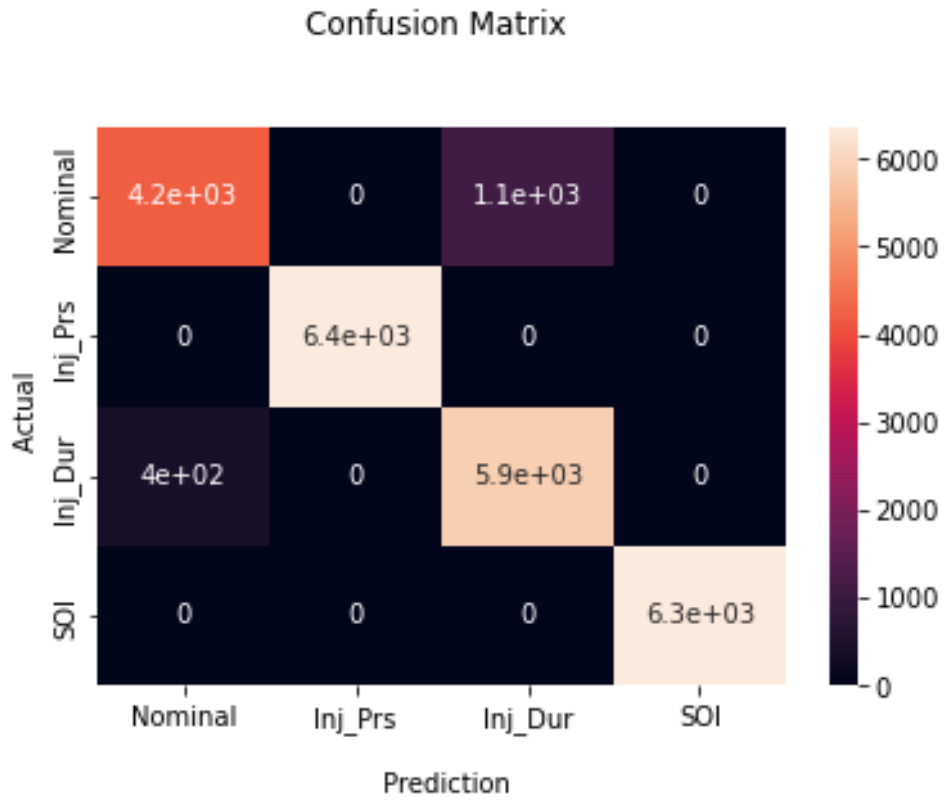

Figure 4.10: Confusion Matrix: Noise level equal to 50 percent of standard deviation

Table 4.3: Comprehensive Diagnosis Report for hybrid ensemble model: Noise level equal to 50 percent of standard deviation

	Precision	**Recall**	**F1-Score**	**Number of Samples**
Nominal	**0.91**	**0.80**	**0.85**	**5279**
Injection Pressure	**1.00**	**1.00**	**1.00**	**6361**
Injection Duration	**0.85**	**0.94**	**0.89**	**6290**
SOI	**1.00**	**1.00**	**1.00**	**6349**
Accuracy			**0.94**	**24279**
Macro Average	**0.94**	**0.93**	**0.93**	**24279**
Weighted Average	**0.94**	**0.94**	**0.94**	**24279**
Accuracy: 0.94				

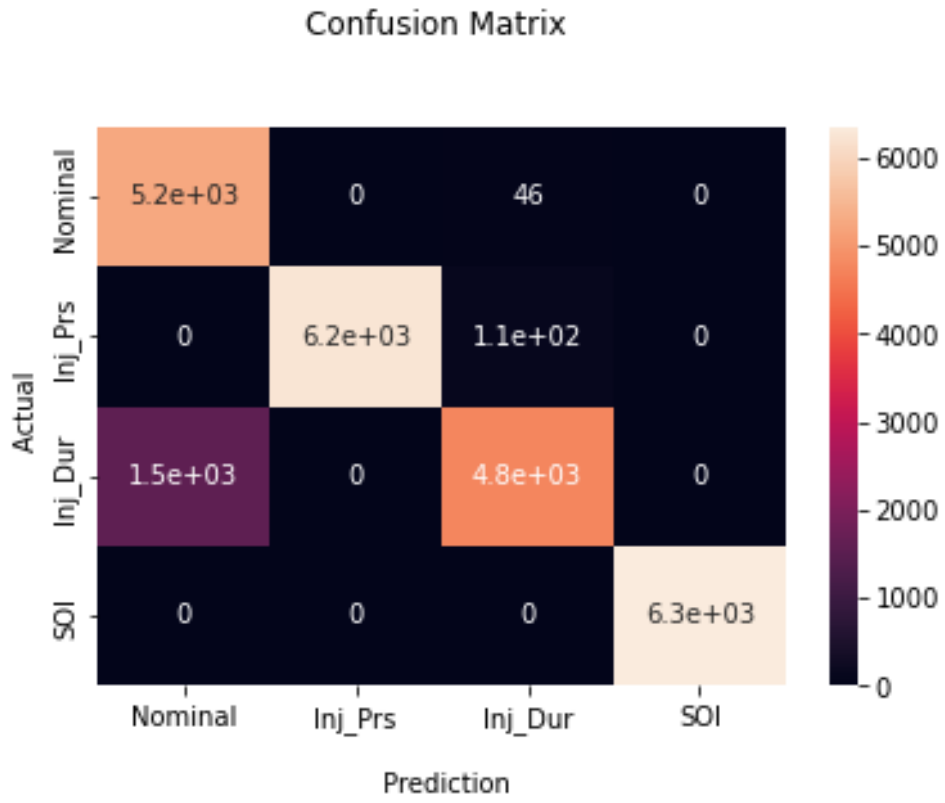

Figure 4.11: Confusion Matrix: Noise level equal to 100 percent of standard deviation

Table 4.4: Comprehensive Diagnosis Report for hybrid ensemble model: Noise level equal to 100 percent of standard deviation

	Precision	**Recall**	**F1-Score**	**Number of Samples**
Nominal	**0.77**	**0.99**	**0.87**	**5279**
Injection Pressure	**1.00**	**0.98**	**0.99**	**6361**
Injection Duration	**0.97**	**0.76**	**0.85**	**6290**
SOI	**1.00**	**1.00**	**1.00**	**6349**
Accuracy			**0.93**	**24279**
Macro Average	**0.94**	**0.93**	**0.93**	**24279**
Weighted Average	**0.94**	**0.93**	**0.93**	**24279**
Accuracy: 0.93				

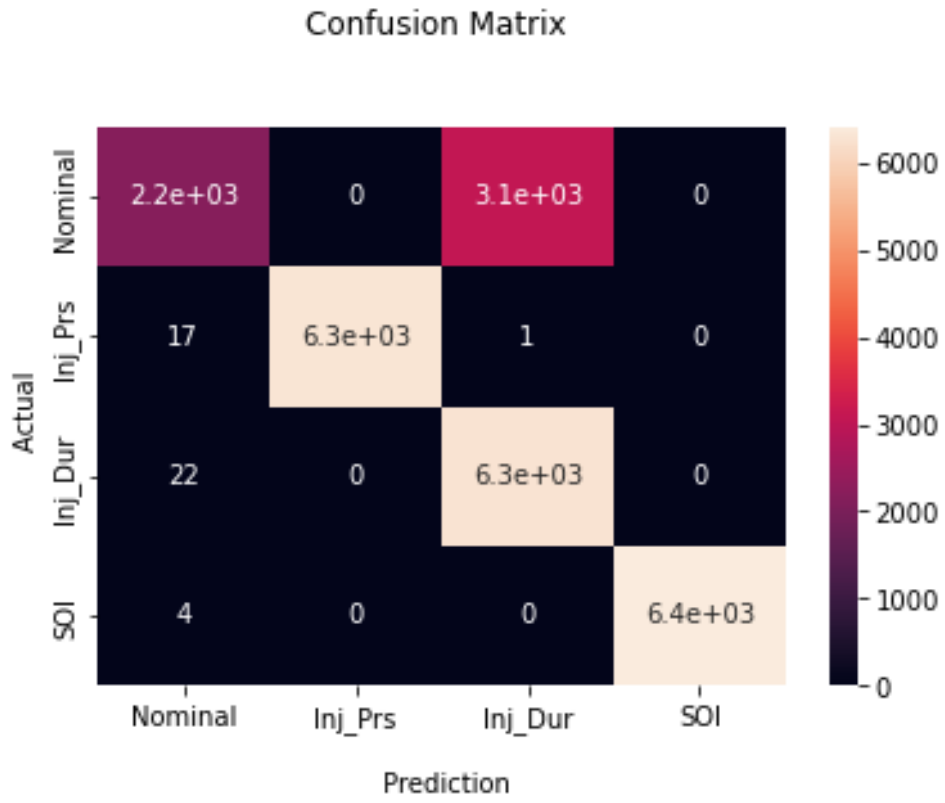

Figure 4.12: Confusion Matrix: Noise level equal to 200 percent of standard deviation

Table 4.5: Comprehensive Diagnosis Report for hybrid ensemble model: Noise level equal to 200 percent of standard deviation

	Precision	**Recall**	**F1-Score**	**Number of Samples**
Nominal	**0.98**	**0.41**	**0.58**	**5250**
Injection Pressure	**1.00**	**1.00**	**1.00**	**6361**
Injection Duration	**0.67**	**1.0**	**0.80**	**6297**
SOI	**1.00**	**1.00**	**1.00**	**6416**
Accuracy			**0.87**	**24279**
Macro Average	**0.91**	**0.85**	**0.85**	**24279**
Weighted Average	**0.91**	**0.87**	**0.86**	**24279**
Accuracy: 0.87				

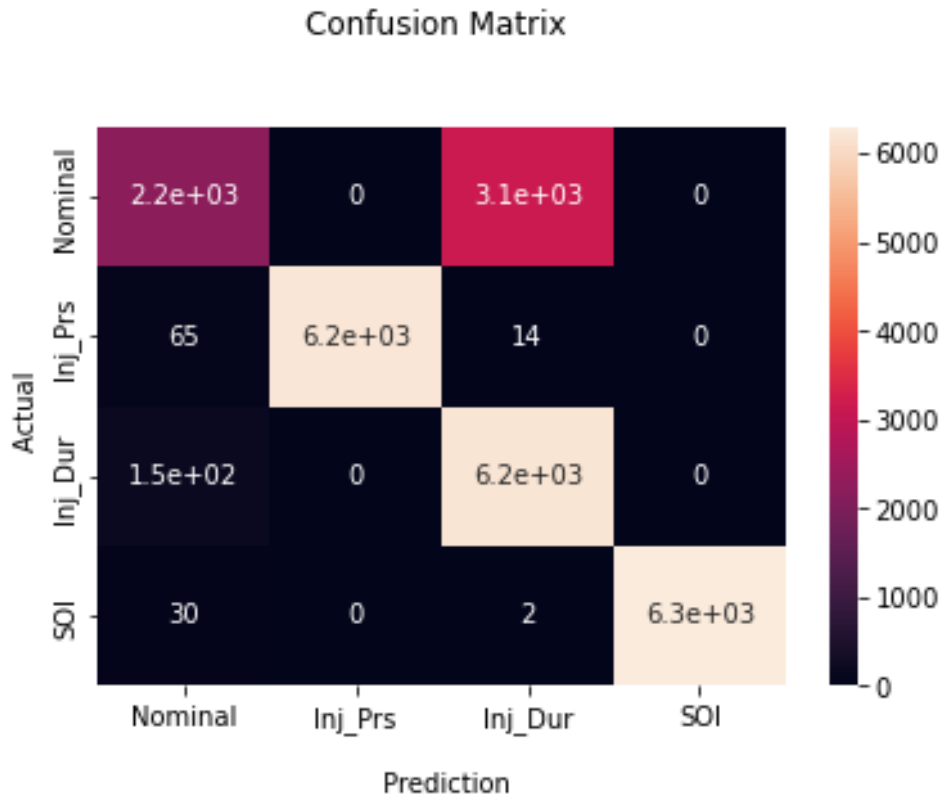

Figure 4.13: Confusion Matrix: Noise level equal to 300 percent of standard deviation

Table 4.6: Comprehensive Diagnosis Report for hybrid ensemble model: Noise level equal to 300 percent of standard deviation

	Precision	**Recall**	**F1-Score**	**Number of Samples**
Nominal	**0.90**	**0.41**	**0.56**	**5319**
Injection Pressure	**1.00**	**0.99**	**0.99**	**6281**
Injection Duration	**0.66**	**0.98**	**0.79**	**6350**
SOI	**1.00**	**0.99**	**1.00**	**6329**
Accuracy			**0.86**	**24279**
Macro Average	**0.89**	**0.84**	**0.84**	**24279**
Weighted Average	**0.89**	**0.86**	**0.85**	**24279**
Accuracy: 0.86				

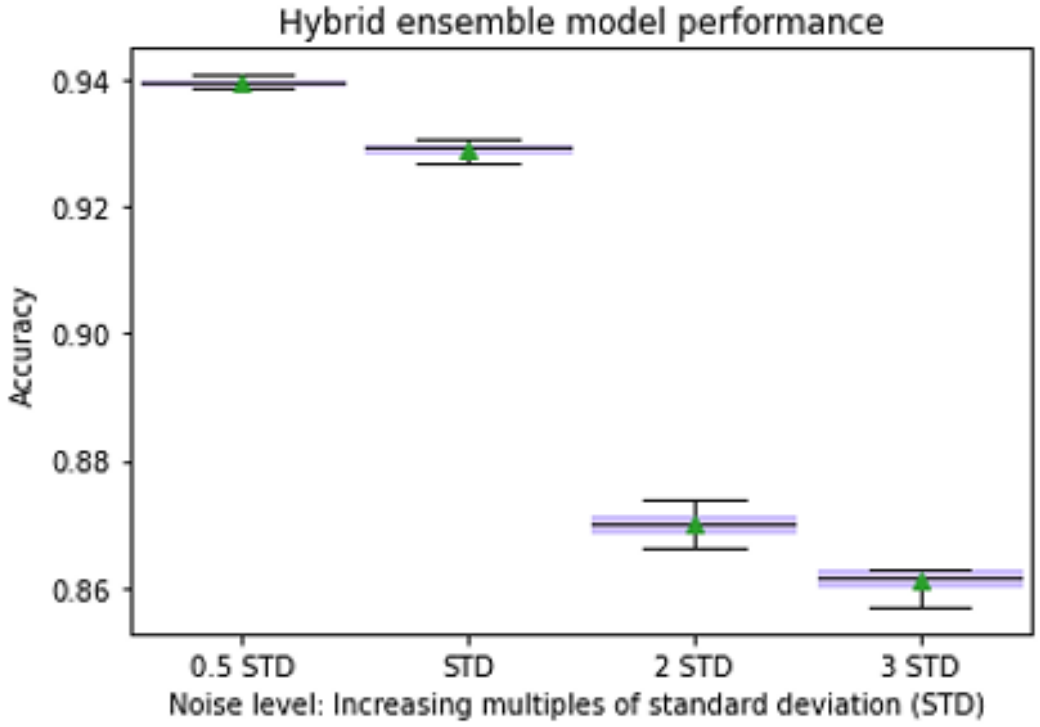

Figure 4.14: Comparative performance over different noise levels: Hybrid ensemble model

4.4 Results and Discussions

4.4.1 Fault Detection and Isolation

The physics-infused 1D-CNN-based ensemble learning model registers an accuracy of 99 percent on fault detection and isolation tasks. Table 4.2 provides a comprehensive diagnosis report of the hybrid ensemble model's performance for the engine data. From the table, it's pretty apparent that not just the accuracy but for the other metrics, such as precision, recall, and F1-Score, and their corresponding variations, namely micro, macro, and weighted, the hybrid ensemble model showcases promising performance.

In addition, the confusion plot, as described in Figure 4.9, provides a detailed understanding of the fault detection and isolation capabilities of the hybrid physics-infused 1D-CNN-based ensemble learning model. Analyzing the confusion plot as mentioned above, it becomes evident that the hybrid ensemble model can accurately classify the four fault classes, namely nominal (no fault) condition, fault due to change in injection pressure (Inj_Prs), fault due to shift in injection duration (Inj_Dur), and the fault due to the delay in the start of injection (SOI). Compared to the off-diagonal elements in the confusion matrix, the larger number of diagonal elements bear witness to efficient fault detection (distinguishing faulty samples from non-faulty ones) and isolation (determining the fault's location or cause). Besides, as visible from Figure 4.9, the number of false positives, i.e., predicting either of the three fault classes (fault due to Inj_Dur, fault due to Inj_Prs, fault due to SOI shift) when the actual sample is non-faulty (nominal), is significantly lower as compared to the number of accurately classified correct fault classes. The ability of the physics-infused 1D-CNN-based ensemble model to achieve significant levels of accuracy without compromising on the false positive rates can be credited to the physics-based engine model, which

makes the overall hybrid diagnostic ensemble framework more sentient by getting involved in the process of parameter construction and tuning of the level-0 and level-1 learners of the ensemble model.

4.4.2 Robustness to noisy data

Figures 4.10, 4.11, 4.12, and 4.13 present the confusion matrices demonstrating the performance of the physics-infused 1D-CNN-based ensemble model over noisy datasets. These evaluations were conducted over datasets comprising varying noise levels, ranging from point five times the standard deviation (or fifty percent of the standard deviation) to three times the original signal's standard deviation (or three hundred percent of the standard deviation). The above-mentioned confusion matrices clearly articulate the high robustness of the hybrid ensemble model when prone to noisy signals. The model's high robustness implies its capacity to perform effectively, even under challenging conditions with imprecise measurements from the onboard vehicle sensors, resulting in deviations from the expected values.

In Figure 4.10, we present the confusion plot depicting the performance of our pre-trained hybrid model under a noise level equivalent to fifty percent of the standard deviation of the original signal. The confusion plot reveals a relatively low number of misclassifications, with the hybrid ensemble model achieving an accuracy of 94 percent. In addition, the hybrid ensemble model demonstrated acceptable performance levels regarding recall, precision, and F1-scores, as evident from Table 4.3 showcasing the comprehensive diagnosis report for the performance of the hybrid ensemble model over a dataset with noise levels equal to fifty percent of the standard deviation of the original signals. The accuracy of the hybrid ensemble model degrades with the increase in the noise levels. The worst accuracy equal to 86 percent was ob-

served for the scenario where the noise level was equal to three hundred percent of the standard deviation of the original signal. However, the decline in accuracy levels was still within acceptable limits. Besides, from the Tables 4.3, 4.4, 4.5, and 4.6, we can discern that the hybrid model, even under the influence of noisy signals, demonstrates decent performance across various metrics such as recall, precision, and F1-scores and their variations such as the micro, macro, and the weighted scores.

The heightened robustness of the hybrid ensemble model can be credited to its superior feature-capturing capabilities facilitated by the incorporation of physics-based information. The inclusion of a physics-based engine model endows the hybrid ensemble learning diagnostics framework with a level of understanding, enabling it to handle noisy data effectively. Moreover, integrating domain knowledge enhances the model's generalizability, allowing it to perform well even on out-of-distribution noisy data samples. Furthermore, the utilization of physics aids in uncovering underlying dependencies within the data, granting the hybrid physics-infused ensemble model an advantage in accurately classifying fault classes, even in the presence of noise. This insightful combination of data-driven and physics-based approaches elevates the hybrid ensemble model's performance and adaptability to diverse practical scenarios.

Chapter 5

Hybrid Physics-Informed 1D CNN-based Prognostics Framework for Remaining Useful Life (RUL) Estimation in Diesel Engines

In this chapter, we aim to address the limitations of existing approaches. Our approach leverages a hybrid deep learning framework infused with physics principles. More specifically, we introduce a comprehensive prognostics model incorporating a physics-based engine model and deep learning components, notably a 1-dimensional Convolutional Neural Network (1D-CNN). This integrated system is designed to identify the pattern of increase in the injection duration (Inj_dur) of an automotive diesel engine. A faulty rise in the duration a fuel injector injects fuel in a diesel engine can lead to various performance issues and potentially damage the engine. This abnor-

mal behavior can be because of several factors, such as the wear-tear of the injectors, faulty or malfunctioning solenoid controlling the injector's opening and closing, and faulty injection timing. Increasing the time the fuel is injected into the combustion cylinder can increase the torque. If this rising fuel injection duration is unchecked and allowed to grow beyond a specific limit, it can lead to engine knocking or detonation, damaging internal engine components and reducing overall engine longevity. Besides, over time, prolonged fuel injection can lead to increased wear and tear of components like pistons, cylinder walls, and fuel injectors, resulting in higher repair and maintenance costs. Also, diesel engines rely on precise fuel injection timing to ensure that the fuel-air mixture is atomized correctly and ignited. If the injection duration is too long, enough air may not be available for complete combustion, leading to partially burned fuel, increased emissions, and reduced engine efficiency. Our physics-informed 1D CNN prognosis model tries to estimate the time remaining to reach the predetermined threshold limit of the injection duration. The time to get the threshold limit is termed the Remaining Useful Life (RUL) of the injection system in the context of this dissertation research work. The devised hybrid prognosis model's input is the combined feature vector, the sensor data coming out of the engine test bed, and the corresponding output of the physics-based engine model. The output of the physics-informed 1D CNN model, at any point in time, is the remaining useful life of the fuel injector from that point in time. The key contributions of the outlined chapter are as follows:

1. We propose an end-to-end hybrid physics-informed 1D CNN-based prognostics framework underpinned by physics principles and leveraging a one-dimensional Convolutional Neural Network (1D CNN) to estimate the remaining useful life of the fuel injector in the context of the increase in the amount of fuel injected

by it. The delineated hybrid physics-informed 1D CNN-based model operates autonomously, is computationally efficient, and can preprocess unprocessed test bed data and accurately perform fuel injector fault prognostics.

2. To assess and validate the practicality of the proposed hybrid prognosis model, we curated an extensive dataset [2] focusing on the pattern of fault increase over time, defined as the increase in the quantity of fuel injected in diesel engines over time, denominated as the "ProgEngine" repository, and is located at (*ProgEngine*). Fellow researchers working in fault prognosis can benefit from the curated data repository.

3. We undertook various tests to assess the hybrid prognostics model's efficacy across different operational scenarios. Furthermore, model showcased resilience and stability when subjected to noisy data samples.

In summary, the merger of physics-based models with the 1D CNN module for diesel engine prognostics helps ensure that predictions are not solely based on historical data, making the model robust to extrapolated and limited data and sensor-induced noises.

Chapter Organization: We start with a background overview of the damage propagation equation, followed by details of the prognostics dataset collection framework, encompassing the setup particulars. An account of the physics-based engine model follows this. Next, we delve into the details of the hybrid fault prognosis architecture proposed in this study. We then assess the effectiveness of the physics-informed 1D CNN model across various evaluation metrics, including the robustness against noisy data. In conclusion, we discuss the acquired empirical findings and explore prospective research directions.

5.1 Experimental Data Collection

5.1.1 Background Overview

One of the significant challenges ahead of us was to settle on the equation for damage propagation (increase in the injection duration) over time. Typical models employed in various application domains to model damage propagation encompass the Eyring model, the Coffin-Manson mechanical crack propagation model, and the Arrhenius model. The following lines of text will elaborate on a few.

Eyring Model:

The foundation of the Eyring Model lies within the realm of chemical reaction rate theory and is firmly grounded in the principles of chemistry and quantum mechanics [95]. It elucidates the relationship between the time to failure and the applied stress. Initially, the core model considers temperature, but its scope can be broadened to encompass other pertinent stress factors. The generalized equation of the Eyring model for stress and temperature can be expressed as follows:

$$t_f = AT^{\alpha} e^{(\frac{\Delta H}{kT} + (B + \frac{C}{T})S_1 + (D + \frac{E}{T})S_2)} \tag{5.1}$$

where,

t_f, refers to the time taken to reach the failure limit

α, A, B, C, D, E, and ΔH are the constants

T, refers to the temperature in Kelvin

S_1 and S_2 are the relevant stress values (e.g., some function of voltage

k, refers to Boltzmann's constant

Arrhenius Model:

The Arrhenius model finds extensive utility across diverse failure mechanisms. Historically, its application has been predominantly directed towards phenomena contingent upon diffusion mechanisms, chemical reactions, or migration processes [95]. This ambit effectively addresses a substantial portion of non-mechanical (or non-material fatigue) failure modes accountable for electronic equipment malfunctions. However, modified iterations of the Arrhenius equation have recently been increasingly harnessed for unconventional applications, including mechanical contexts and other atypical scenarios. The governing equation stands as follows:

$$t_f = Ae^{\frac{\Delta H}{kT}} \tag{5.2}$$

where,

t_f, refers to the time taken to reach the failure limit

T, refers to the temperature at the

ΔH, denotes the activation energy

A, refers to the scaling factor

k, refers to Boltzmann's constant

Proposed Fuel Injector Fault Propagation Model:

From the abovementioned text, we can observe that most fault propagation models have one thing in common, i.e., some exponential behavior in their growth trajectory. Apart from the above-described models, in-depth observation of similar models [34] laid the foundation on which we start by assuming a generalized degradation model for our fuel injector. The degradation model can be characterized by Equation 5.3.

$$d = ae^{bt} \tag{5.3}$$

Where, 'd', refers to the degradation as a function of time. 'a' and 'b' are the constants of the wear propagation model and can be determined by using the boundary conditions and are listed in Figure 5.2. We denote the initial injection duration value as Inj_dur_o. Therefore, the injection duration (Inj_dur) as a function of time 't' can be expressed by Equation 5.4. The injection duration with time follows an exponential trend.

$$Inj\_dur(t) = ae^{bt} + Inj\_dur_o \tag{5.4}$$

5.1.2 Experimental Setup

A comprehensive series of experiments was conducted on a Navistar production DT diesel engine (Figure 5.1(a)), distinguished by its 4-stroke 7.6 L inline 6-cylinder configuration. This engine was provided to Clemson University for testing by Pure Power Technologies, Inc. This standard production design engine incorporates electro-hydraulic fuel injectors and a dual-staged turbocharger, with exhaustive specifications outlined in Table 5.1. Multiple sensors were integrated into the engine setup to acquire data measurements such as air flow rate, fuel flow rate, fuel injection pressure, intake and exhaust pressure, and diverse temperature readings. Individual exhaust temperature measurements were diligently recorded at discrete exhaust ports. These temperature measurements are pivotal in identifying latent faults or irregularities in cylinder performance, especially when harmoniously integrated with data from other sensors. The overarching objective of these experiments encompassed a dual mission: first, to scrutinize and evaluate the overall engine performance, and second, to meticulously prognose potential issues, with a particular emphasis on injection duration monitoring as a precise means to pinpoint cylinder-specific problems.

The engine's speed was dynamically regulated via a dynamometer using the

Table 5.1: Engine geometry and specifications

Displacement	[L]	7.6
Bore	[mm]	116.6
Stroke	[mm]	118.9
Compression Ratio	[-]	16.9

(a) (b)

Figure 5.1: (a) Experimental Setup: Navistar engine (b) INCA-PUMA Indicom setup

test bed automation system by AVL called PUMA. To facilitate seamless communication between the Engine Control Unit (ECU) and the data acquisition and control systems within the test facility, the CAN Calibration Protocol (CCP) was deployed with the aid of ETAS INCA (Figure 5.1(b)). The parameters governing the injection process were meticulously managed by establishing an interface with an open ECU facilitated by the INCA software and CCP. This experimental setup provided an exceptionally granular command level and scrutiny of the engine's operational characteristics throughout the testing procedures.

The prognosis of any failure primarily involves the gradual degradation of components. Because of the gradual degradation, manually simulating the faults in the real test setup becomes difficult. Therefore, to collect data on the subtle deterioration of a fuel injector nozzle, we developed an automated framework based on the Python programming environment. The automated framework mainly consists of

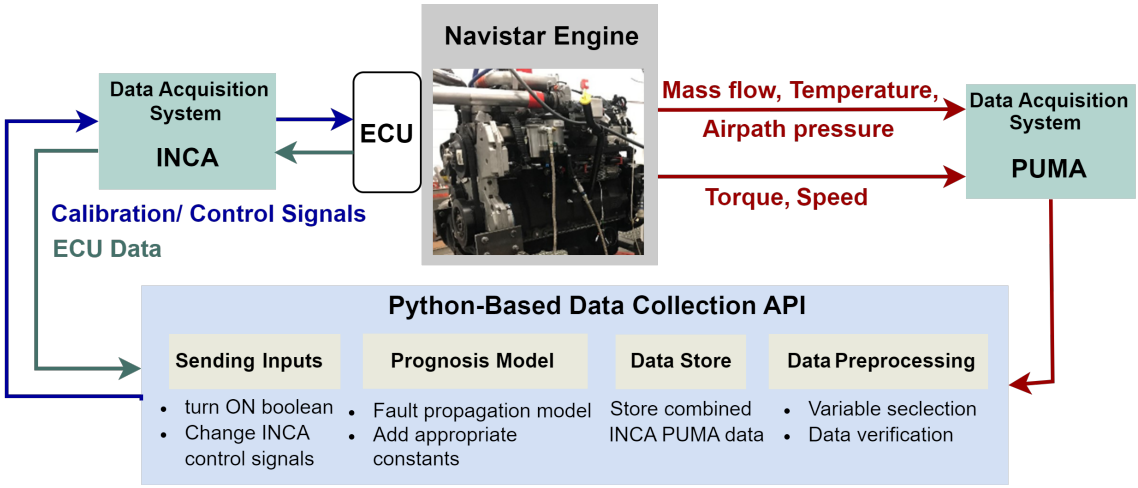

Figure 5.2: Data collection framework

taking the data from both of the data acquisition systems, that is, INCA and PUMA, generating appropriate inputs and fault severity, and finally sending the control inputs to the INCA. The schematic of the data collection framework is shown in Figure 5.2.

To establish the communication from the Python environment to both the data acquisition systems, appropriate Python libraries, drivers, and hardware were installed. We used the INCA-Python Application Programming Interface (API) and ES581.3 INCA adapter to interact with INCA. The INCA-Python API includes embedded Python dynamic libraries with the capability to load and display the workspace environment from the INCA GUI, modify boolean settings for control inputs, and update calibration values for actuators in real-time. On the other hand, CANlib and cantools API, and Kvaser leaf light V2 adapter were utilized to connect with PUMA. CAN frames from PUMA are periodically sent to the Python environment, with the timing of data transmission determined by the bus's transmission rate. To prevent data loss or congestion within the CAN bus, the run-time of the Python loop, which includes read/write commands, was adjusted and synchronized with the CAN bus transmission rate.

The Python-based data collection Application Programming Interface (API) primarily consists of four different components. First, the faulty injection duration input is calculated using the fuel injector fault propagation model as described in Equation 5.4. Second, the input parameters, such as start of injection (SOI) timing, injection pressure, and modified injection duration values, are added to the ECU. The booleans for those values are turned on to activate it. We utilize INCA-python API to deal with input parameters. The third component involves data recording and preprocessing. Here, the data for each time sample/instance is imported from INCA and PUMA. There is no need for data resampling because of importing data at the same time instance from two different data acquisition systems. The preprocessing of data majorly involves the selection of required variables and data verification.

- Variable selection: We thoroughly examined variables derived from the data acquisition setups of INCA and PUMA, systematically eliminating extraneous sensor data. Through a meticulous process involving the application of domain expertise and correlation analysis of features, we judiciously selected nine variables from a pool of twenty encompassed by INCA and PUMA. These chosen PUMA variables encapsulate critical information about the pressure, temperature, and volume flow rates of coolant, engine oil, fuel, intake air, and exhaust gases. Conversely, the INCA variables incorporate override values transmitted as directives to actuators, feedback signals received from actuators, and measurement values obtained post-adjustments. These variables directly quantify the intensity of the fault that is introduced, considering its respective physical measurement units.

- Data verification: Any missing or erroneous values from the readings are corrected using a neural network-based interpolation scheme that utilizes previous

time data to compute the current incorrect values.

The last portion of data collection API involves sending the combined INCA PUMA data to a hybrid 1D CNN-based prognosis model.

5.1.3 Design of Experiment

After establishing a secured data collection framework and a fuel injector fault propagation model, the next steps involved determining the specific speed and load points on an engine map that corresponded to real-world driving scenarios. The objective was to encompass the engine's A and B speeds (1400 rpm and 1600 rpm) while targeting up to 50% of the maximum brake torque the engine could generate at these speeds. This approach aimed to ensure a significant overlap of the engine operating points with the heavy-duty Supplemental Emission Test cycle (SET) and the heavy-duty FTP transient cycle, making the results more applicable to real-world driving conditions. These test cycles are part of the 1998 consent decrees between the United States Environmental Protection Agency (US EPA) and the US heavy-duty engine manufacturers and included as emission standards for heavy-duty engines.

To ensure the reliability of the test runs, certain boundaries were set for injection parameters, such as injection duration and the start of injection, in order to prevent potential misfires or damage to the engine or dynamometer. Additionally, as the fuel injection system utilized an electro-hydraulic unit injector (HEUI) rather than a high-pressure common rail system, injection pressure was constrained by the engine's speed. To prevent the commanded injection pressure from exceeding the capacity of the unit injectors the fault was introduced at a controlled rate at different engine speeds (1000, 1200, 1400, and 1600 rpm). At every engine speed two cases were recorded with different values for the a and b coefficients used in the degradation

Table 5.2: Design of experiment

Engine Speed (rpm)	Inj_press (Mpa)	Inj_dur_start (ms)	SOI (crank angle)	a	b	Torque_init (Nm)
1000	7	0.75	0	0.00027	1.23	100
1000	8	0.8	4.99219	0.001	1.05	175
1200	5	1.2	0	0.0009	1.0085	190
1200	7	0.75	1.99219	0.00139	1.0098	100
1400	7.5	0.75	0	0.00132	1.021	110
1400	7	0.75	-1.0078	0.00032	1.25	95
1600	9	0.8	0	0.00012	1.35	180
1600	9.5	0.70	0	0.00021	1.305	105

model. The injection duration in every case was incremented exponentially, starting from the values mentioned under the $'Inj\_dur\_start'$ column in Table 2. These values were relative to the base calibration in ms. The SOI timing and the injection pressures were maintained at their nominal values and held constant during the entire run for every case. This approach was intended to simulate a failure with the injector including injector deposits that reduce the fuel flow or erosion of the injector nozzle holes that increase the amount of fuel flow. This failure could also be associated with a particular injector's solenoid coil or command signal which may cause the injector to open for more or less time than commanded by the ECU.

5.2 Hybrid Fault Prognostics Framework

The devised fault prognostics framework integrates a comprehensive Simulink/MATLAB-based zero-dimensional multi-physics model of the engine with a fault prognosis module that combines 1D-convolutional, max-pooling, and dense layer architectures. Datasets from onboard sensors are collected from the engine testbed. Subsequently,

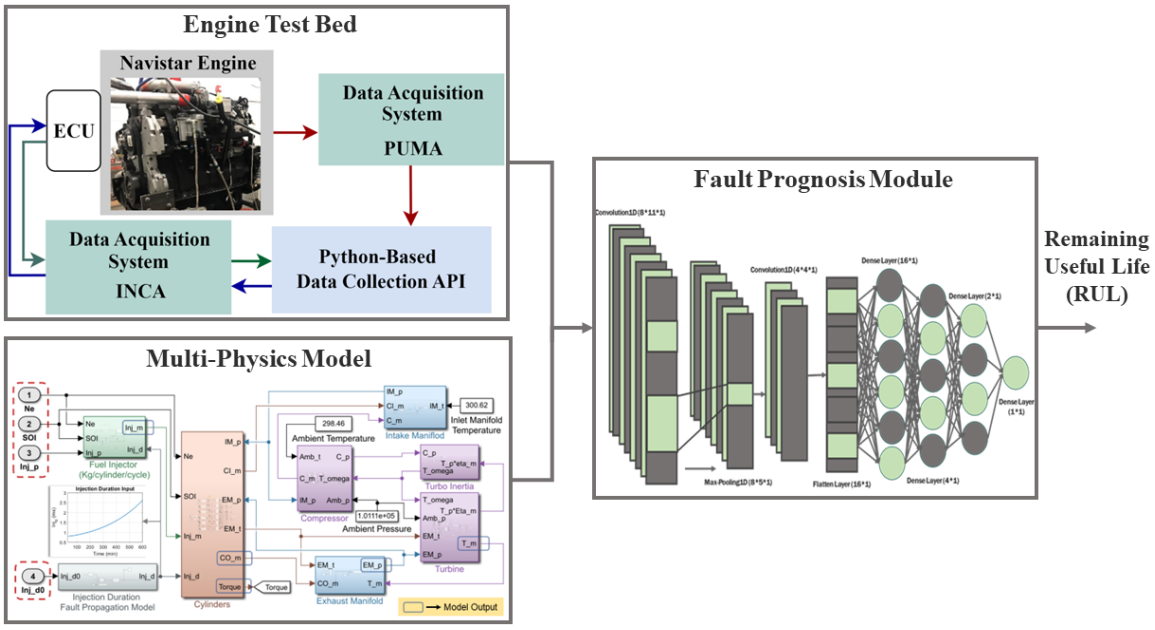

Figure 5.3: Hybrid Physics-Informed 1D CNN-based Fault Prognosis Pipeline

sensor data and the physics-based engine models' outputs are synthesized into a combined feature vector. This vector is processed through a sequence of 1D-Convolutional, max-pooling, and flattening layers within the fault prognosis framework. Figure 5.3 depicts the overall hybrid physics-informed fault prognostics computational pipeline. The various sub-components of the hybrid physics-informed computational prognostics framework are described in the subsequent subsections.

5.2.1 Multi-physics Model

The efficacy of an entirely data-driven model for engine fault prognosis relies predominantly on the dataset's size, specifically the data gathered under various operating conditions and fault severities. The acquisition of experimental data illustrating faulty behavior poses challenges due to potential permanent damage to engine components if faults are induced across all conceivable scenarios during engine operation. The data collected for this study was conducted in a controlled environment, metic-

ulously monitoring every engine output to ensure induced faults did not harm any engine component. This cautious approach, while safeguarding against damage, limited the dataset to lower fault severity levels. In real-life scenarios, faults rarely occur in a confined manner; they often exert significant impacts on the system, making it impractical to simulate in a controlled environment for data collection. Consequently, the role of a physics-based model becomes crucial, as it can function under desired conditions, unlike a test cell engine.

To address the challenges of generalizability and sparse datasets, the integration of a well-established physics-based model, bolstered by domain expertise, with a data-driven model is proposed. This combination can provide a comprehensive solution, overcoming limitations posed by the controlled environment and ensuring a more realistic representation of engine fault scenarios.

Numerous physics-based models for diesel engines exist in the literature, encompassing 0D, 1D, 2D, and 3D Computational Fluid Dynamics (CFD), among others [8, 74, 116]. 0D models, devoid of spatial dependency, solely rely on time. Comprising algebraic and ordinary differential equations, these models are exclusively time-dependent, revolving around a singular variable. In contrast, models with 1D, 2D, or 3D configurations introduce either one, two, or three spatial dimensions into the selected coordinate system. Constructing and fine-tuning such models demands considerable domain expertise. The computational time and accuracy of these models are typically directly proportional, presenting a pivotal factor in model selection. For the engine fault prognosis, we require a model that simulates faster on a standard Windows machine while providing an intuitive understanding of the diesel engine's actual behavior. Striking a balance between computational efficiency and accuracy led us to opt for the 0D model.

This paper introduces a high-fidelity, 0D physics-based engine model designed

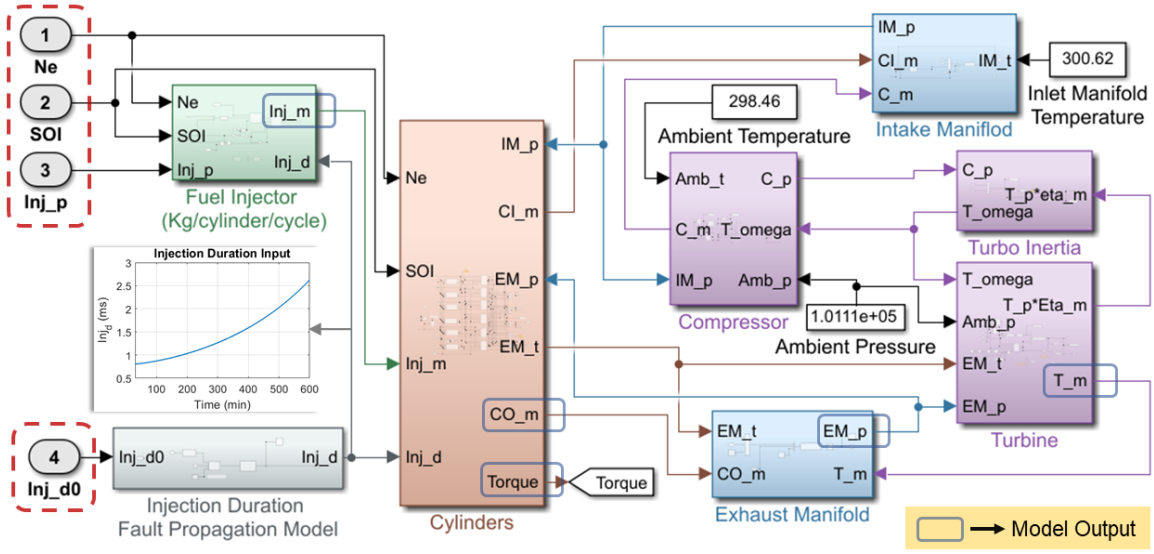

Figure 5.4: Overview of physics-based engine model. The connection and flow direction between inputs and outputs of all subsystems are mentioned. The output measurements and input parameters of the engine model are represented in blue line and red dashed line rectangular box, respectively.

for a 7.6-liter Navistar engine located at Clemson University's Advanced Combustion and Renewable Fuels Lab. The model is created using the Mean Value Engine Model (MVEM) technique, which simplifies the engine cycle by utilizing an effective value from the combustion cycle [106]. Implementation of this model takes place in the MATLAB and Simulink environments. Figure 5.4 presents the outline of the engine model, encompassing pressure, mass flow, temperature, and total torque across various engine subsystems. Fourteen state variables (illustrated in Figure 5.4) are utilized to capture these parameters.

The input/control parameters of the physics-based diesel engine model are mentioned below:

1. Engine speed (rpm) - N_e

2. Start of injection (Crank angle degree - CAD) - SOI

3. Injection duration (ms) - Inj_d

4. Injection pressure (MPa) - Inj_p

The modeling details are elaborated in the following subsections.

5.2.1.1 Injection duration fault propagation modeling

The injection duration fault propagation model is constructed based on Equation 5.4. This model utilizes the nominal input/control parameter injection duration (Inj_dur_o) and a degradation function to calculate faulty injection duration values. In the Simulink, we modeled the degradation function using the clock block to provide the current time value, along with the 'a' and 'b' constants from Table 5.2. The injection duration plot in Figure 5.4 illustrates sample fault propagation values. This plot is generated over a period of nearly 600 minutes at an engine speed of 1400 RPM, providing better clarity regarding the functionality of the fault propagation model. The computed faulty Inj_d is then inputted into the fuel injector model along with nominal Inj_p, SOI, and N_e.

5.2.1.2 Cylinder and fuel injector subsystems

The cylinder subsystem consists of the multi-cylinder model. The multi-cylinder model involves six separate cylinders and fuel injectors. This allows us to simulate and study the effect of a single faulty injector/cylinder behavior on the entire system. As the combustion process is an essential characteristic of engine operation, the correctness of the physics-based engine model strongly depends on the combustion model. In our multi-cylinder model, each cylinder model is a high-fidelity 0D thermodynamic model, described in [31]. This thermodynamic cylinder model was used to determine the mass flow rate, exhaust pressure, exhaust temperature, and

composition based on intake and fueling conditions. The fuel injector model emulates the behavior of a high pressure common rail fuel injector system from the experimental setup. We primarily utilize a principle of conservation of mass for the modeling of fuel injectors. This injector model outputs the injector fuel flow rate and needs inputs of Inj_d, Inj_p, SOI, and N_e.

5.2.1.3 Turbocharger and manifolds subsystems

The turbocharger consists of subsystems, including a turbine, compressor, and turbo inertial model. We derive the turbocharger model from the work of Wahlstrom and Eriksson [116]. Two states, namely intake and exhaust manifold pressure, are employed in modeling the intake and exhaust manifolds. The modeling of the pressure of manifolds relies on the fundamental principles of mass conservation and the ideal gas law. The differential equation for calculating manifold pressure is detailed below,

$$\frac{d}{dt}IM_p = \frac{R_a IM_t}{IM_v}(C_m - CI_m), \tag{5.5}$$

$$\frac{d}{dt}EM_p = \frac{R_e EM_t}{EM_v}(CO_m - T_m) \tag{5.6}$$

We mention all the parameters from Equation 5.5 and Equation 5.6 in the appendix.

5.2.1.4 Model tuning and validation

We collected a separate dataset for tuning and validation of the model. This dataset comprises steady-state data covering various operating conditions, ranging from an engine speed of 1000 to 1800 RPM and a torque of 100 to 600 Nm. The steady-state data were generated by varying inputs/control parameters for five dif-

ferent engine speeds. The physics-based model was tuned using data from engine speeds of 1000, 1200, 1600, and 1800 RPM and validated using data from 1400 RPM. We employed the least square optimization technique for model tuning [36]. This technique minimizes the error between modeled and experimental values, identifying the optimum value for the tuning parameters.

As a validation result, the model exhibited a Mean Percent Error (MPE) ranging from 4-7% over the 1400 RPM data, which is the unseen dataset for the model. We selected the output variables from the physics-based model for the fault prognosis module based on two factors. First, the chosen outputs should be correlated with the induced faults, and second, they should align with the outputs from the experimental data. The selected output variables are highlighted with a blue rectangular box in Figure 5.5. In total, five outputs from the physics-based model and nine from the engine testbed are sent to the fault prognosis module.

5.2.2 1 Dimensional Convolutional Neural Network (1D CNN)

Convolutional Neural Networks (CNNs) are a class of deep neural networks (DNNs) primarily designed for tasks involving images, leveraging their ability to learn and extract features automatically from spatial data. Convolutional Neural Networks are efficient in identifying the underlying patterns in the data owing to their stratified architectural design, comprising convolutional, pooling, and fully connected layers. 1D CNNs are a variant of the broader Convolutional Neural Networks (CNNs) class designed to process one-dimensional sequential data. They have found immense use in time-series analysis, signal processing, fault detection, and identifying patterns in sequences such as DNA and RNA. Besides, 1D CNN architecture has a faster real-time execution and lower computational hardware requirement when compared

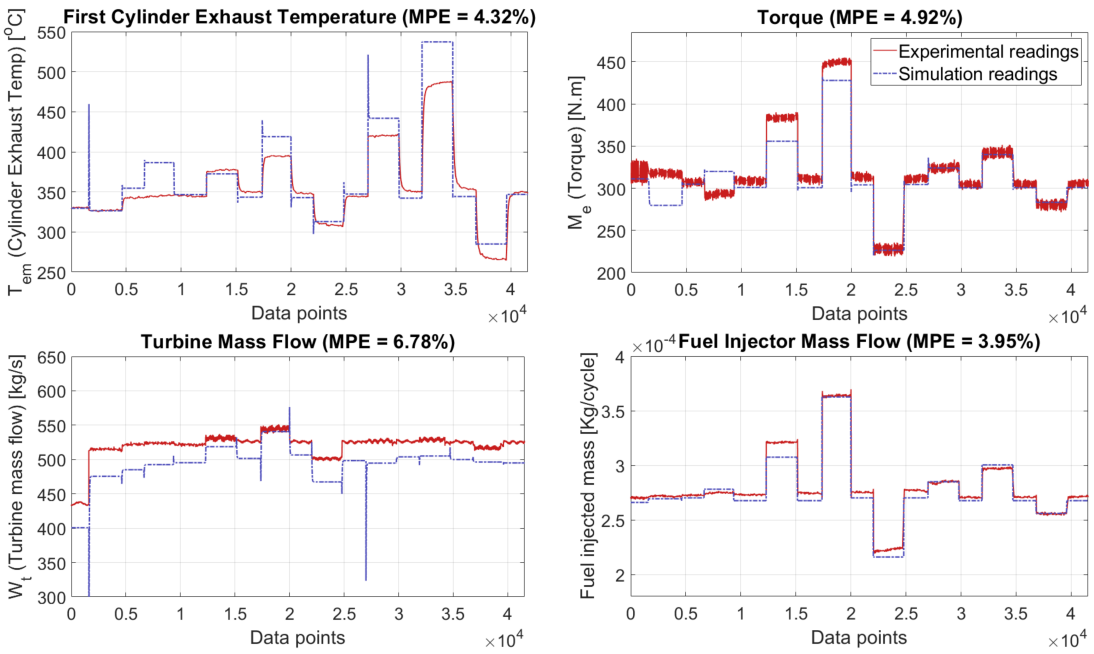

Figure 5.5: Physics-based model validation plots for the engine speed of 1400 rpm with different operating points. Dashed purple and red lines denote experimental and simulation readings, respectively. The Mean Percent Error (MPE) of each model's output is mentioned in their title.

to regular CNN. Our devised prognostics module consists of convolutional, pooling, and fully connected layers. In the proposed fault prognosis model, 1D convolutional layers consist of filters or kernels that slide across the input feature vector, that is, a combination of the output signals coming out of the engine test bed (9-element feature vector) and the output of the physics-based engine model (5-element feature vector), to perform element-wise multiplication and aggregation operations, enabling the network to decipher the underlying spatiotemporal patterns within the data. The outputs of the 1D convolutional layers are then transformed by a Rectified Linear Unit (ReLU) activation function, shown in Equations 5.7, to unveil the nonlinear latent relationship between the inputs and outputs. After the 1D convolutional layers, the model employs 1D-max pooling layers that downsample the feature maps produced by the preceding 1D convolutional layers.

$$ReLU(x) = max(x, 0) \tag{5.7}$$

The purpose of 1D-max pooling layers is to figure out the most significant feature from the input vector, filter out unnecessary information and ultimately enable a more concise feature vector representation for subsequent processing. Following the max pooling layer, we have the fully connected layers responsible for learning higher-level representations and transforming them into the final output that indicates the remaining useful life (RUL) from that specific instant in time. We trained the fault prognosis model using the Adam optimization technique, which involved adjusting the weights and biases to minimize a Mean Squared Error (MSE) loss function. Figure 5.6 showcases the optimal physics-informed 1D-CNN model for fault prognostics. In essence, using the 1D CNN module in the overall physics-informed fault prognosis framework helps extract critical information from input sensor data and make

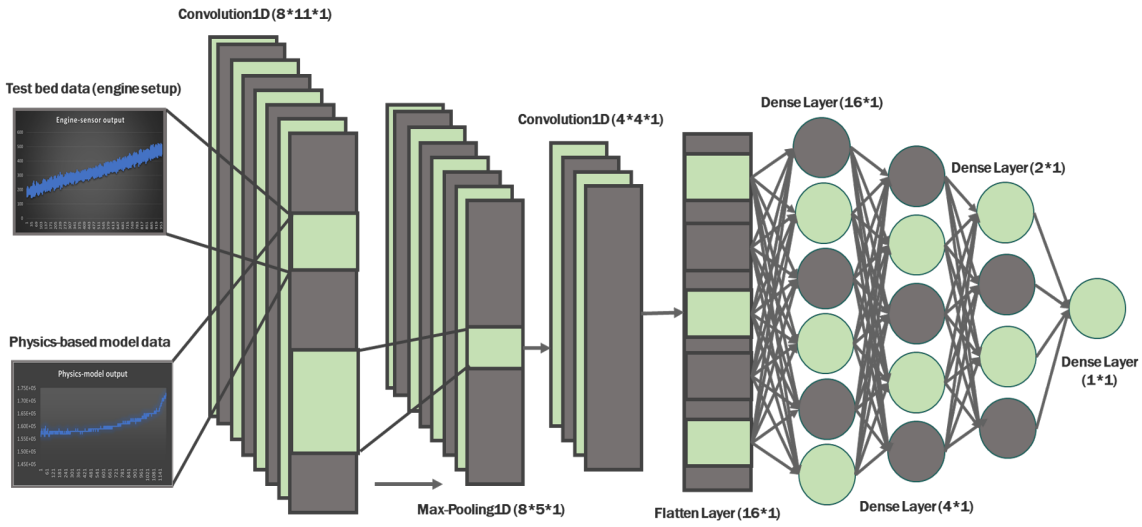

Figure 5.6: Hybrid 1D CNN-based Fault Prognosis Model

Remaining Useful Life (RUL) predictions based on inherent patterns within the data.

5.2.3 Computational Resources

The fault prognosis model is trained on Lenovo ThinkPad X1, using Tensorflow-Keras (Tensorflow GPU 2.9.1), with an Intel(R) Core(TM) i7-10750H CPU clocked at 2.60GHz. This device has six cores, 12 logical processors, 16 GB of RAM, and an NVIDIA GeForce GTX 1650Ti with Max-Q Design GPU.

5.3 Hybrid Model Evaluation

5.3.1 Evaluation Metrics

In our research, the fault prognosis model assessment revolves around the Mean Absolute Error (MSE) and Mean Squared Error (MSE), which serve as the evaluation metric and the primary loss function. The MAE quantifies the disparity between predicted and actual values. It measures the average of the absolute differences between

the predicted and actual values. The formula for MAE is given by Equation 5.8. The assessment of the model's performance involves utilizing MAE values to compare the predicted Remaining Useful Life (RUL) against the actual RUL values.

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \tag{5.8}$$

The training process of the physics-informed 1D CNN model involves calculating the gradient of the loss function (in this case, MSE) concerning the model's parameters and using optimization algorithms (Adam algorithm in our case) to update these parameters to minimize the MSE. Equation 5.9 showcases the expression for MSE.

$$MSE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|^2}{n} \tag{5.9}$$

5.3.2 Robustness to Noisy Data

To replicate real-world conditions for a diesel engine in production, we deliberately introduced diverse noise levels to the data collected from test-bed sensors. Specifically, we incorporated multiple Additive White Gaussian Noise (AWGN) levels into the data samples. These noisy datasets were then run through a pre-trained physics-informed 1D CNN prognostics model. We introduced three distinct noise levels equal to the standard deviation (100 percent of the standard deviation), twice the standard deviation (200 percent of the standard deviation), and thrice the standard deviation (300 percent of the standard deviation) of the original signal. This exercise evaluated the model's resilience when confronted with noisy data, imitating the typical conditions experienced during practical operations.

5.3.3 Hyperparameter Tuning

Hyperparameter optimization (or tuning) involves meticulously searching for an ideal set of hyperparameters tailored to a specific learning algorithm. The primary objective is to figure out the configuration that results in the most efficient model, minimizing a predefined cost function for a given dataset. We experimented with diverse hyperparameters, such as the number of layers, number of neurons per layer, batch size, number of epochs, learning rate, number of convolutional filters, kernel size for convolutional layers, and batch size for purely data-driven and hybrid fault prognostic models. Furthermore, we implemented regularization techniques such as L1 regularization during the model training to address potential overfitting concerns. The hybrid fault prognostics model underwent hyperparameter optimization using a five-fold cross-validation approach through the grid-search technique. Figure 5.7 presents box plots illustrating MAE values over the validation set for a subset (six) of the total hyperparameter combinations explored, and the corresponding network configurations are demonstrated in Table 5.3. The model with the optimal hyperparameter combination (configuration shown in the second column of Table 5.3) was preserved and subsequently evaluated on noisy datasets. The training utilized the Adam optimizer with a learning rate set at 0.001. The 1D-convolutional and dense layers employed the *ReLU* and *Tanh* activation functions, respectively, while the outermost dense layer utilized linear activation. The network topology of the best hybrid fault prognostics model is detailed in Figure 5.6. A parallel hyperparameter optimization process was conducted for the pure data-driven 1D-CNN-based prognostics model, excluding the physics-based engine model input, serving as a benchmark for the hybrid model.

Table 5.3: Hyperparameter combinations of hybrid physics-informed 1D CNN model (f:filter number,k:kernel size,n:number of nodes)

Configuration	1	2	3	4	5	6
1D Convolutional Layer	f(16)/k(1*2)	f(8)/k(1*4)	f(8)/k(1*2)	f(16)/k(1*2)	f(4)/k(1*4)	f(4)/k(1*2)
1D Max pooling layer	k(1*2)	k(1*2)	k(1*2)	k(1*2)	k(1*2)	k(1*2)
1D Convolutional layer	f(4)/k(1*2)	f(4)/k(1*2)				
Flatten layer						
Dense layer	n(12)	n(16)		n(8)		
Dense layer	n(4)	n(4)	n(4)	n(4)	n(4)	n(4)
Dense layer	n(2)	n(2)	n(2)	n(2)	n(2)	n(2)
Dense layer	n(1)	n(1)	n(1)	n(1)	n(1)	n(1)
Learning Rate	0.01	0.001	0.01	0.01	0.001	0.001
Batch Size	100	100	50	50	100	200
Epochs	1000	1000	500	2000	1000	1000

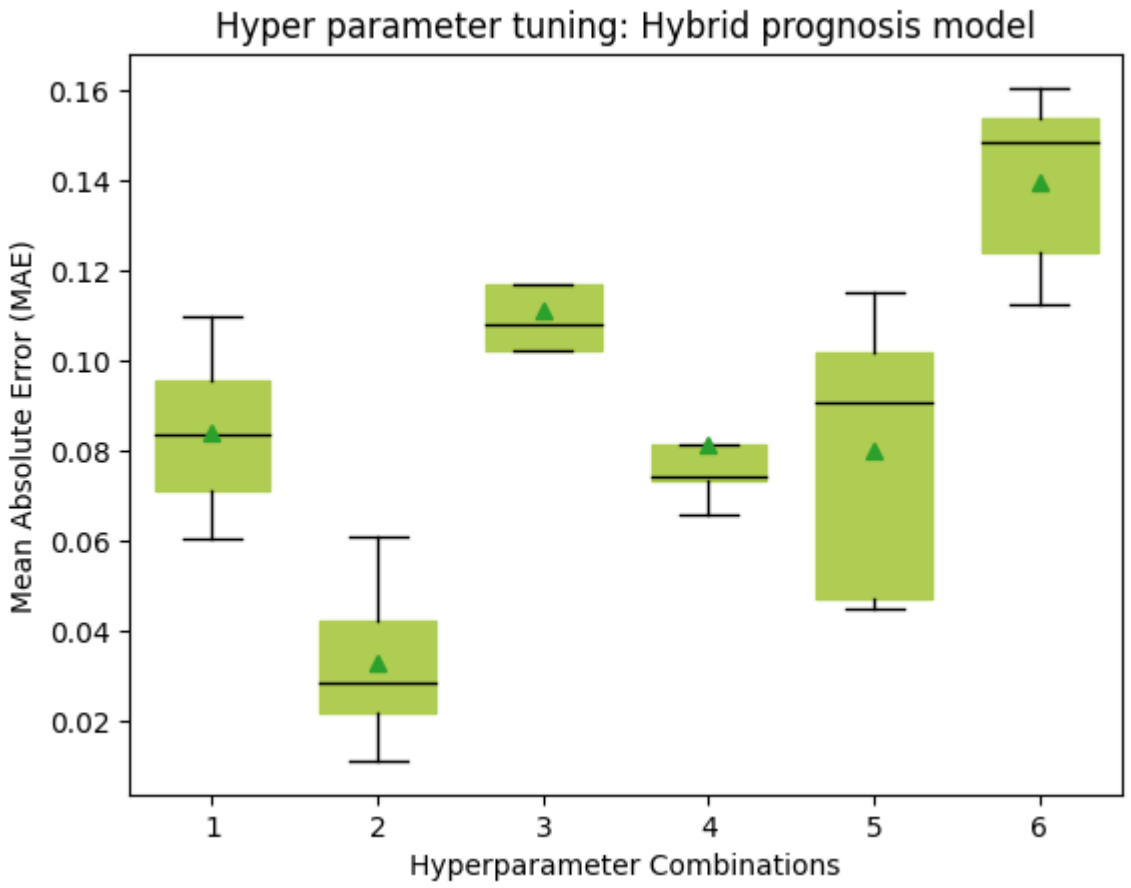

Figure 5.7: Hybrid prognosis model performance corresponding to various hyperparameter combinations

Table 5.4: Computational performance comparison: Data-driven and Hybrid models

Data-driven model		Hybrid model	
Criteria	**Values**	**Criteria**	**Values**
# Model Parameters	**351**	**# Model Parameters**	**461**
Computational Complexity	**2011**	**Computational Complexity**	**2076**
FLOPS	**59482.7 (0.000594G)**	**FLOPS**	**870137.5 (0.000870G)**
Mean Absolute Error (MAE)	**0.1019**	**Mean Absolute Error (MAE)**	**0.022**

5.4 Computational Efficiency

This evaluation criteria offers insights into the computational intricacies of a given model. The computation time is determined for a specific machine on which the model is run by assessing the necessary number of FLOPS (Floating Point Operations Per Second), as shown by Equation 5.10, to produce the model's output. Complexity is calculated in terms of 'Big O' notations and helps determine the number of operations the model performs independently of the platform it runs. The comparison of both Computational complexity and FLOPS with the Mean Absolute Error (MAE) aids in striking a trade-off between the computational cost and performance associated with the trained model. Table 5.4 compares the computational performance of the pure data-driven and the hybrid physics-informed prognostics model. The table shows that with a minor increase in FLOPS and computational complexity, the hybrid model registers less MAE, almost one-fifth, compared to the data-driven counterpart. Hence, the performance-to-cost ratio is higher for the hybrid model when compared to the data-driven model.

$$FLOPS = cores * \frac{cycles}{second} * \frac{FLOPs}{cycle} \tag{5.10}$$

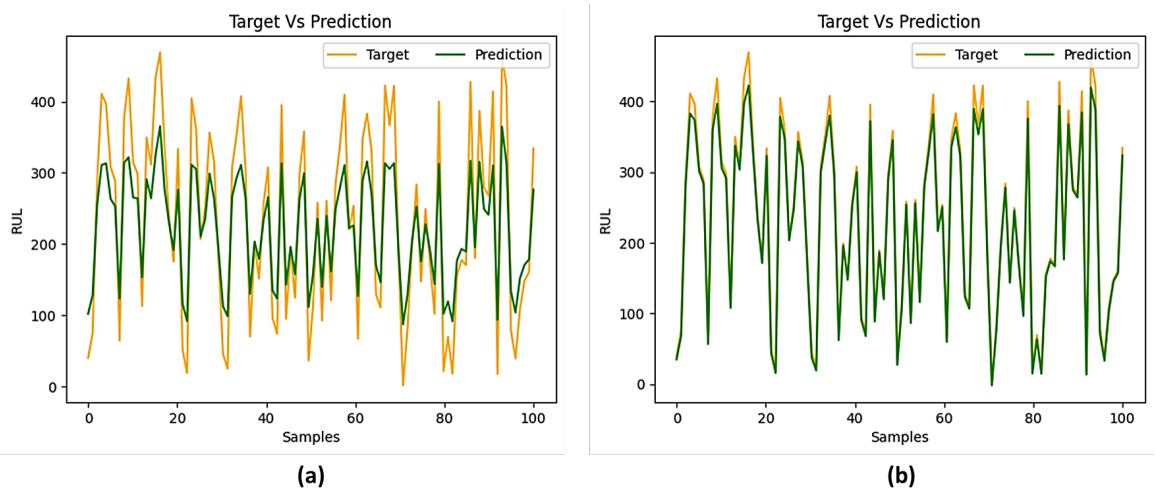

Figure 5.8: RUL Prediction:(a)Pure Data-driven model (b) Hybrid physics-informed 1D CNN-based Prognosis Model

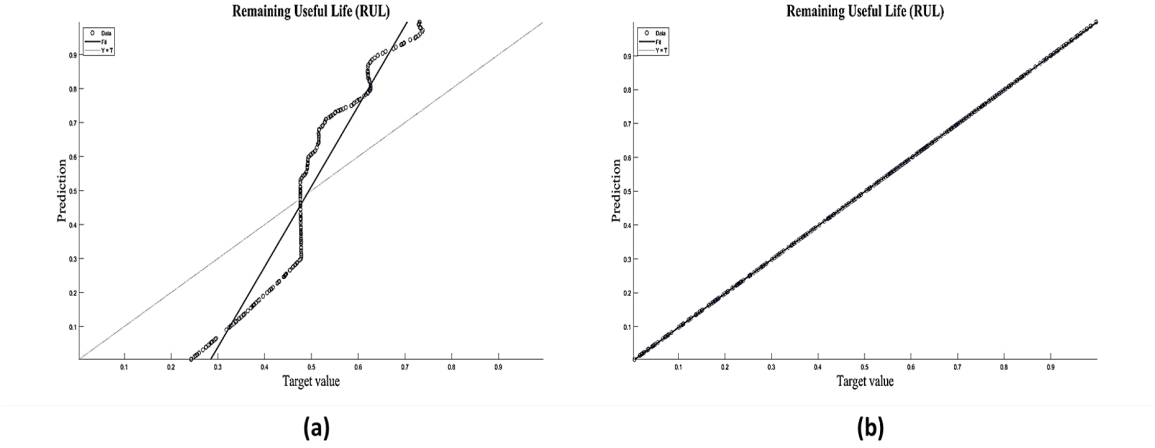

Figure 5.9: Regression plot:(a) Pure Data-driven model (b) Hybrid physics-informed 1D CNN-based Prognosis Model

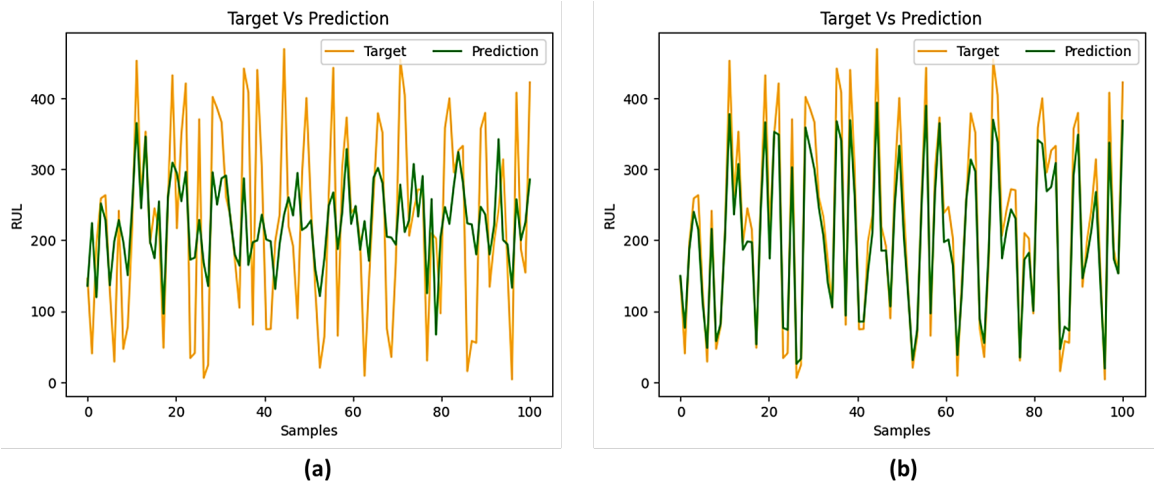

Figure 5.10: Robustness to noise level equal to standard deviation: (a) Pure data-driven model (b) Hybrid 1D CNN model

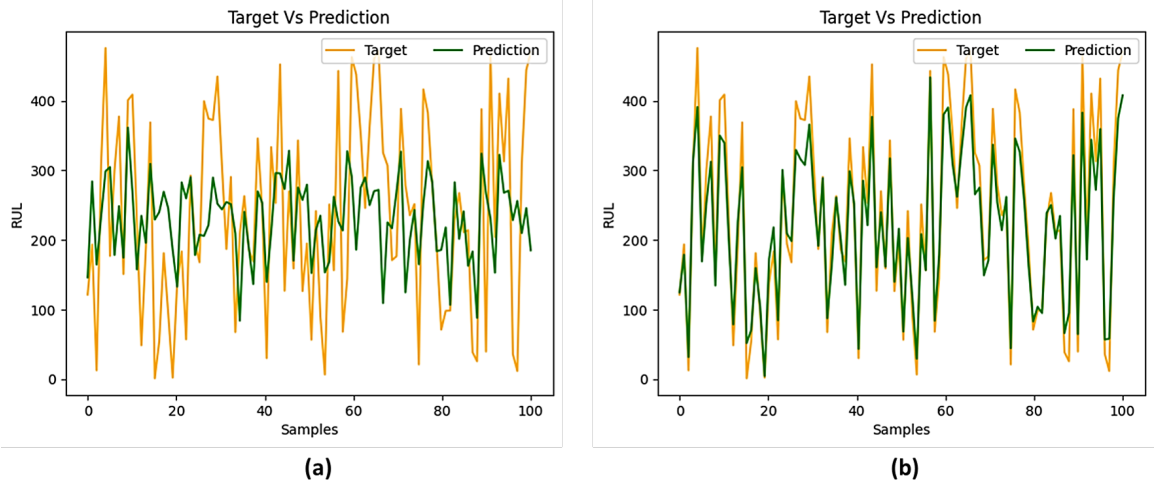

Figure 5.11: Robustness to noise level twice the standard deviation: (a) Pure data-driven model (b) Hybrid 1D CNN model

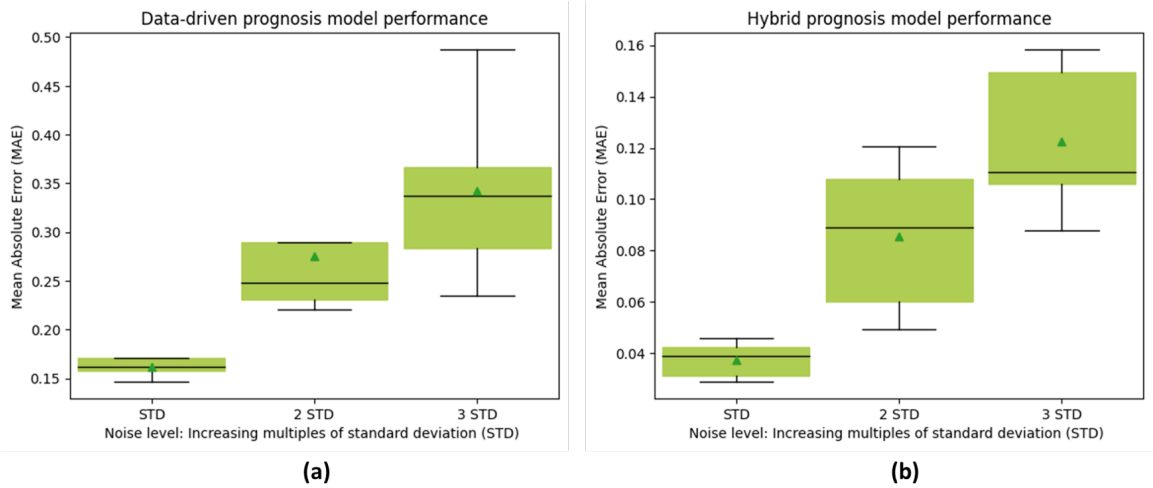

Figure 5.12: Performance with increasing noise levels: (a) Pure data-driven model (b) Hybrid 1D CNN model

5.5 Results and Discussions

5.5.1 Fault prognosis for extrapolated data

Extrapolation estimates the model's predictive capabilities on completely unseen data. It is a crucial metric for evaluating the performance of any model when it comes to determining its utility in practical applications where real-time data are scarce, and it is tough to collect data related to all scenarios. Thus, the chosen model must be able to imitate the system's behavior in the presence of an insufficient or out-of-domain dataset. The models were trained on datasets pertaining to three engine speeds, namely, 1000 rpm, 1200 rpm, and 1400 rpm. The trained models were tested on a completely unseen dataset corresponding to an engine speed of 1600 rpm. On comparing the prediction plots in Figure 5.8, we can observe that the hybrid physics-informed 1D CNN-based prognostics model outperforms the pure data-driven. The hybrid model can better trace the ground truth values more efficiently when compared to the pure data-driven model on a wholly extrapolated test set.

On the test set, the trained hybrid physics-informed prognostics model registered an MAE of 0.022, compared to 0.1019 recorded by the pure data-driven model. Besides from the regression plots, as shown in Figure 5.9, it's evident that the hybrid prognostics model outperforms the data-driven one. The superior performance and efficient computational performance of the trained hybrid model on an extrapolated unseen dataset, when compared to the pure data-driven model, can be attributed to the physics information going into the hybrid model, which guides the parameter tuning of the model, making the model more generalizable and enhancing the model's predicting capabilities for an unseen out of domain dataset.

5.5.2 Robustness to noisy data

During engine operations, measurements may be noisy or incomplete, and this is where the physics-informed 1D CNN model can be helpful when obtaining a clean and comprehensive data sample is challenging. Figure 5.12 showcases box plots demonstrating the performance of the pure data-driven and the physics-informed 1D CNN prognostics models over noisy datasets. The box plots effectively communicate the robust stability of the hybrid model in the presence of noisy signals. The hybrid model's high robustness to noisy data indicates its ability to operate efficiently, even when imprecise measurements from the vehicle's onboard sensors may lead to deviations from the actual values. Figure 5.10 compares the RUL predictions made by the data-driven and the hybrid model when exposed to noise levels equal to 100 percent of the standard deviation of the original signals. When subjected to similar noise levels, the hybrid model records an average MAE of 0.09 compared to 0.27 of the data-driven model. From the plot, it is pretty evident that the hybrid model can more precisely trace down the ground truth values. Besides, Figure 5.11 compares the

performance of both the models when subjected to noise levels equal to 200 percent of the standard deviation of the original signal. In this case, too, we can observe that the hybrid model, with an average MAE of 0.12, is much more resilient to noises in the data when compared to a data-driven model with an average MAE of 0.34. Another key observation from the box plots in Figure 5.12 is that the hybrid model is much more stable to increment in the noise levels from 100 percent of the standard deviation to 300 percent of the standard deviation of the signal. The increment in the error levels, with increasing noise, for the hybrid model is much lower than for the data-driven model.

The hybrid framework's enhanced robustness can be attributed to integrating the physics model's output, which improved its feature-capturing abilities. The merger of the physics-based engine model renders the hybrid framework much more cognizant and makes it robust to noisy data samples. Including the physics-based model helped the hybrid model gain insights into the underlying principles of the engine system, and this enhanced interpretability of the hybrid model makes it less noise-prone and improves its generalizability.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The dissertation outlines two hybrid frameworks for fault diagnosis in an automotive diesel engine. We used the dataset from an actual inline 6-cylinder 4-stroke 7.6 L Navistar production DT diesel engine. Besides, a low-fidelity Simulink-based engine simulation model was used as the physics component of the hybrid fault diagnosis framework. The Auto Encoder model forms the data-driven module of the comprehensive hybrid fault detection and isolation architectures and serves as the dimensionality reduction mechanism for the data coming from the test bed (experimental engine setup).

We used a pure data-driven 1D-CNN model, without the physics component, to benchmark the performance of the Hybrid Physics-Infused 1D-CNN based Deep Learning Framework. For fault detection and isolation, the hybrid model, with an accuracy of 98 percent, outperforms the data-driven model, which has an accuracy of 92 percent. When generating fewer false signals or false positives, i.e., predicting either of the three faults even when the actual sample is nominal (no-fault), the

hybrid model outshines the data-driven model. As evident from Figures 3.13, 4.9, the hybrid registers a lower false positive ratio of 0.029 (150/5100) compared to 0.262 (1100/4200) of the data-driven model. The capability of the hybrid model to achieve higher overall accuracy without compromising much on the false positive detections makes it compatible with the actual practical applications, where we do not intend to limit the vehicle performance and day-to-day operations without a valid reason just because of false warnings. Also, from the Tables 3.9, 3.10, it is observable that individual class-wise, macro, and weighted precision, recall, and F1-scores are better for the hybrid model as compared to the data-driven model. Besides, while evaluating the robustness to noise, it is quite clear that the hybrid model is a cut above the pure data-driven model. From the Figures, 3.15, 3.16, 4.11, 4.12, the false signals, i.e., detecting one of the three fault cases even when the actual sample is nominal (no-fault), are pretty low for the hybrid physics-infused 1D-CNN model when set side by side with the pure data-driven model for the same level of noise. The ability of the hybrid model to have better accuracy, in contrast to the data-driven model, while maintaining a lower number of the generated false signals can be quite pivotal in practical applications where noisy data is a norm.

Besides, the second fault diagnostics approach involves the integration of physics-based models with 1D-CNN blocks within an ensemble learning framework, thereby enhancing the system's diagnostic capabilities. The proposed methodology leverages a real-world dataset from an inline 6-cylinder 4-stroke 7.6 L Navistar production DT diesel engine. Moreover, a low-fidelity Simulink-based engine simulation model is incorporated as the physics component within the hybrid fault diagnosis framework. An Auto Encoder model is another crucial component of the comprehensive hybrid fault detection and isolation architecture. This powerful model reduces the dimensionality of the data acquired from the test bed, which is set up with an ex-

perimental engine configuration. By combining advanced computational techniques with real-world engine data, the proposed method demonstrates high accuracy and effectiveness in identifying and localizing faults in the diesel engine. The hybrid ensemble model, which combines physics and data-driven 1D-CNN components for fault detection and isolation in automotive diesel engines, exhibited a remarkable accuracy of 99 percent on the test bed data. Of particular significance is the hybrid ensemble model's ability to generate fewer false signals or false positives, i.e., incorrectly predicting faults in nominal (no-fault) samples. Figure 4.9 demonstrates that the hybrid model achieved a significantly lower false positive ratio of 0.032 (98+73 = 171 samples out of 5193 total nominal samples).

The hybrid ensemble model's capacity to achieve heightened overall accuracy while concurrently mitigating the occurrence of false positives is important in the context of practical applications. This attribute holds particular relevance in contexts where unwarranted false warnings could significantly hamper vehicular performance and disrupt routine operations. Furthermore, detailed examination, as presented in Table 4.2, indicates that the hybrid ensemble model also exhibits high individual class-wise, macro, micro, and weighted precision, recall, and F1-scores. Additionally, we evaluated the robustness of the hybrid ensemble model to noise, and As evident from Figures 4.10, 4.11, 4.12, and 4.13, the hybrid physics-infused 1D-CNN model exhibited considerably fewer false signals even under the presence of noisy conditions. This characteristic is especially valuable in real-world applications where noisy data is expected.

Also, this dissertation outlines a hybrid physics-informed 1D CNN-based approach for fault prognostics in automotive diesel engines. The method integrates sensor data, physics-based models, and 1D-CNN models within a comprehensive framework, significantly enhancing the framework's prognostic capabilities. The hy-

brid model uses onboard sensor data from a 7.6L, 6-cylinder, 4-stroke diesel engine. Besides, the hybrid framework uses the outputs of a 0D low-fidelity physics-based engine model. By combining the real-time data and the physics-based information, the model achieved significant accuracy when predicting the fuel injection system's remaining useful life (RUL). The results show that the hybrid model outperforms the pure data-driven model. On predicting the RUL on an extrapolated out-of-domain dataset, the hybrid model registered a mean accuracy of 97.8 percent compared to 89.81 percent of the pure data-driven model. Besides, the hybrid model showcases a better performance-to-cost ratio than the data-driven model. For the criterion of robustness to noisy data, the hybrid model features a mean accuracy of 96 percent, 91.5 percent, and 89.4 percent when compared to 84.5 percent, 76 percent, and 67 percent as exhibited by the pure data-driven model when estimating the RUL for noise levels equal to 100 percent of standard deviation, 200 percent of standard deviation, and 300 percent of standard deviation of the original signals, respectively. The superior functionality of the hybrid model, when compared to its pure data-driven counterpart across both the extrapolated and noisy datasets, confirms its practical applicability. In conclusion, the amalgamation of physics-based knowledge with a data-driven 1D CNN model amplifies the efficacy and dependability of the overall hybrid prognostics framework, rendering it a valuable instrument for real-world engine applications.

6.2 Summary of Contributions

The dissertation makes an attempt to address the research challenges as outlined in Chapter 1 (Section 1.3). The specific contributions of this dissertation are enumerated as follows:

1. **Hybrid Physics-Infused 1D-CNN based Deep Learning Framework for Diesel Engine Fault Diagnostics (Chapter 3)**

   (a) A novel end-to-end physics-infused 1D-CNN-based deep learning framework for diesel engine fault detection is conceptualized. The outlined hybrid architecture is completely autonomous in processing the raw test bed data and diagnosing the corresponding fault scenarios.

   (b) The Hybrid fault diagnostics model can accentuate the features that contain the most vital information pertinent to the diesel engine fault. Especially when it comes down to fault classification, the hybrid model, leveraging the underlying physical laws of the system, can help improve the accuracy compared to a pure data-driven or physics-derived fault detection model.

   (c) To train and test the feasibility of the proposed hybrid architecture, we curated a rare-of-its-kind large-scale diesel engine fault database, termed *NavicEngine* [1], under different operating conditions. The database is located at (*NavicEngine*).

   (d) When exposed to noisy sensor data, the hybrid model outperforms the pure data-driven model. This superior performance of the hybrid model can be attributed to the hybrid model's enhanced generalizability owing to the inclusion of physics-based information.

2. **Hybrid Physics-Infused 1 Dimensional-Convolutional Neural Network (1D-CNN) based Ensemble Learning Framework for Diesel Engine Fault Diagnostics (Chapter 4)**

   (a) An innovative hybrid framework that integrates physics principles into a

1D-CNN-based ensemble deep learning model to detect faults in diesel engines. This hybrid architecture can process raw data from the test bed and autonomously diagnose fault scenarios without external intervention.

(b) Utilizing ensemble learning, which amalgamates the strength of several 1D-CNN sub-models, offers a strategy to alleviate the shortcomings of individual models, thus enhancing both accuracy and robustness against noisy data. Furthermore, by integrating the principles of physics with the 1D-CNN framework, we strengthen our ability to discern underlying data patterns, culminating in a performance level that adequately addresses the intricate dynamics of a diesel engine system.

(c) We tested the robustness of the hybrid ensemble model against a dataset with varying noise levels. Notably, the model demonstrated remarkable levels of accuracy as measured by a spectrum of evaluation metrics, rendering it suitable for practical diesel engine applications where noisy data is a norm.

3. **Hybrid Physics-Informed 1D CNN-based Prognostics Framework for Remaining Useful Life (RUL) Estimation in Diesel Engines (Chapter 5)**

(a) State-of-the-art hybrid physics-informed 1D CNN-based prognostics framework underpinned by physics principles and leveraging a one-dimensional Convolutional Neural Network (1D CNN) to estimate the remaining useful life of the fuel injector in the context of the increase in the amount of fuel injected by it. The delineated hybrid physics-informed 1D CNN-based model operates autonomously, is computationally efficient, and can preprocess unprocessed test bed data and accurately perform fuel injector

fault prognostics.

(b) Empirical results witness the hybrid prognostics model's capability to capture the diesel engine's nonlinear behavior, which helps in accurate RUL estimation even in the presence of noisy sensor data.

(c) To assess and validate the practicality of the proposed hybrid prognosis model, we curated an extensive dataset [2] focusing on the pattern of fault increase over time, defined as the increase in the quantity of fuel injected in diesel engines over time, denominated as the "ProgEngine" repository, and is located at (*ProgEngine*). Fellow researchers working in fault prognosis can benefit from the curated data repository.

6.3 Future Work

Despite the advantages of hybrid models for fault diagnostics over conventional physics-based or data-driven models, there is much room for future explorations in this domain. Currently, both the outlined diagnostics frameworks focus on determining the presence of a fault and its cause. For the experimental data, the proposed model showcased promising results. One avenue for future research is to perform severity quantification, i.e., to find the numerical strength of the fault along with detection and isolation. Another avenue for future work is to record the acoustic signals generated during the experiment using microphones. Audio signals, encapsulating the fault information, can be used as an additional modality with the hybrid model to improve the generalizability and enhance model robustness to noisy and out-of-distribution data.

Similarly, there are numerous avenues for further exploration of hybrid prognostics. One of the domains to explore would be employing better and more ad-

vanced data-driven models, such as Generative Adversarial Networks (GANs) and Transformer networks, to complement the physics-based module of the hybrid framework. Such state-of-the-art data-driven models are better at capturing the nonlinear behavior of diesel engines, thus resulting in better RUL estimations. Besides, we can use multi-modal data comprising time-varying vibration signals, infrared camera images, and the traditional onboard sensor. Multiple data modalities may further hone the predictive capabilities of the hybrid prognostics framework, which is crucial for predictive maintenance and safe operations of a diesel engine.

Appendices

Appendix A Nomenclature for Physics-based Engine Model for Fault Diagnostics

Notation

P_{im}	Intake manifold pressure (Pa)
P_{em}	Exhaust manifold pressure (Pa)
W_c	Compressor mass flow (Kg/sec)
W_t	Turbine mass flow from (Kg/sec)
W_{ei}	Cylinder in mass flow (Kg/sec)
W_{eo}	Cylinder out mass flow (Kg/sec)
T_{im}	Intake manifold temperature (K)
T_{em}	Cylinder exhaust temperature (K)
M_e	Torque (Nm)
R_a	Intake ideal-gas constant (J/KgK)
R_e	Exhaust ideal-gas constant (J/KgK)
V_{im}	Intake manifold volume (m^3)
V_{em}	Exhaust manifold volume (m^3)
U_{delta}	Fuel mass flow $(Kg/cycle)$
ω_t	Rotational turbine speed (r/min)

Appendix B Nomenclature for Physics-based Engine Model for Fault Prognostics

Notation

C_m	Compressor mass flow (Kg/sec)
EM_t	Cylinder exhaust temperature (K)
CI_m	Cylinder in mass flow (Kg/sec)
CO_m	Cylinder out mass flow (Kg/sec)
R_e	Exhaust ideal-gas constant (J/KgK)
EM_p	Exhaust manifold pressure (Pa)
EM_v	Exhaust manifold volume (m^3)
Inj_m	Fuel mass flow $(Kg/cycle)$
R_a	Intake ideal-gas constant (J/KgK)
IM_p	Intake manifold pressure (Pa)
IM_t	Intake manifold temperature (K)
IM_v	Intake manifold volume (m^3)
T_ω	Rotational turbine speed (r/min)
T_m	Turbine mass flow from (Kg/sec)

Bibliography

[1] shubhes/navicengine. `https://github.com/shubhes/NavicEngine`. (Accessed on 08/18/2023).

[2] shubhes/progengine. `https://github.com/shubhes/ProgEngine`. (Accessed on 01/03/2024).

[3] Appanpillai Agasthian, Rajendra Pamula, and Lakshmi Annamalai Kumaraswamidhas. Fault classification and detection in wind turbine using cuckoo-optimized support vector machine. *Neural Computing and Applications*, 31:1503–1511, 2019.

[4] Mohammed S Alhajeri, Fahim Abdullah, Zhe Wu, and Panagiotis D Christofides. Physics-informed machine learning modeling for predictive control using noisy data. *Chemical Engineering Research and Design*, 186:34–49, 2022.

[5] Anas H Aljemely, Jianping Xuan, Osama Al-Azzawi, and Farqad KJ Jawad. Intelligent fault diagnosis of rolling bearings based on lstm with large margin nearest neighbor algorithm. *Neural Computing and Applications*, 34(22):19401–19421, 2022.

[6] Diah Harnoni Apriyanti, Luuk J Spreeuwers, and Peter JF Lucas. Deep neural networks for explainable feature extraction in orchid identification. *Applied Intelligence*, pages 1–16, 2023.

[7] Yunhao Ba, Guangyuan Zhao, and Achuta Kadambi. Blending diverse physical priors with neural networks. *arXiv preprint arXiv:1910.00201*, 2019.

[8] Seungju Baek, Seungchul Woo, Youngkun Kim, and Kihyung Lee. Prediction of turbocharged diesel engine performance equipped with an electric supercharger using 1d simulation. *Energy*, 185:213–228, 2019.

[9] Garima Bhatia and Natarajan Gunasekaran. Heat-up of diesel particulate filters: 2d continuum modeling and experimental results. *SAE transactions*, pages 574–586, 2003.

[10] Hamed Bolandi, Gautam Sreekumar, Xuyang Li, Nizar Lajnef, and Vishnu Naresh Boddeti. Physics informed neural network for dynamic stress prediction. *Applied Intelligence*, pages 1–16, 2023.

[11] Piero P Bonissone and Harold E Johnson. Expert system for diesel electric locomotive repair. *Human Systems Management*, 4(4):255–262, 1984.

[12] Weronika Borek-Marciniec and Pawel Ksieniewicz. Neural network architecture with intermediate distribution-driven layer for classification of multidimensional data with low class separability. *Applied Intelligence*, 53(21):26050–26066, 2023.

[13] Nestor Diego Rivera Campoverde and Juan Fernando Chica Segovia. Expert system design for fault diagnosis in diesel engines. *International Journal of Systems Applications, Engineering & Development*, 10(2016), 2016.

[14] Jian Chen and Robert Bond Randall. Improved automated diagnosis of misfire in internal combustion engines based on simulation models. *Mechanical Systems and Signal Processing*, 64:58–83, 2015.

[15] Jiangce Chen, Justin Pierce, Glen Williams, Timothy W Simpson, Nicholas Meisel, Sneha Prabha Narra, and Christopher McComb. Accelerating thermal simulations in additive manufacturing by training physics-informed neural networks with randomly-synthesized data. *Journal of Computing and Information Science in Engineering*, pages 1–14, 2024.

[16] Rui Chen. *Model-based fault detection and diagnosis of selective catalytic reduction systems for diesel engines*. PhD thesis, University of Illinois at Urbana-Champaign, 2014.

[17] Yuwen Chen, Bin Song, Yuan Zeng, Xiaojiang Du, and Mohsen Guizani. A deep learning-based approach for fault diagnosis of current-carrying ring in catenary system. *Neural Computing and Applications*, pages 1–13, 2021.

[18] Sheng-Min Chiu, Yow-Shin Liou, Yi-Chung Chen, Chiang Lee, Rong-Kang Shang, and Tzu-Yin Chang. Identifying key grid cells for crowd flow predictions based on cnn-based models with the grad-cam kit. *Applied Intelligence*, 53(11):13323–13351, 2023.

[19] Raluca F Constantinescu, Peter D Lawrence, Phillip G Hill, and Terrence S Brown. Model-based fault diagnosis of a two-stroke diesel engine. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, volume 3, pages 2216–2220. IEEE, 1995.

[20] Luca Cornolti, Angelo Onorati, Tarcisio Cerri, Gianluca Montenegro, and Federico Piscaglia. 1d simulation of a turbocharged diesel engine with comparison of short and long egr route solutions. *Applied energy*, 111:1–15, 2013.

[21] Francisco A Cubillos, Eduardo Vyhmeister, Gonzalo Acuña, and Pedro I Alvarez. Rotary dryer control using a grey-box neural model scheme. *Drying Technology*, 29(15):1820–1827, 2011.

[22] Arka Daw, Anuj Karpatne, William D Watkins, Jordan S Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge-Guided Machine Learning*, pages 353–372. Chapman and Hall/CRC, 2017.

[23] Arinan Dourado and Felipe AC Viana. Physics-informed neural networks for missing physics estimation in cumulative damage models: a case study in corrosion fatigue. *Journal of Computing and Information Science in Engineering*, 20(6):061007, 2020.

[24] Jian Du, Jianqin Zheng, Yongtu Liang, Qi Liao, Bohong Wang, Xu Sun, Haoran Zhang, Maher Azaza, and Jinyue Yan. A theory-guided deep-learning method for predicting power generation of multi-region photovoltaic plants. *Engineering Applications of Artificial Intelligence*, 118:105647, 2023.

[25] Vikas Dwivedi and Balaji Srinivasan. A normal equation-based extreme learning machine for solving linear partial differential equations. *Journal of Computing and Information Science in Engineering*, 22(1):014502, 2022.

[26] Enas Elgeldawi, Awny Sayed, Ahmed R Galal, and Alaa M Zaki. Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. In *Informatics*, volume 8, page 79. MDPI, 2021.

[27] Lars Eriksson. Modeling and control of turbocharged si and di engines. *Oil & Gas Science and Technology-Revue de l'IFP*, 62(4):523–538, 2007.

[28] Justin Flett and Gary M Bone. Fault detection and diagnosis of diesel engine valve trains. *Mechanical Systems and Signal Processing*, 72:316–327, 2016.

[29] Erik Frisk and Lars Nielsen. Robust residual generation for diagnosis including a reference model for residual behavior. *Automatica*, 42(3):437–445, 2006.

[30] Song Fu, Shisheng Zhong, Lin Lin, and Minghang Zhao. A re-optimized deep auto-encoder for gas turbine unsupervised anomaly detection. *Engineering Applications of Artificial Intelligence*, 101:104199, 2021.

[31] Brian Gainey and Benjamin Lawler. A fuel cell free piston gas turbine hybrid architecture for high-efficiency, load-flexible power generation. *Applied Energy*, 283:116242, 2021.

[32] Weize Gao, Shanxiong Chen, Chongsheng Zhang, Bofeng Mo, and Xuxing Liu. Obm-cnn: a new double-stream convolutional neural network for shield pattern segmentation in ancient oracle bones. *Applied Intelligence*, 52(11):12241–12257, 2022.

[33] Rana A Genedy, Matthias Chung, and Jactone A Ogejo. Physics-informed neural networks for predicting liquid dairy manure temperature during storage. *Neural Computing and Applications*, pages 1–16, 2023.

[34] Kai Goebel, Hai Qiu, Neil Eklund, and Weizhong Yan. Modeling propagation of gas path damage. In *2007 IEEE aerospace conference*, pages 1–8. IEEE, 2007.

[35] Zhiqiang Gong, Weien Zhou, Jun Zhang, Wei Peng, and Wen Yao. Joint deep reversible regression model and physics-informed unsupervised learning for temperature field reconstruction. *Engineering Applications of Artificial Intelligence*, 118:105686, 2023.

[36] Giorgio Grisetti, Tiziano Guadagnino, Irvin Aloise, Mirco Colosi, Bartolomeo Della Corte, and Dominik Schlegel. Least squares optimization: From theory to practice. *Robotics*, 9(3):51, 2020.

[37] Hongwei Guo, Xiaoying Zhuang, Xiaolong Fu, Yunzheng Zhu, and Timon Rabczuk. Physics-informed deep learning for three-dimensional transient heat transfer analysis of functionally graded materials. *Computational Mechanics*, pages 1–12, 2023.

[38] Qun Guo, Xinhao Zhang, Jing Li, and Gang Li. Fault diagnosis of modular multilevel converter based on adaptive chirp mode decomposition and temporal convolutional network. *Engineering Applications of Artificial Intelligence*, 107:104544, 2022.

[39] Ekaterina Gurina, Nikita Klyuchnikov, Ksenia Antipova, and Dmitry Koroteev. Forecasting the abnormal events at well drilling with machine learning. *Applied Intelligence*, pages 1–16, 2022.

[40] Lino Guzzella and Christopher H. Onder. *Mean-Value Models*, pages 21–146. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[41] Imen Hamrouni, Hajer Lahdhiri, Khaoula Ben Abdellafou, Ahamed Aljuhani, and Okba Taouali. Anomaly detection for process monitoring based on machine learning technique. *Neural Computing and Applications*, 35(5):4073–4097, 2023.

[42] John B Heywood. *Internal combustion engine fundamentals*. McGraw-Hill Education, 2018.

[43] Sangjin Hong, Margaret S Wooldridge, Hong G Im, Dennis N Assanis, and Heinz Pitsch. Development and application of a comprehensive soot model for 3d cfd reacting flow studies in a diesel engine. *Combustion and Flame*, 143(1-2):11–26, 2005.

[44] Seong Hyeon Hong, Junlin Ou, and Yi Wang. Physics-guided neural network and gpu-accelerated nonlinear model predictive control for quadcopter. *Neural Computing and Applications*, 35(1):393–413, 2023.

[45] Akin Ilhan. Forecasting of river water flow rate with machine learning. *Neural Computing and Applications*, 34(22):20341–20363, 2022.

[46] Rolf Isermann. Model base fault detection and diagnosis methods. In *Proceedings of 1995 American Control Conference-ACC'95*, volume 3, pages 1605–1609. IEEE, 1995.

[47] Rolf Isermann. Model-based fault-detection and diagnosis–status and applications. *Annual Reviews in control*, 29(1):71–85, 2005.

[48] Jake Janssen, Ghadir Haikal, Erin DeCarlo, Michael Hartnett, and Matthew Kirby. A physics-informed general convolutional network for the computational modeling of materials with damage. *Journal of Computing and Information Science in Engineering*, pages 1–67, 2023.

[49] Joseph Jenkins, Adeline Paiement, Yann Ourmières, Julien Le Sommer, Jacques Verron, Clément Ubelmann, and Hervé Glotin. A dnn framework for learning lagrangian drift with uncertainty. *Applied Intelligence*, pages 1–11, 2023.

[50] Junzhong Ji and Yao Yao. A novel cnn framework to extract multi-level modular features for the classification of brain networks. *Applied Intelligence*, pages 1–18, 2022.

[51] Xiaowei Jia, Anuj Karpatne, Jared Willard, Michael Steinbach, Jordan Read, Paul C Hanson, Hilary A Dugan, and Vipin Kumar. Physics guided recurrent neural networks for modeling dynamical systems: Application to monitoring water temperature and quality in lakes. *arXiv preprint arXiv:1810.02880*, 2018.

[52] Jiajie Jiang, Hui Li, Zhiwei Mao, Fengchun Liu, Jinjie Zhang, Zhinong Jiang, and He Li. A digital twin auxiliary approach based on adaptive sparse attention network for diesel engine fault diagnosis. *Scientific Reports*, 12(1):1–18, 2022.

[53] Tinu Theckel Joy, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Hyperparameter tuning for big data using bayesian optimisation. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2574–2579. IEEE, 2016.

[54] Daniel Jung, Lars Eriksson, Erik Frisk, and Mattias Krysander. Development of misfire detection algorithm using quantitative fdi performance analysis. *Control Engineering Practice*, 34:49–60, 2015.

[55] U Kiencke. Engine misfire detection. *Control engineering practice*, 7(2):203–208, 1999.

[56] Byungsoo Kim, Vinicius C Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. In *Computer graphics forum*, volume 38, pages 59–70. Wiley Online Library, 2019.

[57] Jerzy Kowalski, Bartosz Krawczyk, and Michał Woźniak. Fault diagnosis of marine 4-stroke diesel engines using a one-vs-one extreme learning ensemble. *Engineering Applications of Artificial Intelligence*, 57:134–141, 2017.

[58] Wanxiang Li, Zhiwu Shang, Maosheng Gao, Shiqi Qian, Baoren Zhang, and Jie Zhang. A novel deep autoencoder and hyperparametric adaptive learning for imbalance intelligent fault diagnosis of rotating machinery. *Engineering Applications of Artificial Intelligence*, 102:104279, 2021.

[59] Yanshu Li. Exploring real-time fault detection of high-speed train traction motor based on machine learning and wavelet analysis. *Neural Computing and Applications*, pages 1–14, 2022.

[60] Pengfei Liang, Bin Wang, Guoqian Jiang, Na Li, and Lijie Zhang. Unsupervised fault diagnosis of wind turbine bearing via a deep residual deformable convolution network based on subdomain adaptation under time-varying speeds. *Engineering Applications of Artificial Intelligence*, 118:105656, 2023.

[61] Pengfei Liang, Wenhui Wang, Xiaoming Yuan, Siyuan Liu, Lijie Zhang, and Yiwei Cheng. Intelligent fault diagnosis of rolling bearing based on wavelet transform and improved resnet under noisy labels and environment. *Engineering Applications of Artificial Intelligence*, 115:105269, 2022.

[62] Beibei Liu, Gemma Mason, Julian Hodgson, Yiying Tong, and Mathieu Desbrun. Model-reduced variational fluid simulation. *ACM Transactions on Graphics (TOG)*, 34(6):1–12, 2015.

[63] Bolan Liu, Changlu Zhao, Fujun Zhang, Tao Cui, and Jianyun Su. Misfire detection of a turbocharged diesel engine by using artificial neural networks. *Applied thermal engineering*, 55(1-2):26–32, 2013.

[64] Xiaoyang Liu, Shulin Liu, Jiawei Xiang, Ruixue Sun, and Yuan Wei. An ensemble and shared selective adversarial network for partial domain fault diagnosis

of machinery. *Engineering Applications of Artificial Intelligence*, 113:104906, 2022.

[65] Xu Liu, Wei Peng, Zhiqiang Gong, Weien Zhou, and Wen Yao. Temperature field inversion of heat-source systems via physics-informed neural networks. *Engineering Applications of Artificial Intelligence*, 113:104902, 2022.

[66] Xu Liu, Xiaoya Zhang, Wei Peng, Weien Zhou, and Wen Yao. A novel meta-learning initialization method for physics-informed neural networks. *Neural Computing and Applications*, 34(17):14511–14534, 2022.

[67] Yunqing Liu, Jinlei Liu, Chengjin Qin, Yanrui Jin, Zhiyuan Li, Liqun Zhao, and Chengliang Liu. A deep learning-based acute coronary syndrome-related disease classification method: a cohort study for network interpretability and transfer learning. *Applied Intelligence*, 53(21):25562–25580, 2023.

[68] C Ludwig and M Ayoubi. Fault detection schemes for a diesel engine turbocharger. In *Proceedings of 1995 American Control Conference-ACC'95*, volume 3, pages 2118–2122. IEEE, 1995.

[69] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490*, 2019.

[70] Pavlo Lyubarskyy and Dirk Bartel. 2d cfd-model of the piston assembly in a diesel engine for the analysis of piston ring dynamics, mass transport and friction. *Tribology International*, 104:352–368, 2016.

[71] Lokesh Malviya and Sandip Mal. A novel technique for stress detection from eeg signal using hybrid deep learning model. *Neural Computing and Applications*, 34(22):19819–19830, 2022.

[72] Ivan Markovsky and Sabine Van Huffel. Overview of total least-squares methods. *Signal processing*, 87(10):2283–2302, 2007.

[73] Fadila Maroteaux and Charbel Saad. Combined mean value engine model and crank angle resolved in-cylinder modeling with nox emissions model for real-time diesel engine simulations at high engine speed. *Energy*, 88:515–527, 2015.

[74] Pradeep Menon and Mayank Mittal. Modeling and simulation of diesel engines using cfd and its applications in optimizing various in-cylinder techniques. In *Engine Modeling and Simulation*, pages 89–143. Springer, 2021.

[75] David D Morrison. Optimization by least squares. *SIAM Journal on Numerical Analysis*, 5(1):83–88, 1968.

[76] Dimitrios Moshou, Athanasios Natsis, Dimitrios Kateris, Xanthoula-Eirini Pantazi, Ioannis Kalimanis, and Ioannis Gravalos. Fault detection of fuel injectors based on one-class classifiers. *Modern Mechanical Engineering*, 2014, 2013.

[77] Christian Moya and Guang Lin. Dae-pinn: a physics-informed neural network model for simulating differential algebraic equations with application to power networks. *Neural Computing and Applications*, 35(5):3789–3804, 2023.

[78] Nikhil Muralidhar, Mohammad Raihanul Islam, Manish Marwah, Anuj Karpatne, and Naren Ramakrishnan. Incorporating prior domain knowledge into deep neural networks. In *2018 IEEE international conference on big data (big data)*, pages 36–45. IEEE, 2018.

[79] Peter Nabende and Tom Wanyama. An expert system for diagnosing heavy-duty diesel engine faults. In *Advances in computer and information sciences and engineering*, pages 384–389. Springer, 2008.

[80] Thi Nguyen Khoa Nguyen, Thibault Dairay, Raphaël Meunier, and Mathilde Mougeot. Physics-informed neural networks for non-newtonian fluid thermo-mechanical problems: An application to rubber calendering process. *Engineering Applications of Artificial Intelligence*, 114:105176, 2022.

[81] Hongli Niu, Kunliang Xu, and Weiqing Wang. A hybrid stock price index forecasting model based on variational mode decomposition and lstm network. *Applied Intelligence*, 50:4296–4309, 2020.

[82] Oameed Noakoasteen, Shu Wang, Zhen Peng, and Christos Christodoulou. Physics-informed deep neural networks for transient electromagnetic analysis. *IEEE Open Journal of Antennas and Propagation*, 1:404–412, 2020.

[83] Chady Nohra, Hassan Noura, Chady El Moucary, and Rafic Younes. A nonlinear approach with gain schedule control adaptation for a complete-model diesel-engine diagnosis. In *2008 3rd International Symposium on Communications, Control and Signal Processing*, pages 689–696. IEEE, 2008.

[84] Chady Nohra, Hassan Noura, and Rafic Younes. A linear approach with μ-analysis control adaptation for a complete-model diesel-engine diagnosis. In *2009 Chinese Control and Decision Conference*, pages 5415–5420. IEEE, 2009.

[85] David Nyman. Injector diagnosis based on engine angular velocity pulse pattern recognition, 2020.

[86] Hanqing Ouyang, Zhicheng Zhu, Kuangqi Chen, Beichen Tian, Biao Huang, and Jia Hao. Reconstruction of hydrofoil cavitation flow based on the chain-style physics-informed neural network. *Engineering Applications of Artificial Intelligence*, 119:105724, 2023.

[87] Ismail Enes Parlak and Erdal Emel. Deep learning-based detection of aluminum casting defects and their types. *Engineering Applications of Artificial Intelligence*, 118:105636, 2023.

[88] Francisco Payri, Pablo Olmeda, Jaime Martín, and Antonio García. A complete 0d thermodynamic predictive model for direct injection diesel engines. *Applied Energy*, 88(12):4632–4641, 2011.

[89] Prashanth Pillai, Anshul Kaushik, Shivanand Bhavikatti, Arjun Roy, and Virendra Kumar. A hybrid approach for fusing physics and data for failure prediction. *International Journal of Prognostics and Health Management*, 7(4), 2016.

[90] Li Qi, Yuwei Ren, Yixian Fang, and Jinglin Zhou. Two-view lstm variational auto-encoder for fault detection and diagnosis in multivariable manufacturing processes. *Neural Computing and Applications*, 35(29):22007–22026, 2023.

[91] Hongbo Qu, Hongchen Liu, Shuang Jiang, Jiabin Wang, and Yonghong Hou. Discovery the inverse variational problems from noisy data by physics-constrained machine learning. *Applied Intelligence*, 53(9):11229–11240, 2023.

[92] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[93] Oleg Rogov. Fault detection in diesel engines using deep learning and qarma algorithms. 2022.

[94] Patrick Sauer. Model-based fault detection and diagnosis for the fuel system of a six-cylinder heavy duty diesel engine. 2021.

[95] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management*, pages 1–9. IEEE, 2008.

[96] Basheer Shaheen, Ádám Kocsis, and István Németh. Data-driven failure prediction and rul estimation of mechanical components using accumulative artificial neural networks. *Engineering Applications of Artificial Intelligence*, 119:105749, 2023.

[97] Mingqi Shao, Jin Wang, and Sibo Wang. The intelligent fault diagnosis of diesel engine based on the ensemble learning. In *Journal of Physics: Conference Series*, volume 1549, page 042106. IOP Publishing, 2020.

[98] Abhishek Sharma, V Sugumaran, and S Babu Devasenapati. Misfire detection in an ic engine using vibration signal and decision tree algorithms. *Measurement*, 50:370–380, 2014.

[99] Smriti Sharma and Subhamoy Sen. Real-time structural damage assessment using lstm networks: regression and classification approaches. *Neural Computing and Applications*, 35(1):557–572, 2023.

[100] Sheng Shen, Hao Lu, Mohammadkazem Sadoughi, Chao Hu, Venkat Nemani, Adam Thelen, Keith Webster, Matthew Darr, Jeff Sidon, and Shawn Kenny. A physics-informed deep learning approach for bearing fault detection. *Engineering Applications of Artificial Intelligence*, 103:104295, 2021.

[101] Shubhendu Kumar Singh. *Hybrid Machine Learning Approach for Predictive Modeling of Complex Systems*. PhD thesis, State University of New York at Buffalo, 2019.

[102] Shubhendu Kumar Singh, Ruoyu Yang, Amir Behjat, Rahul Rai, Souma Chowdhury, and Ion Matei. Pi-lstm: Physics-infused long short-term memory network. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 34–41. IEEE, 2019.

[103] Sneha Singh, Sagar Potala, and Amiya R Mohanty. An improved method of detecting engine misfire by sound quality metrics of radiated sound. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(12):3112–3124, 2019.

[104] Anna G Stefanopoulou, Ilya Kolmanovsky, and James S Freudenberg. Control of variable geometry turbocharged diesel engines for reduced emissions. *IEEE transactions on control systems technology*, 8(4):733–745, 2000.

[105] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[106] Congbiao Sui, Enzhe Song, Douwe Stapersma, and Yu Ding. Mean value modelling of diesel engine combustion based on parameterized finite stage cylinder process. *Ocean Engineering*, 136:218–232, 2017.

[107] G Sujesh and S Ramesh. Modeling and control of diesel engines: A systematic review. *Alexandria engineering journal*, 57(4):4033–4048, 2018.

[108] József Z Szabó and Péter Bakucz. Real-time misfire detection of large gas engine using big data analytics. In *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 000215–000220. IEEE, 2018.

[109] Mehdi Taghizadeh, Mohammad Amin Nabian, and Negin Alemazkoor. Multifidelity physics-informed generative adversarial network for solving partial differential equations. *Journal of Computing and Information Science in Engineering*, pages 1–15, 2023.

[110] SS Talebi, A Madadi, AM Tousi, and M Kiaee. Micro gas turbine fault detection and isolation with a combination of artificial neural network and off-design performance analysis. *Engineering Applications of Artificial Intelligence*, 113:104900, 2022.

[111] Jianfeng Tao, Chengjin Qin, Weixing Li, and Chengliang Liu. Intelligent fault diagnosis of diesel engines via extreme gradient boosting and high-accuracy time–frequency information of vibration signals. *Sensors*, 19(15):3280, 2019.

[112] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 847–855, 2013.

[113] Peter Turney and Michael Halasz. Contextual normalization applied to aircraft gas turbine engine diagnosis. *Applied Intelligence*, 3:109–129, 1993.

[114] JA Twiddle and NB Jones. Fuzzy model-based condition monitoring and fault diagnosis of a diesel engine cooling system. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 216(3):215–224, 2002.

[115] Julian Von Schleinitz, Michael Graf, Wolfgang Trutschnig, and Andreas Schröder. Vasp: An autoencoder-based approach for multivariate anomaly detection and robust time series prediction with application in motorsport. *Engineering Applications of Artificial Intelligence*, 104:104354, 2021.

[116] Johan Wahlström and Lars Eriksson. Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 225(7):960–986, 2011.

[117] Shengdong Wang, Zhenbao Liu, Zhen Jia, and Zihao Li. Incipient fault diagnosis of analog circuit with ensemble hkelm based on fused multi-channel and multi-scale features. *Engineering Applications of Artificial Intelligence*, 117:105633, 2023.

[118] Yanyan Wang, Ning Ren, Jin Li, Bin Liu, Qingtao Si, and Ruitian Zhang. Research on fault detection and diagnosis method of diesel engine air system based on deep learning. In *International Conference on Data Mining and Big Data*, pages 328–341. Springer, 2021.

[119] Yunsong Wang and Fulei Chu. Real-time misfire detection via sliding mode observer. *Mechanical Systems and Signal Processing*, 19(4):900–912, 2005.

[120] N Watson. Dynamic turbocharged diesel engine simulator for electronic control system development. 1984.

[121] Hilde JP Weerts, Andreas C Mueller, and Joaquin Vanschoren. Importance of tuning hyperparameters of machine learning algorithms. *arXiv preprint arXiv:2007.07588*, 2020.

[122] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[123] Zheng Xiaojun, Yang Shuzi, Zhou Anfa, and Shi Hanmin. A knowledge-based diagnosis system for automobile engines. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1042–1047. IEEE, 1988.

[124] Xiaojian Xu, Xinping Yan, Chenxing Sheng, Chengqing Yuan, Dongling Xu, and Jianbo Yang. A belief rule-based expert system for fault diagnosis of marine diesel engines. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(2):656–672, 2017.

[125] Zhaoyi Xu, Yanjie Guo, and Joseph Homer Saleh. A physics-informed dynamic deep autoencoder for accurate state-of-health prediction of lithium-ion battery. *Neural Computing and Applications*, 34(18):15997–16017, 2022.

[126] Gen-Ting Yan and Guang-Fu Ma. Fault diagnosis of diesel engine combustion system based on neural networks. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, volume 5, pages 3111–3114. IEEE, 2004.

[127] Ruoyu Yang, Shubhendu Kumar Singh, Mostafa Tavakkoli, Nikta Amiri, Yongchao Yang, M Amin Karami, and Rahul Rai. Cnn-lstm deep learning architecture for computer vision-based modal frequency detection. *Mechanical Systems and signal processing*, 144:106885, 2020.

[128] Yatao Yang, Yuqing He, Haolin Guo, Ziliang Chen, and Li Zhang. Semantic segmentation supervised deep-learning algorithm for welding-defect detection of new energy batteries. *Neural Computing and Applications*, 34(22):19471–19484, 2022.

[129] Qunwang Yao, Yi Qin, Xin Wang, and Quan Qian. Multiscale domain adaption models and their application in fault transfer diagnosis of planetary gearboxes. *Engineering Applications of Artificial Intelligence*, 104:104383, 2021.

[130] Jianbo Yu and Yue Zhang. Challenges and opportunities of deep learning-based process fault detection and diagnosis: a review. *Neural Computing and Applications*, 35(1):211–252, 2023.

[131] Tong Yu and Hong Zhu. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020.

[132] Xianbiao Zhan, Huajun Bai, Hao Yan, Rongcai Wang, Chiming Guo, and Xisheng Jia. Diesel engine fault diagnosis method based on optimized vmd and improved cnn. *Processes*, 10(11):2162, 2022.

[133] Qi Zhang, Zhenliang Ma, Pengfei Zhang, Erik Jenelius, Xiaolei Ma, and Yuanqiao Wen. User-station attention inference using smart card data: a knowledge graph assisted matrix decomposition model. *Applied Intelligence*, pages 1–17, 2023.

[134] Haipeng Zhao, Zhiwei Mao, Jinjie Zhang, Xudong Zhang, Nanyang Zhao, and Zhinong Jiang. Multi-branch convolutional neural networks with integrated cross-entropy for fault diagnosis in diesel engines. *Measurement Science and Technology*, 32(4):045103, 2021.

[135] Xiaoyu Zhao, Zhiqiang Gong, Yunyang Zhang, Wen Yao, and Xiaoqian Chen. Physics-informed convolutional neural networks for temperature field prediction of heat source layout without labeled data. *Engineering Applications of Artificial Intelligence*, 117:105516, 2023.

[136] Yike Zhao, Xin Zhang, Jiaxu Wang, Lei Wu, Zhiwen Liu, and Lei Wang. A new data fusion driven-sparse representation learning method for bearing intelligent diagnosis in small and unbalanced samples. *Engineering Applications of Artificial Intelligence*, 117:105513, 2023.

[137] Guo-qiang Zhong, Huai-yu Wang, Kun-yang Zhang, and Bao-zhu Jia. Fault diagnosis of marine diesel engine based on deep belief network. In *2019 Chinese Automation Congress (CAC)*, pages 3415–3419. IEEE, 2019.

[138] Zhicheng Zhu, Jia Hao, Jin Huang, and Biao Huang. Bc-pinn: an adaptive physics informed neural network based on biased multiobjective coevolutionary algorithm. *Neural Computing and Applications*, 35(28):21093–21113, 2023.

[139] Navid Zobeiry and Keith D Humfeld. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Engineering Applications of Artificial Intelligence*, 101:104232, 2021.