FEDERAL UNIVERSITY OF TECHNOLOGY — PARANÁ

HILKIJA GAIUS TOSSO

MODEL-BASED AI APPROACH FOR KNOCK CONTROL IN SPARK IGNITION ENGINE BY VIBRATION BLOCK MODELING

PONTA GROSSA

2023

HILKIJA GAIUS TOSSO

MODEL-BASED AI APPROACH FOR KNOCK CONTROL IN SPARK IGNITION ENGINE BY VIBRATION BLOCK MODELING

Abordagem de IA Baseada em Modelo para o Controle de Detonações em Motores de Ignição por Centelha através da Modelagem por Bloco de Vibração

Dissertation presented as a requirement for obtaining the title of Master in Electrical Engineering from the Federal University of Technology — Paraná (UTFPR). Concentration Area: Control and instrumentation.

Advisor: Prof. Ph.D. Max Mauro Dias Santos

PONTA GROSSA

2023

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa

HILKIJA GAIUS TOSSO

MODEL-BASED AI APPROACH FOR KNOCK CONTROL IN SPARK IGNITION ENGINE BY VIBRATION BLOCK MODELING

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Controle E Processamento De Energia.

Data de aprovação: 22 de Dezembro de 2023

Dra. Fernanda Cristina Correa, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Joao Francisco Justo Filho, Doutorado - Usp-Universidade de São Paulo

Dr. Rui Tadashi Yoshino, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 22/12/2023.

To my loving family and inspiring mentors, who have unwaveringly supported and guided me in every step of this journey. This accomplishment is not solely mine, but a reflection of your belief, love, and dedication.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to God, the guiding force in my life, for providing me with the strength, wisdom, and perseverance needed to complete this journey.

I am profoundly grateful to Prof. Max Mauro Dias Santos, Ph.D and Saulo Anderson Bibiano Jardim. Their expertise, understanding, and patience have significantly enriched my academic experience.

I would also like to thank the UTFPR Support Foundation - FUNTEF, the Foundation Scholarship Programme Araucária, and Renault do Brasil. Their financial support and belief in my research fueled this project's progression and successful completion.

I extend my special thanks to the academic community at UTFPR and our department PPGEE-PG. Their commitment to fostering an environment of intellectual curiosity and growth has been invaluable. The knowledge and skills I've acquired here will undoubtedly serve me well in my future endeavors.

I would also like to express my heartfelt thanks to my friends and colleagues. Their unwavering encouragement and good humor have been a constant source of joy throughout the course of my studies.

Lastly, I must express my deepest gratitude to my family for their continuous support and endless encouragement throughout my years of study. This achievement would not have been possible without them. I dedicate this work and give special thanks to my family for inspiring me to reach my full potential.

I recognize that this research would not have been possible without the collaboration and shared intellectual curiosity of all those mentioned. Thank you all for your part in my journey.

“Progress is impossible without change, and those who cannot change their minds cannot change anything.” (SHAW, 1856 - 1950)

ABSTRACT

Engine Knock poses a significant threat to the integrity and performance of spark ignition engines, leading to decreased efficiency, power, and overall performance. Traditional control strategies utilize knock sensors to identify and mitigate these adverse effects, although these may not always provide accurate or timely responses. This presents an innovative model-based approach to knock control, utilizing Artificial Intelligence (AI) techniques. Specifically, we propose a time-series Artificial Neural Network (ANN) model designed to simulate SI Engine Cylinder block vibrations. This study investigates transformer models, Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN) models. These models are trained using vibrational data from engine blocks, thereby developing a comprehensive understanding of the unique behaviors of the engine. Among the models, the Attention LSTM model showed slightly superior performance, outperforming the other two models across the full range of engine vibration conditions. Despite promising results, further refinement and tuning of these models are required for real-world implementation. The proposed AI model offers a promising direction for advancements in knock control technology, with potential implications for enhancing engine durability, performance, and fuel efficiency.

Keywords: model-based knock control; engine control unit calibration; artificial neural networks; attention models.

RESUMO

A Detonação do Motor representa uma ameaça significativa para a integridade e o desempenho dos motores de ignição por faísca, levando a uma diminuição da eficiência, potência e desempenho geral. As estratégias de controle tradicionais utilizam sensores de detonação para identificar e mitigar esses efeitos adversos, embora estes nem sempre possam fornecer respostas precisas ou oportunas. Este trabalho apresenta uma abordagem inovadora baseada em modelos para o controle de detonação, utilizando técnicas de Inteligência Artificial. Especificamente, propomos um modelo de Rede Neural Artificial de séries temporais projetado para simular as vibrações do bloco do cilindro do motor de Ignição por Faísca. Este estudo investiga modelos baseados em mecanismos de atenção, *Long Short Term Memory* (LSTM), e modelos de Rede Neural Convolucional. Esses modelos são treinados usando dados vibracionais de blocos de motores, desenvolvendo assim uma compreensão abrangente dos comportamentos únicos do motor. Entre os modelos, o modelo LSTM com mecanismos de atenção apresentou um desempenho ligeiramente superior, superando os outros dois modelos em toda a gama de condições de vibração do motor. Apesar dos resultados promissores, são necessários mais aprimoramentos e ajustes desses modelos para a implementação no mundo real. O modelo de IA proposto oferece uma direção promissora para avanços na tecnologia de controle de detonação, com implicações potenciais para melhorar a durabilidade, o desempenho e a eficiência de combustível.

Palavras-chave: controle de detonação baseado em modelo; calibração da unidade de controle do motor; redes neurais artificiais; mecanismos de atenção.

LIST OF ALGORITHMS

Algorithm 1 – Preprocess Data for Input and Target Sequences 65
Algorithm 2 – Adam Optimizer Algorithm 68

LIST OF FIGURES

Figure 1 – Control Structure of the Spark Ignition Engine. 13
Figure 2 – Knock detection methods 17
Figure 3 – Knock Level and Noise Modeling Proposition 20
Figure 4 – Typical combustion pressure traces. 23
Figure 5 – Possible signals to use to characterize the knock event. 24
Figure 6 – Schematic representation of the EMS control system in a SI engine. 30
Figure 7 – Transformer model architecture. 40
Figure 8 – Knock Level and Noises signal distribution at 1700 rpm. 46
Figure 9 – Knock Level and Noises signal distribution at 2700 rpm. 48
Figure 10 – Knock Level and Noises signal distribution at 3600 rpm. 49
Figure 11 – Knock Level and Noises signal distribution at 4700 rpm. 50
Figure 12 – Spearman's Correlation Analysis at 1700 rpm in Full Load. . . . 52
Figure 13 – Spearman's Correlation Analysis at 2700 rpm in Full Load. . . . 53
Figure 14 – Spearman's Correlation Analysis at 3600 rpm in Full Load. . . . 54
Figure 15 – Spearman's Correlation Analysis at 4700 rpm in Full Load. . . . 55
Figure 16 – Spearman's Correlation Analysis at 2700 rpm with Throttle partially closed. 56
Figure 17 – Spearman's Correlation Analysis at 3600 rpm with Throttle partially closed. 57
Figure 18 – Spearman's Correlation Analysis at 4700 rpm with Throttle partially closed. 58
Figure 19 – 1D-CNN Architecture. 59
Figure 20 – Bidirectional LSTM Model Architecture. 61
Figure 21 – Attention LSTM Model Development Architecture. 63
Figure 22 – Box Plot of MAPE per Windows Length for each model in 1700 rpm at Full Load. 72
Figure 23 – Comparing Models Performance per Speed. 76

LIST OF TABLES

Table 1 – 1D-CNN performance per combination of Number of Filter and Kernel Size in 1700 rpm at Full Load with Windows length of 25. 71
Table 2 – Bi-LSTM performance per combination of each Layers units in 1700 rpm at Full Load with Windows length of 25. 73
Table 3 – Attention LSTM performance per combination of each Layers units in 1700 rpm at Full Load with Windows length of 25. 73
Table 4 – 1D-CNN Model Evaluation 74
Table 5 – Bi-LSTM Model Evaluation 75
Table 6 – Attention LSTM Model Evaluation 75
Table 7 – MAPE (%) of each model per SI Engine cylinder vibration range in all operation condition. 77

LIST OF ABBREVIATIONS AND INITIALS

Adam	Adaptive Moment Estimation
AE	Acoustic Eission
AFR	Air-to-fuel ratio
ANN	Artificial Neural Network
Bi-LSTM	Bidirectional Long Short-Term Memory
BPANNs	Back Propagation Artificial Neural Networks
CA	Crank Angle
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Networks
DISI	Direct Injected Spark Ignition
DoE	Design of Experiment
DWT	Discrete Wavelet Transform
ECU	Engine Control Unit
ICE	Internal Combustion Engines
IMEP	Indicated Mean Effective Pressure
IMPG	Integral Modulus of Pressure Gradient
IMPO	Integral Modulus of Pressure Oscillations
KI	Knock Intensity
KP	Knock Pressure
LSTM	Long Short-Term Memory
MAPO	Maximum Amplitude of Pressure Oscillations
MBC	Model-Based Calibration
PFP	Firing Pressure
PP	Peak Pressure
PP	Peak Pressure Timing
RCMs	Rapid compression machines
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SA	Spark Advance
SI	Spark Ignition
TDC	Top Dead Center
UTFPR	Universidade Tecnológica Federal do Paraná
VVT	Variable Valve Timings

SUMMARY

1 INTRODUCTION . . . 13
1.1 Engine knock . . . 15
1.2 Machine Learning for Time-Series Modeling in SI Engines . . . 17
1.3 Justification . . . 18
1.4 Proposed Method . . . 19
1.5 Objectives . . . 20
1.5.1 General Objectives . . . 20
1.5.2 Specific Objectives . . . 20
1.6 Contributions . . . 20
1.7 Thesis structure . . . 21

2 LITERATURE REVIEW . . . 22
2.1 Overview of SI Engine Knock Phenomenon. . . . 22
2.2 Knock Detection and Control Fundamentals . . . 26
2.2.1 Knock Detection . . . 27
2.2.2 Knock Control Strategies . . . 30
2.3 SI Engine Knock Intensity Modeling . . . 32
2.3.1 Related Works . . . 32
2.4 Time Serie Forecasting using ANN . . . 35
2.4.1 Introduction to ANN Models . . . 35
2.4.2 Time Serie Forecasting with Long Short-Term Memory (LSTM) . . . 36
2.4.3 Time Serie Forecasting with Convolutional Neural Networks (CNN) . . . 38
2.4.4 Transformers for Time Serie Forecasting . . . 39

3 MATERIAL AND METHODS . . . 43
3.1 Data Collection . . . 44
3.1.1 Dataset Description . . . 45
3.1.2 Correlation Analysis . . . 50
3.2 Models Architectures . . . 54
3.2.1 1-D CNN Model Development . . . 57
3.2.2 Bidirectional LSTM Model Development . . . 59
3.2.3 Attention LSTM Model Development . . . 61
3.3 Experiment Setup . . . 63
3.3.1 Data Prepocessing . . . 64
3.3.2 Hyper-parameters Grid Search . . . 65
3.3.3 Model Training . . . 67
3.3.4 Models Evaluation . . . 69

4 RESULTS AND ANALYSIS . . . 71

5 FINAL CONSIDERATIONS . . . 79

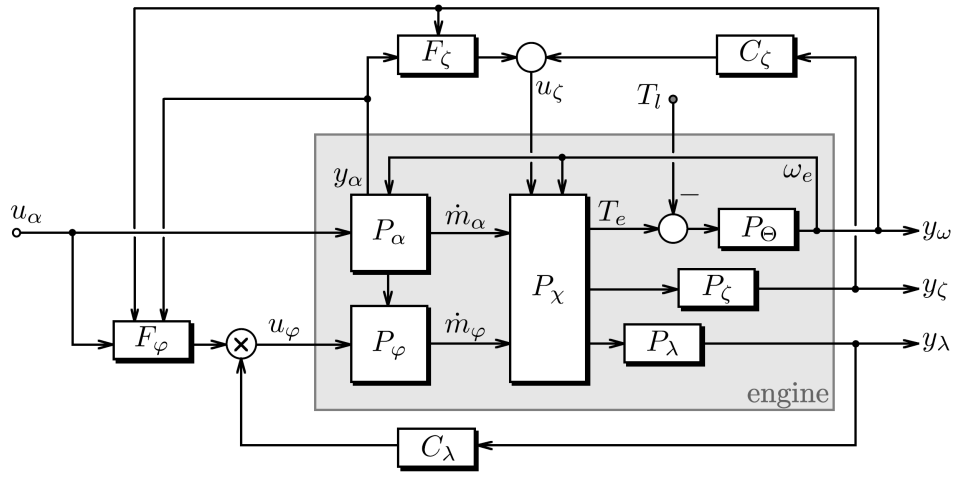
REFERENCES . . . 81

1 INTRODUCTION

The world has become increasingly aware of the pressing need for environmental sustainability, and the transition to electrification of our transportation systems has been gaining momentum. Nonetheless, a significant portion of the global transport infrastructure is still powered by internal combustion engines (REITZ *et al.*, 2020). Hence, enhancing combustion engine efficiency and reducing fossil fuel dependence are crucial research areas due to their significant impact on our environment and economy.

Spark Ignition (SI) Engines are intricate machines whose efficiency and performance rely on controlling numerous parameters. Control parameter values, typically stored in lookup tables and adjusted across the engine's operational range, are set via test bench mapping. Occasionally, a feedback control strategy is employed to optimize these values and enhance engine performance (CORTI; MORO, 2007; DIAS *et al.*, 2018).

Figure 1 – Control Structure of the Spark Ignition Engine.



Source: Guzzella and Onder (2009)

According to Guzzella and Onder (GUZZELLA; ONDER, 2009), one can conceive a simplified control-oriented model for an SI engine, as depicted in Fig. 1. The model comprises significant components such as the fuel and air paths, which are denoted as P_φ and P_α, respectively. These paths primarily define the mixture entering the cylinder. Additionally, the combustion block P_χ is responsible for determining the torque produced by the engine.

Other critical outputs include the knock signal y_ζ and the engine-out air/fuel ratio y_λ, which are measured by a knock sensor P_ζ and a λ sensor P_λ respectively. These sensors are ideally positioned as close as possible to the exhaust valves. The engine speed ω_e is produced by the block P_Θ, which incorporates the engine's rotational inertia and is influenced by the engine torque T_e and the load torque T_l.

Significant control loops in the model, as indicated in Fig. 1, include the fuel-injection feedforward loop, the air/fuel ratio feedback loop, the ignition angle feedforward loop, and the knock feedback loop. Many engine systems also incorporate other feedforward or feedback loops, such as idle and cruise speed control, exhaust gas recirculation for emission reduction during a cold start or for lean-burn engines, secondary air injection for faster catalyst light-off, and canister purge management to prevent hydrocarbon evaporation.

The calibration of internal combustion engines (ICE) is an essential part of engine optimization, as it involves finding the best settings for various adjustable parameters such as the air-to-fuel ratio AFR, spark advance, and variable valve timings (VVT) (YU *et al.*, 2022). This calibration maximizes output power while minimizing fuel consumption, pollutant emissions, and noise. The process is complex as it necessitates attention to several intricate engine systems, such as the combustion and gas exchange systems, and their interactions.

A widely used tool in this process is the dynamometer (dyno), which enables engine performance evaluation during the engine development process, leading to the calibration of engines before they reach the customers. The traditional calibration method involves manual test bench-based dynamometer calibration, which resembles a grid search for optimization. This method involves dividing the parameter space into a grid for each operating condition/operating point and manually measuring engine performance at different grid knots. The grid knot yielding the optimal engine performance is selected for the current operating point, and this process is repeated for all operating points defined by a specific driving cycle.

In essence, the engine calibration process can be likened to a problem-solving task to find the most effective correlation or mapping function $m : X \rightarrow Y$. Here, X represents all the conceivable situations that an engine could encounter during operation, while Y illustrates all the achievable adjustments that can be made to the engine's parameters. The task is to establish the optimal connection between these

two realms. Considering the time-consuming nature of engine performance evaluation on dyno, calibration can be a costly optimization problem. With the rise in adjustable engine parameters due to advances in engine technology and increasingly stringent fuel economy and emissions regulations, the complexity of engine calibration problems has increased (YU *et al.*, 2022).

The Design of Experiment (DoE) concept has been widely used in off-board calibration to increase efficiency. This procedure involves selecting a limited number of test points for evaluation on dyno, and these evaluated test points are used for model-based calibration (MBC). This method saves the calibration budget compared to the traditional manual method, which evaluates all points on dyno.

However, certain variables pose a challenge to model due to their inherent complexity, unpredictability, and insufficient theoretical knowledge about them (TOSSO *et al.*, 2022). One such example is the Knock model, which presents significant difficulties to researchers worldwide.

Engine knock is a crucial phenomenon for the engine development and calibration process. For combustion chamber design during engine development, it is imperative to ensure resilience against knocking combustion, especially for engine operation with a wide range of fuels. Likewise, during the engine production stage, the engine control system must be tuned to enable non-knocking combustion across various operating conditions.

Knocking reduces engine performance and efficiency in SI engines, restricting the maximum compression ratio. Beyond these challenges, engine knock can also lead to engine damage when severe, and being a noise source, it is commonly viewed as a drivability issue. Therefore, investigating engine knock is of utmost importance (MAURYA; MAURYA, 2019).

In the following sections, we will delve deeper into the phenomenon of engine knock and explore the potential of machine learning techniques for time-series modeling in Spark Ignition engines.

1.1 Engine knock

Knock in SI engines is attributed to an anomalous combustion phenomenon that, if prolonged, can result in detrimental effects such as piston ring breakage, cylinder

head erosion, piston crown and top land erosion, piston melting, restricted engine compression ratio, or vehicle acceleration performance, elevated air pollution, reduced engine efficiency, significant increase in engine-specific fuel consumption, potential structural damage to the engine over an extended period, and noise generation (ZHEN *et al.*, 2012).

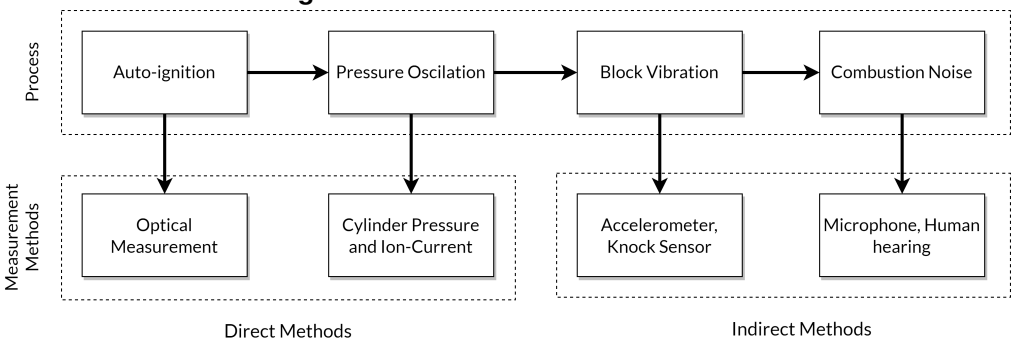
The challenges associated with comprehending knock stem from the cyclical variability of the knock phenomenon. Specifically, the timing of knock onset within each cycle, as well as the rate of energy release and changes in pressure upon its occurrence, contribute to the complexity of this phenomenon (CHUN; HEYWOOD, 1989).

Cyclic pressure fluctuations and other unobservable factors, such as residual mass variations or temperature hot spots, cause the unpredictable nature of knocking combustion in reciprocating engines. As a result, accurately predicting engine knocking is nearly impossible for practical applications. To address this issue, current knock control algorithms utilize a combination of open-loop control actions and stochastic rules, which adapt engine operation based on previous knock detection events. Accurate knock detection is crucial for conventional controller performance, as failure to detect knock can result in engine damage, while false detection can reduce engine efficiency (MAURYA; MAURYA, 2019).

Various techniques currently exist for knock detection, including auto-ignition detection, ion current detection, in-cylinder combustion pressure detection, engine block vibration detection, and combustion noise detection. In particular, in-cylinder combustion pressure detection is widely used in laboratory settings for accurate knock detection in individual cylinders due to its high signal-to-noise ratio and sensitivity despite its high cost. The literature categorizes these knock detection methods into direct and indirect approaches, as indicated in Fig. 2. Direct methods rely on in-cylinder pressure measurements to detect the high-frequency knock signature related to the resonant frequencies of the combustion chamber generated by radial or circumferential vibration modes. These frequencies depend on the speed of sound in the combustion chamber and cylinder bore. On the other hand, indirect methods involve advanced signal processing techniques utilizing the cylinder block vibration signals gathered by accelerometers, which have benefits such as affordability, excellent durability, and sensitivity. As such, they are increasingly adopted as practical solutions for mass-produced engines. However, detecting cycle-by-cycle and cylinder-by-cylinder knock

conditions is crucial for their application in on-board control systems, while identifying optimal sensor locations, particularly for multi-cylinder engines, requires careful attention (SIANO; BOZZA, 2013).

Figure 2 – Knock detection methods



Source: Adapted from Maurya and Maurya (2019)

In the present works, we delve into the application of Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Attention Model architectures for modeling knock noises (cylinder blocks vibration) in spark ignition engines. The anticipated benefits of this exploration are multifold. Not only does this approach potentially enhance the predictability of engine knock, but it also provides a robust mathematical representation of the knock phenomenon. This predictive modeling could allow for real-time estimation of knock likelihood under current operating conditions, enabling a proactive approach to knock control. As a result, this approach could facilitate real-time fine-tuning of engine operation, prevent knock occurrence, and ensure the engine operates near its maximum efficiency point. Furthermore, integrating such a model into the engine control unit (ECU) could pave the way for developing advanced control algorithms, thereby improving engine performance, efficiency, and durability.

1.2 Machine Learning for Time-Series Modeling in SI Engines

Advancements in Machine Learning (ML) offer promising avenues for the creation of robust, predictive models in a variety of fields, including the automotive industry (FU *et al.*, 2022; TURKSON *et al.*, 2016; GÖLCÜ *et al.*, 2005; BHATT; SHRIVASTAVA, 2022). Specifically, ML algorithms show high potential in time-series modeling, which is increasingly relevant to SI engines (XIE *et al.*, 2023; OFNER *et al.*, 2022; SIANO; PANZA; D'AGOSTINO, 2015).

The fundamental characteristic of time-series data, its sequential nature, makes it suitable for applying ML techniques. In the context of SI engines, data points are a sequence of measurements taken over time, capturing the dynamic behavior of various engine parameters. Machine Learning offers powerful tools capable of handling complex temporal dependencies. Techniques such as Transformers, LSTM networks, and CNNs are particularly well-suited for this task (MITTELMAN, 2015; ABBASIMEHR; PAKI, 2022; KARIM *et al.*, 2019; WEN *et al.*, 2022; ZENG *et al.*, 2023; AHMED *et al.*, 2023). These techniques can recognize patterns over various time scales, capture long-term dependencies, and model non-linear relationships, making them exceptionally powerful in time-series prediction.

In the context of SI engine control, these ML-driven time-series models can predict the engine's future behavior and adapt control strategies accordingly. For instance, predicting engine knock, a phenomenon that can cause severe engine damage, ahead of time would allow for preemptive corrective measures. Such a proactive approach could significantly enhance engine performance and longevity.

Moreover, creating these predictive models could enable a more in-depth understanding of complex phenomena like engine knocks. These models could shed light on the underlying mechanisms driving these phenomena by identifying key features and highlighting intricate relationships between various engine parameters. This additional insight could prove invaluable in designing and optimizing future engine models.

Given these compelling benefits, there is a clear imperative for integrating ML techniques into SI engine control. Developing and implementing ML-driven time-series models could revolutionize engine calibration, enhancing efficiency, performance, and overall engine life.

1.3 Justification

Knock is a major constraint in realizing maximum engine performance and efficiency in internal combustion engines. This phenomenon not only restricts the engine's performance but, if not properly controlled, can also lead to severe engine damage. Thus, an accurate knock signal is critical for successful knock control strategies. Creating a robust Artificial Neural Network (ANN) model capable of accurately simulating the knock noise in Spark Ignition Engines can revolutionize how engine knock is managed. The

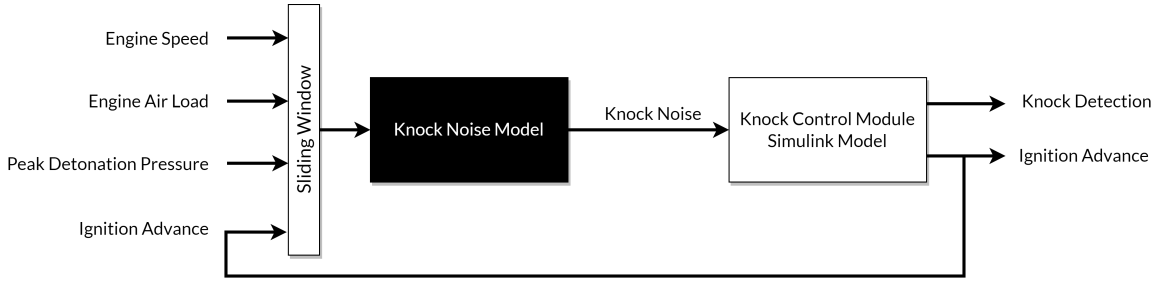
model will represent the knock phenomenon mathematically, facilitating more predictive and proactive control. Current control strategies are largely reactive, responding only to detected knock events. The predictive model developed in this research will allow for real-time estimation of knock likelihood under the current operating conditions, enabling a more proactive approach to knock control. This will allow for real-time fine-tuning of engine operation, preventing knock from occurring and ensuring the engine operates near its maximum efficiency point. Furthermore, integrating such a model into the engine control unit (ECU) could pave the way for developing advanced control algorithms, thereby improving engine performance, efficiency, and durability.

Moreover, with its significant reliance on biofuels like ethanol, the Brazilian energy matrix presents a compelling case for continued research and development in Internal Combustion Engine (ICE) technologies (MALAQUIAS *et al.*, 2019). The inherent properties of ethanol as a biofuel, coupled with the constant evolution and improvement of ICE technologies, make this an area of significant potential. While electric vehicles have gained much attention globally, it is important to consider the broader environmental, social, ethical, and economic impacts they bring. In contrast, using biofuels in advanced ICEs offers a more sustainable path for global mobility, particularly in Brazil's energy landscape.

1.4 Proposed Method

This research aims to devise a time-series ANN model capable of accurately simulating the Knock Noise in Spark Ignition Engines, as illustrated in Figure 3. This model will leverage the correlation between peak detonation pressure KP and engine block vibrations, as detected by a piezoelectric sensor, to predict knock likelihood. The current knock calibration process is complex and time-consuming, often requiring up to five weeks of engine development on the test bench. The proposed model will reduce the time necessary to calibrate the knock level, making the engine development process more efficient and cost-effective. Ultimately, this model could be integrated into the ECU, providing advanced, real-time knock control. This model aims to streamline the knock calibration workflow for automobile manufacturers in São José dos Pinhais.

Figure 3 – Knock Level and Noise Modeling Proposition



Source: Own authorship (2023)

.

1.5 Objectives

1.5.1 General Objectives

To model knock noises in spark ignition engines based on the correlation between peak detonation pressure and cylinder block vibrations in the SI engine using artificial neural network models, specifically the 1D-CNN, Bidirectional LSTM, and Attention LSTM models.

1.5.2 Specific Objectives

- Introduce a comprehensive overview of SI engine knocking.
- Show the current strategies for modeling knock likelihood and level.
- Make the dataset acquisition of SI Engine under different operating conditions.
- Perform processing and statistical analysis of collected data.
- Train, process, and optimize the models.
- Analysis and show the advantages of the strategy.

1.6 Contributions

This research offers several significant contributions to engine development and calibration. The primary contribution is the development of a time-series artificial neural network (ANN) model that can effectively simulate the knock noise in Spark ignition engines. This model can potentially reduce the time and resources required to adjust the knock level, thereby streamlining the engine calibration process.

Additionally, the proposed model provides a deeper understanding of the knock phenomenon. This enhanced understanding can lead to improved engine performance and efficiency. Furthermore, integrating this model into the ECU could pave the way for developing advanced control algorithms, improving engine performance, efficiency, and durability.

1.7 Thesis structure

The thesis will be structured as follows:

- Chapter 1: Introduction
- Chapter 2: Literature Review
- Chapter 3: Material and Methods
- Chapter 4: Results and Analysis
- Chapter 5: Final Considerations

Chapter 2 will comprehensively review the literature on knock detection and time-series modeling. Chapter 3 will describe the methodology to develop the proposed model approach. Chapter 4 will present and analyze the results. Finally, Chapter 5 will summarize the main contributions of the thesis and provide directions for future research.

2 LITERATURE REVIEW

In this chapter, a comprehensive review of the existing literature relevant to this study will be conducted. This will involve a deep dive into each key theme presented in this work. The aim is to provide a solid foundation of the concepts and definitions pertinent to these themes and an understanding of their context and applicability. Additionally, this section will showcase an overview of the various applications, demonstrating the significance and diversity of approaches found within the academic literature.

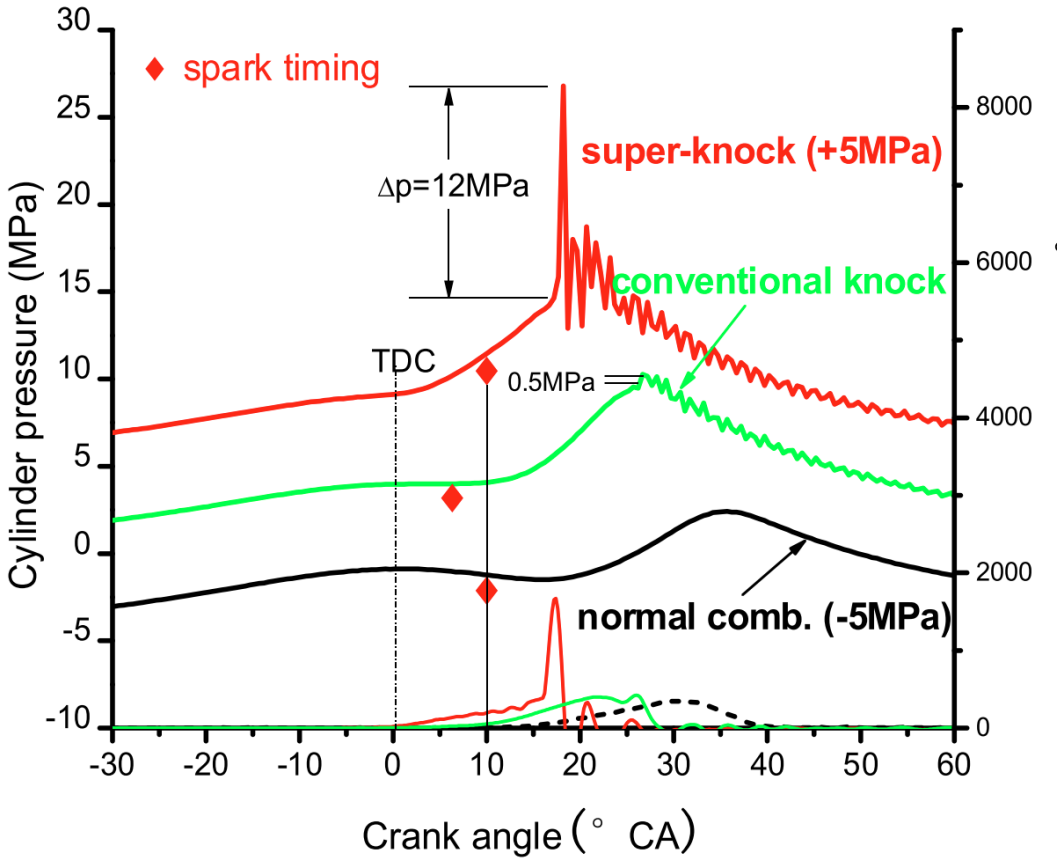
The materials supporting this study have been meticulously selected from various recognized academic research databases, including IEEE Xplore, Elsevier, and Springer. The search was also extended to include research libraries of leading universities worldwide using the Google Scholar search engine. The selection process initially prioritized publications of high relevance, impact, or number of citations within the field without any date restrictions. Subsequently, the most influential works published within the last three years were singled out, focusing on diversifying the research perspectives. Moreover, certain references were individually chosen owing to their affiliation with significant organizations or providing extensive and clear insights on the subject matter.

2.1 Overview of SI Engine Knock Phenomenon.

Engine knock is a term that originates from the distinctive sound produced by abnormal combustion within a SI engine. Figure 4 shows cylinders' typical combustion pressure traces. According to Towers and Hoekstra (TOWERS; HOEKSTRA, 1998), this abnormal combustion often occurs ahead of the normal flame front, leading to a pressure rise that is not only faster but also, at times, premature compared to the standard combustion cycle. Under certain circumstances, this can escalate to a high-intensity pressure spike.

The process behind this phenomenon is complex. As explained by Towers and Hoekstra (TOWERS; HOEKSTRA, 1998), when such an abrupt pressure increase occurs within the combustion chamber, it manifests as a pressure wave that travels across the chamber. On hitting the chamber wall, this wave is not entirely absorbed but is partly attenuated and then reflected across the combustion chamber as a slightly weaker wave.

Figure 4 – Typical combustion pressure traces.



Source: Zhi Wang, Liu, and Reitz (2017)

This reflection process repeats, causing the wave's energy to dissipate into the engine structure, thus causing vibrations or scatter throughout the gases within the chamber. As a result, the gases within the combustion chamber start vibrating at the resonant frequency of the chamber, which leads to the characteristic knocking sound (TOWERS; HOEKSTRA, 1998).

Moreover, the abrupt pressure increase and subsequent vibrations can erode vital engine components, including the piston, rings, and head gaskets. This potential for significant engine damage underscores the importance of understanding and effectively controlling engine knock (TOWERS; HOEKSTRA, 1998).

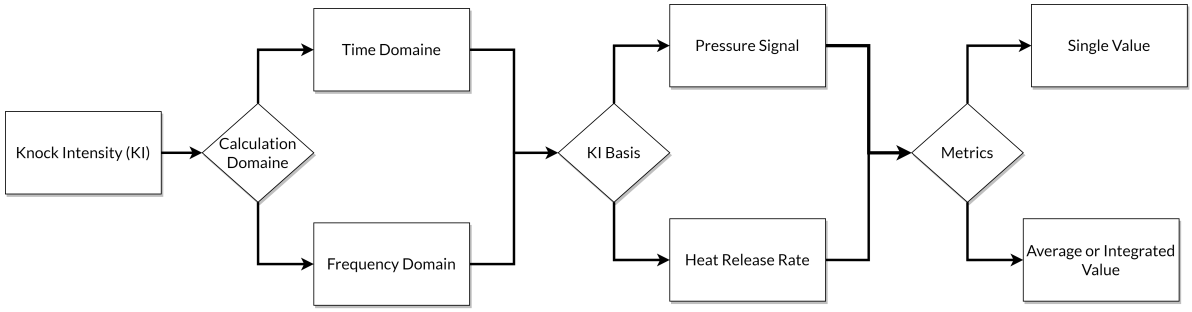
Knock intensity is typically characterized by the maximum peak-to-peak amplitude of the knock signal, which is derived by filtering the in-cylinder pressure signal to exclude high and low-frequency components (KALGHATGI, 2018).

Interestingly, even when engine operating conditions are carefully calibrated to avoid knock, extremely intense knock events, informally termed as "superknock" or

"megaknock", can occasionally occur in turbo-charged engines. According to Kalghatgi (KALGHATGI, 2018), superknock is believed to occur due to developing detonations, where the pressure wave ignited by autoignition is amplified by the autoignition reaction front.

The sporadic occurrence of super-knock presents a significant challenge in developing advanced gasoline engines. Despite pre-ignition being a prerequisite for super-knock, it does not invariably lead to this phenomenon. Pre-ignition can result in a range of outcomes, from super-knock and heavy knock to slight knock and non-knock, with the knock intensity increasing if the knock onset occurs near the Top Dead Center TDC. Rapid compression machines RCMs provide an appropriate platform to investigate pre-ignition and super-knock under engine-like conditions, using synchronous high-speed, high-resolution photography and pressure acquisition. RCM experiments have confirmed that super-knock is a result of detonation, with data indicating that the transition from pre-ignition to super-knock involves a hotspot-induced deflagration transitioning to hotspot-induced detonation, followed by high-pressure oscillation (WANG, Z.; LIU; REITZ, 2017).

Figure 5 – Possible signals to use to characterize the knock event.



Source: Adapted from Shahlari and Ghandhi (2012)

Knock intensity KI in engines can be evaluated through various metrics that can be broadly categorized based on the domain of the data used for the computation, the parameters involved, and the manner of calculation (SHAHLARI; GHANDHI, 2012). The knock intensity can be determined using time-domain or frequency-domain data, as depicted in Fig 5.

The parameters upon which knock intensity can be based include the pressure signal and its derivatives or the heat release rate and its derivatives. The latter depends on the pressure signal, among other parameters. The knock intensity metrics can be

calculated based on a single value like the maximum quantity value or an average or integrated value (SHAHLARI; GHANDHI, 2012).

All time-domain knock intensity metrics are derived from a filtered pressure trace, with high- or band-pass filtering carried out to eliminate the low-frequency content associated with compression and standard flame propagation heat release. The specifics of the filtering operation, such as filter type, order, and roll-off characteristics, significantly impact the results and hence need to be explicitly specified (SHAHLARI; GHANDHI, 2012).

A criterion for assessing knock intensity is proposed in some research (RICHARD *et al.*, 2009). It is based on the energy discharged via spontaneous auto-ignition of the residual fresh charge at the knock timing. This energy is normalized by the total energy discharged through flame propagation. This criterion, denoted by K_n, is expressed using the following equation:

$$K_n = K_1(1 - bmf.max(1, \Phi))(\Pi - 1)(1 - \frac{\theta_{knock}}{K_2}N_{eng})^{1/2} \tag{1}$$

Here, bmf represents the fuel burnt mass fraction, Π stands for the compression ratio, Φ_{knock} symbolizes the knock phasing in the expansion stroke, K_1 is a predefined parameter utilized to adjust the global knock intensity level, K_2 details the maximum crank angle at which knock remains audible, and N_{eng} refers to the engine speed.

Using this equation, four distinct categories can be formulated to classify the knock level:

- $K_n < 0.5$: No knock
- $0.5 \leq K_n < 1$: Trace knock
- $1 \leq K_n < 1.5$: Medium knock
- $1.5 \leq K_n$: Strong knock

Alternatively, other studies (ZHEN *et al.*, 2012) put forth a knock index based on the heat release rate and cumulative heat release.

The rate of heat release $ROHR$, which is determined by the in-cylinder pressure signal, is elucidated as follows:

$$ROHR = (\frac{\gamma}{1-\gamma})P\frac{dV}{d\theta} + (\frac{1}{1-\gamma})\frac{dP}{d\theta} \tag{2}$$

The Cumulative Heat Release, denoted by CHR_{NET}, is defined by:

$$CHR_{NET} = \int ((\frac{\gamma}{1-\gamma})P\frac{dV}{d\theta} + (\frac{1}{1-\gamma})V\frac{dP}{d\theta}) \tag{3}$$

In these equations, h refers to the crank angle degree, γ is the ratio of specific heats, P is the cylinder combustion pressure, and V is the cylinder volume.

An abrupt decline in the net cumulative heat release will likely correlate with a substantial increase in heat losses. The parameter CHR_{NET} strongly connects with knock intensity, while no relation exists with the sensor position.

Other commonly used indicators in the literature (ZHEN *et al.*, 2012; SHAHLARI; GHANDHI, 2012) are the Maximum Amplitude of Pressure Oscillations MAPO, the Integral Modulus of Pressure Gradient IMPG, and the Integral Modulus of Pressure Oscillations IMPO. MAPO is related to the peak of the pressure oscillations due to engine knock, IMPG is related to the modulus of pressure gradient, and IMPO represents the energy contained in the high-frequency oscillations of the cylinder pressure signal. The following equations give these indicators based on the high-frequency analysis of cylinder pressure data:

$$MAPO = max(\mid \hat{p} \mid_{\theta_0}^{\theta_{0+\zeta}}) \tag{4}$$

$$IMPG = \frac{1}{N}\sum_{1}^{N}\int_{\theta_0}^{\theta_{0+\zeta}} \mid \frac{d\hat{p}}{d\theta} \mid d\theta \tag{5}$$

$$IMPO = \frac{1}{N}\sum_{1}^{N}\int_{\theta_0}^{\theta_{0+\zeta}} \mid \hat{p} \mid d\theta \tag{6}$$

Where N represents the number of computed cycles, θ_0 is the crank angle corresponding to the beginning of the window of calculation, ζ is the value of the window of calculation, $\hat{p}$ is the filtered in-cylinder pressure.

2.2 Knock Detection and Control Fundamentals

The following section delves into knock detection and control fundamentals in Spark Ignition (SI) engines. It explores various knock detection methods, including in-cylinder pressure analysis, engine block vibration analysis, exhaust gas temperature monitoring, and chemical luminescence emissions analysis. Each method comes with

its own advantages and limitations, and the choice often depends on cost, accuracy, and practicality.

We will also discuss knock control strategies, which aim to balance the need for advanced ignition timing for optimal engine performance and fuel economy with the risk of knock occurrence. These strategies often involve complex algorithms and require careful calibration to account for various influencing factors such as engine speed, load, air/fuel ratio, and environmental conditions.

The section will further explore the knock intensity modeling in SI engines. Accurate modeling is key to the design and efficacy of knock control systems. Various modeling techniques will be discussed, ranging from parametric to machine learning and data-driven techniques.

This comprehensive exploration of knock detection and control fundamentals serves as a foundation for understanding the complexities of managing knock in SI engines and the ongoing research aimed at improving these processes.

2.2.1 Knock Detection

The effectiveness of a traditional controller predominantly depends on accurate knock detection as inaccuracies can compromise engine efficiency, with a non-detection potentially resulting in severe damage to the engine (BARES *et al.*, 2018).

Knock detection methods often involve analyzing in-cylinder pressure signals to study the combustion processes influenced by a knock directly. This method, typically used as a calibration reference for detection strategies based on sensors, identifies the high-frequency knock signature associated with the combustion chamber resonant frequencies excited by a rapid pressure rise. The resonant frequency of the vibration mode can be calculated using the equation (ZHEN *et al.*, 2012):

$$f_{m,n} = \frac{\alpha_{m,n} c_s}{\pi B} \tag{7}$$

where $\alpha_{m,n}$ is the wave number determined by Bessel's equations, m and n represent the numbers of radial and circumferential pressures nodes respectively, c_s is the speed of sound inside the combustion chamber (approximately 1000 m/s), and B is the cylinder bore.

Despite its widespread use in experimental research, this method has significant drawbacks. The required sensors are costly, and their direct contact with hot, high-pressure mixtures inside the cylinder chamber can reduce their lifespan and accuracy. Furthermore, while the sensor measurements can represent the whole combustion chamber pressure value under non-knocking conditions, knock introduces significant non-homogeneities, preventing the extrapolation of local data to the global domain. Additionally, the need for a sensor set for each cylinder increases costs and necessitates using various hardware and software (ZHEN *et al.*, 2012).

Other knock detection methods have been developed following the method based on in-cylinder pressure analysis. One such method is based on engine block vibration analysis. This approach determines the occurrence of knock by measuring the engine's vibration level, which increases when knock creates extensive pressure waves within the combustion chamber. These pressure waves can radiate sound within the audible frequency range, excite engine block vibrations, and produce an audible knock signal. Nonintrusive vibration sensors, with their excellent durability and low cost, have made this method the most practical for mass-production car engines (ZHEN *et al.*, 2012).

Another method involves monitoring exhaust gas temperature, as there is a clear correlation between engine knock and this temperature. In knocking combustion conditions, the exhaust gas temperature is reduced, making this technique efficient for knock detection. The signal is not affected by engine noise, making the detection pure, and the method is quick, convenient, and applicable to all types of engines.

Chemical luminescence emissions of end-gas can also be used to analyze the chemical reactions caused by end-gas autoignition. Spectroscopic and chemiluminescence measurements can identify CH, HCO, HCHO, and OH radicals as markers of different combustion phases, enabling knock detection based on intermediate radicals and species analysis (ZHEN *et al.*, 2012).

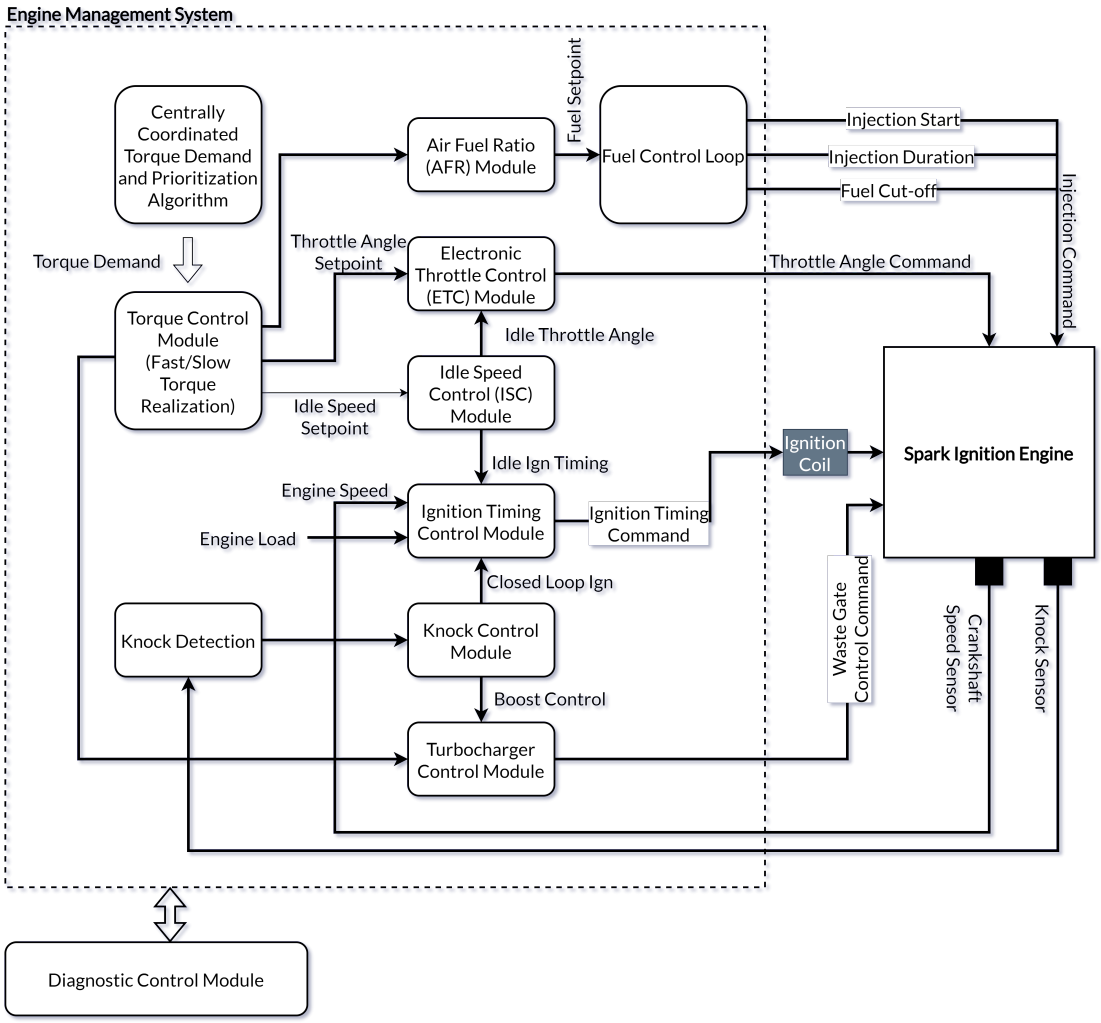
In the study conducted by Lagana et al., the authors explored ion current signal as a sensing device for identifying combustion and detonation in spark ignition (SI) engines, potentially replacing phase and knock sensors. The researchers found that the area under the ion current signal curve could be used to determine the cylinder under combustion. Furthermore, they established a correlation between the energy of a frequency band of the ion current signal and the energy of the knock sensor signal to

monitor detonation. The study concluded that the ion current could effectively detect combustion and detonation under various conditions, such as variations in the fuel mixture, spark plug mileage, and spark plug electrode area size. However, the authors noted the need for further experiments and exploration of engine conditions for practical applications, particularly in determining the optimal DC level voltage applied to the spark plug and identifying the boundaries within which ion currents could be used for different types of fuels and spark plug wear (LAGANA *et al.*, 2018).

Lastly, methods based on heat release analysis have been developed. As knock occurs, the heat transfer of the combustion chamber increases. When knock intensities exceed 0.2 Mpa, they influence the heat flux, and intensities above 0.6 Mpa can cause the peak heat flux up to 2.5 times higher than in non-knocking combustion. This method leverages that knocking combustion generates much higher wall heat fluxes than normal combustion and that temperature variations can be further enhanced when knock occurs (ZHEN *et al.*, 2012).

The ignition control module's advancement of the ignition timing within the torque structure is a significant factor in maximizing both the output torque and the engine's fuel economy, as shown in Fig. 6. However, optimal spark timing often coincides with the occurrence of knock in combustion, which can, under extreme conditions, damage the engine. This presents a necessary trade-off in engine control between maximizing spark advance and evading the onset of engine knock (ASHOK, B.; ASHOK, S. D.; KUMAR, 2016). Knock detection forms a crucial part of the knock control module, with various techniques utilized to detect knock in the combustion chamber. These techniques range from in-cylinder pressure transducers and engine block-mounted accelerometers to ionization signals from combustion. Despite being highly effective, in-cylinder pressure transducers are costly for production engines, prompting dedicated knock sensors - piezoelectric acceleration sensors fastened to the engine block - in production vehicles. Usually, one or two such sensors are used depending on the application's requirements. The primary role of these knock sensors is to extract knock-characteristic features that differentiate between normal engine noises and knock (ASHOK, B.; ASHOK, S. D.; KUMAR, 2016).

Figure 6 – Schematic representation of the EMS control system in a SI engine.



Source: Adapted from B Ashok, S Denis Ashok, and Kumar (2016)

.

2.2.2 Knock Control Strategies

The propensity of engine knock is influenced by numerous known and unknown factors such as compression ratio, spark advance, speed, load, air/fuel ratio, fuel quality, humidity, and other environmental conditions. Closed loop knock control, regulating spark advance based on knock intensity, is typically needed to account for these variables. Knock can cause audible annoyance at low speeds and potential engine damage at high speeds or loads. Therefore, most knock controllers aim to regulate the incidence or likelihood of audible or potentially damaging knock events. However, classifying knock intensities into knocking and non-knocking cycles and correlating them to engine damage, especially considering the variable dependencies on speed and load, poses challenges.

One suggested method is using a cylinder pressure heat release index to calibrate knock intensity values. A more common practice is to gradually advance the spark until the knock is barely audible or detectable, then determine a threshold that classifies a small percentage of cycles as responsible for the audible or "potentially damaging" knock events. This approach requires extensive calibration efforts across different engine speeds and load conditions. Therefore, strategies for simplified calibration or adaptive knock threshold have been proposed, such as scaling the threshold according to the mean non-knocking intensity, adjusting the threshold based on the variance of recently observed cycles, or using statistical hypothesis testing to adapt the threshold to maintain a constant knock probability or confidence value. (PEYTON JONES; SPELINA; FREY, 2014)

The conventional knock control approach gradually explores the knock limit by incrementally adjusting the spark angle by a factor of K_{adv} each engine cycle until a knock event transpires. Once the knock event is detected, the spark is quickly decreased by a significantly larger factor of K_{ret} to ensure the engine transitions back to a secure operating zone. Consequently, this control strategy can be mathematically represented as follows:

$$\theta(i) = \begin{cases} \theta(i-1) - K_{\text{ret}} & \text{if knocking} \\ \theta(i-1) + K_{\text{adv}} & \text{Then otherwise} \end{cases} \tag{8}$$

where $\theta(i)$ represents the spark advance at cycle i relative to the spark advance at borderline knock.

In more basic Spark Ignition SI engine control units , Spark Advance SA typically follows an open-loop control based on pre-calibrated lookup tables. This approach's drawback is its lack of a feedback signal, making knock onset susceptible to variations in engine operating conditions, fuel quality, and other factors. Therefore, a closed-loop knock control is usually necessary. Given its simplicity, vibration measurement is commonly employed in the industry as a closed-loop signal for knock detection. However, the quality of knock detection can be compromised due to noise and vibrations. Since knock and supplies dependable data directly influence cylinder pressure, in-cylinder pressure sensors are deemed more precise for knock detection and are predominantly used in lab tests. A widely used reference metric for knock detection based on in-cylinder

pressure is the Maximum Amplitude of Pressure Oscillations (MAPO), calculated by determining the maximum absolute value of the filtered pressure signal.

2.3 SI Engine Knock Intensity Modeling

The design and efficacy of knock control systems in Spark Ignition (SI) engines hinge on the accuracy and sophistication of the models used to represent the knock process. A diverse array of models have been proposed and utilized, each with its strengths and limitations. One of the advanced techniques employed is Computational Fluid Dynamics CFD, providing a deep physical understanding of the knocking combustion phenomena. However, CFD models are often computationally intensive, making them less feasible for real-time or on-board applications (PEYTON JONES; SHAYESTEHMANESH; FREY, 2020).

Another aspect to consider is the deterministic nature of many of these models. While this can simplify the modeling process, it limits their ability to accurately reproduce the statistical characteristics of cyclic knock intensity data observed in experimental conditions. This has led to a shift towards models derived from empirical data collected from physical engine tests, which can capture the inherent randomness in engine knock behavior.

In recent years, research has been increasing to advance the state of knock intensity modeling. Various approaches have been explored, from parametric modeling using dual log-normal models to machine learning and data-driven techniques. Each method brings a different perspective and set of tools to accurately modeling knock intensity.

This section aims to provide a comprehensive literature review of the various knock-intensity modeling techniques.

2.3.1 Related Works

In 2017, Trimby et al. developed a unified approach for reconstructing engine cylinder pressure using time-delay feedforward artificial neural networks, applying it to both crank kinematics and block vibration measurements (TRIMBY *et al.*, 2017). This innovative method was tested on data from a three-cylinder direct injected spark

ignition DISI engine. The authors identified several limitations with recurrent neural networks, which have been used in previous studies for cylinder pressure reconstruction, and they offered a solution using time-delay networks. The results obtained from the time-delay networks presented a significant improvement in accuracy over previous studies that utilized recurrent networks for pressure reconstruction. Furthermore, block vibration–based reconstruction results demonstrated that cylinder pressure can be successfully reconstructed under various engine conditions. This study presents a significant advancement in engine cylinder pressure reconstruction, showcasing the potential of time-delay neural networks in accurately modeling complex engine behaviors (TRIMBY *et al.*, 2017).

In 2020, Ricci et al. explored three machine learning approaches for predicting an SI engine's knock onset and intensity (RICCI *et al.*, 2020). The study leveraged various input parameters influencing the knock phenomenon, such as engine speed, air-fuel ratio, maximum internal cylinder pressure, combustion timing, and physical air conditions in the plenum, obtained from a CFD-1D engine model. This model was calibrated using experimental data, which were then used to train the machine learning models. The authors tested the trained models' ability to predict outputs based on samples not part of the training set. The predicted outputs were compared to the actual ones to evaluate the model's accuracy regarding Root Mean Square Error RMSE and coefficient of determination (R2).

The study further investigated the influence of engine parameters on the knock event by comparing different machine-learning approaches and tools, all based on the same dataset. The goal was to develop a robust real-time neural module capable of adjusting engine timing to prevent knock events. The high-performance SI engine used in the study was characterized by a displacement of about 2.5 L and was supercharged with a Gasoline Direct Injection (GDI) high-pressure system.

Ricci and his team focused on leveraging Machine Learning methods such as Back Propagation Artificial Neural Networks BPANNs and Random Forest RF algorithms, using MATLAB and Python to predict knock events and the corresponding knock index. The dataset was extracted from a CFD-1D model and calibrated using experimental data. The internal structures of the models were optimized to maximize predictive capability while minimizing computational effort. The study found that each tool could predict knock events accurately, with Python providing the highest number of well-predicted

events. However, as training samples increased, MATLAB showed superior performance regarding the percentage error rate (RICCI *et al.*, 2020).

In 2021, Jafari et al. explored the feasibility of reconstructing in-cylinder pressure using a structure-borne Acoustic Emission AE sensor, a more cost-efficient alternative to in-cylinder pressure transducers (JAFARI *et al.*, 2021). Their study leveraged the correlation between AE indicators and in-cylinder pressure parameters in both the time and crank angle domains. The reconstruction was performed in the crank angle domain using the Hilbert transform of AE to mitigate the impact of engine speed fluctuations. They utilized complex cepstrum signal processing and a feed-forward neural network to create a reconstruction regime. Their results indicated that the pressure could be reconstructed accurately using AE, regardless of engine load, speed, and fuel type. Their innovative approach combined cepstrum analysis and neural networks to reconstruct pressure, demonstrating good accuracy in estimating critical in-cylinder parameters such as Peak Pressure PP, Peak Pressure Timing PP, and Indicated Mean Effective Pressure IMEP. This study is a clear demonstration of the potential of low-cost AE sensors and data-driven techniques in reconstructing in-cylinder pressure, advancing our understanding and ability to monitor and control SI engine performance (JAFARI *et al.*, 2021).

In 2022, Kefalas et al. presented a novel approach for estimating combustion parameters, which is crucial for controlling emissions, fuel consumption, and efficiency in SI engines (KEFALAS *et al.*, 2022). Instead of relying on intrusive pressure sensors, which are expensive and uncertain in durability, the authors investigated the potential of a virtual sensor based on vibration signals acquired from a knock sensor (KS). The proposed method involved using Discrete Wavelet Transform DWT for signal preprocessing and extracting informative features, which would then be used to perform regression tasks with Extreme Gradient Boosting (XGBoost) models. This data-driven approach was applied to data from two different single-cylinder gas engines. It was able to successfully estimate combustion parameters, such as Peak Firing Pressure PFP, and the crank angle CA corresponding to 50% of Mass Fraction Burned (MFB50). The study concluded that the proposed approach can potentially replace expensive in-cylinder pressure sensors with a low-cost knock sensor without any additional machining steps required for mounting the sensors and without the calibration of filters. Future steps in this line of research will involve the comparison and collection of various methods for

in-cylinder pressure reconstruction and combustion parameters estimation using knock sensor signals and the implementation of the approach for a closed-loop control strategy (KEFALAS *et al.*, 2022). This work contributes significantly to the research on knock intensity modeling and offers potential for practical applications in SI engines.

2.4 Time Serie Forecasting using ANN

In this section, we will explore the application of ANNs, specifically focusing on Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), and Transformers for time series forecasting.

We will first introduce the concept of ANNs and their basic structure, followed by a detailed discussion on LSTM networks, a recurrent neural network capable of learning long-term dependencies. We will then delve into the application of CNNs, widely used for image and speech recognition tasks, for time series forecasting. Finally, we will discuss using Transformers, a novel type of ANN that has achieved state-of-the-art results in many machine learning tasks, for time series forecasting.

Through this section, we aim to provide a comprehensive understanding of how these advanced ANNs can be utilized for time series forecasting and how they overcome the limitations of traditional ANNs in handling sequence data.

2.4.1 Introduction to ANN Models

Artificial Neural Networks ANN are machine learning models inspired by the biological neural networks that constitute animal brains. The fundamental building block of an ANN is the artificial neuron or node, a simplified model of a biological neuron (DREW; MONSON, 2000).

In a biological neuron, dendrites receive signals from other neurons, and if the accumulated signal exceeds a certain threshold, the neuron fires, sending a signal down the axon to other neurons. Similarly, in an artificial neuron, the node receives inputs from various sources, applies a weight, and passes them through an activation function to produce an output (DREW; MONSON, 2000).

ANNs are composed of layers of these nodes. The first layer is the input layer, which receives the raw data, and the last is the output layer, which produces the final

output. In between, there may be one or more hidden layers, which transform the input into something the output layer can use (ZOU; HAN; SO, 2009).

The power of ANNs comes from their ability to learn the optimal weights for the inputs through training. During training, the network is exposed to a dataset and adjusts its weights based on the error of its predictions. This is typically done using a method called backpropagation, which involves calculating the gradient of the loss function concerning the weights and adjusting the weights in the direction that minimally decreases the loss (ZOU; HAN; SO, 2009).

ANNs have been used in various applications, from image and speech recognition to natural language processing and even in predicting complex phenomena. They are particularly good at handling non-linear relationships and high-dimensional data (ISLAM, 2022; XIAO; WANG, S.; PRUCKA, 2013; DONAHUE *et al.*, 2015).

However, traditional ANNs suffer from some limitations. For example, they assume that all inputs and outputs are independent, which is not the case in time-series or sequence data. This has led to the development of more advanced types of ANNs, such as Recurrent Neural Networks RNN, Convolutional Neural Networks (CNNs), and Transformers, designed to handle such data.

2.4.2 Time Serie Forecasting with Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks, a specialized form of Recurrent Neural Networks (RNN), have been introduced to overcome the limitations of traditional RNNs in sequence prediction tasks, such as time series forecasting. Unlike traditional ANNs and RNNs, LSTM networks can learn long-term dependencies in sequence data [Siami-Namini et al. 2018].

In a typical RNN, the hidden units are connected to form a directed cycle. This enables the network to use information from previous time steps to predict future time steps. However, RNNs have a significant limitation in that they suffer from the vanishing and exploding gradient problem, leading to difficulties in learning long-term dependencies in the data.

The mathematical representation of an RNN can be formulated as follows. Let's denote $i = (i_1, i_2, ..., i_T)$ as a sequence of length T, and h_t as the memory state of the RNN at time step t. An RNN updates its memory state based on (SIAMI-NAMINI; TAVAKOLI;

NAMIN, 2019):

$$h_t = \sigma(W_i i_t + W_h h_{t-1} + b_t) \tag{9}$$

In this equation, σ represents a non-linear activation function, typically a sigmoid or a hyperbolic tangent function. W_i and W_h are the weight matrices that play a crucial role in any deep learning model, influencing how much the current input i_t and previous memory state h_{t-1} affect the updated memory state h_t. Lastly, b_t is a bias term, a constant added to the product of the weights and their respective inputs to offset the result. This bias term b_t helps make the model more flexible to fit the data.

LSTM networks, on the other hand, introduce a new structure called a memory cell, capable of remembering a value for an arbitrary length of time. A memory cell consists of three main components: an input gate, a forget gate, and an output gate. The input gate determines how much of the new information should be stored in the cell, the forget gate determines how much of the current cell state should be forgotten, and the output gate determines how much of the current cell state should be output to the rest of the network.

The forget gate uses a sigmoid function to decide what information to retain in the LSTM memory based on the previous state h_{t-1} and current input x_t. This output f_t lies between 0 and 1, with 0 indicating complete removal of past learned values and 1 implying total preservation, calculated as:

$$f_t = \sigma(W_{f_h}[h_{t-1}], W_{f_x}[x_t], b_f) \tag{10}$$

The input gate decides if new information should be added to the LSTM memory. It has a sigmoid layer deciding which values to update and a $tanh$ layer creating candidate values for addition to the LSTM memory. Their outputs i_t and c_t are computed as:

$$i_t = \sigma(W_{i_h}[h_{t-1}], W_{i_x}[x_t], b_i) \tag{11}$$

$$c_t = \sigma(W_{c_h}[h_{t-1}], W_{c_x}[x_t], b_c) \tag{12}$$

The LSTM memory is updated by forgetting the current value using the forget gate and adding the new candidate value $i_t * c_t$ as:

$$c_t = f_t * c_{t-1} + i_t * c_t \tag{13}$$

The output gate uses a sigmoid layer to determine what part of the LSTM memory contributes to the output, then maps these values between -1 and 1 using a tanh function, and multiplies the result by the output of the sigmoid layer, as shown in:

$$o_t = \sigma(W_{o_h}[h_{t-1}], W_{o_x}[x_t], b_o) \tag{14}$$

$$h_t = o_t * tanh(c_t) \tag{15}$$

These gates allow LSTM networks to effectively learn and remember information over long sequences, making them especially suited for time series forecasting tasks. In such tasks, it is often crucial to consider recent observations and observations from the more distant past. LSTM networks can dynamically decide which observations to remember and which ones to forget, leading to better forecasting performance than traditional RNNs [Siami-Namini et al. 2018].

2.4.3 Time Serie Forecasting with Convolutional Neural Networks (CNN)

CNNs are another type of ANN that is particularly good at processing grid-like data, such as images. They comprise one or more convolutional layers, followed by one or more fully connected layers as in a standard multilayer neural network. This ANN architecture can also be leveraged to model temporal dependencies in time series data. By applying convolutional operations, 1D-CNNs can effectively extract local and global temporal patterns within the series, making them a potentially powerful tool for forecasting (LI, Z. *et al.*, 2021).

The paper by Borovykh et al (BOROVYKH; BOHTE; OOSTERLEE, n.d.) explains the operation of convolutional neural networks in time series forecasting. In CNNs, the input is convolved with a weight matrix (or filter) at each layer to produce a feature map. All output values in this map share the same weights, meaning they all detect the same pattern. This characteristic of CNNs reduces the number of learnable parameters, making training more efficient.

The input to each convolutional layer is typically three-dimensional (height, width,

and number of channels). The first layer is convolved with M1 three-dimensional filters to create the output feature map. For a one-dimensional input $x = (x_t)_{t=0}^{N-1}$ of size N with no zero padding, the output feature map from the first layer is computed by convolving each filter w_h^1 for $h = 1,...,M_1$ with the input:

$$a^1(i,h) = (w_h^1) * x)(i) = \sum_{j=-\infty}^{\infty} w_h^1(j)x(i-j) \tag{16}$$

where $w_h^1 \in \Re^{1xkx1}$ and $a^1 \in \Re^{1xN-k+1xM_1}$. This output is then passed through a non-linearity $f^1 = h(a^1)$.

In subsequent layers $l = 2,...,L$ the input feature map, $f^{l-1} \in \Re^{1xN_{l-1}xM_{l-1}}$ is convolved with a set of M_l filters $w_h^l \in \Re^{1xkxM_{l-1}}$, $h = 1,...,M_l$, to create a feature map $a^l \in \Re^{1xN_lxM_l}$:

$$a^l(i,h) = (w_h^l) * f^{l-1})(i) = \sum_{j=-\infty}^{\infty} \sum_{m=1}^{M_{l-1}} w_h^l(j,m) f^{l-1}(i-j,m) \tag{17}$$

This output then also goes through the non-linearity to give $f^1 = h(a^1)$.

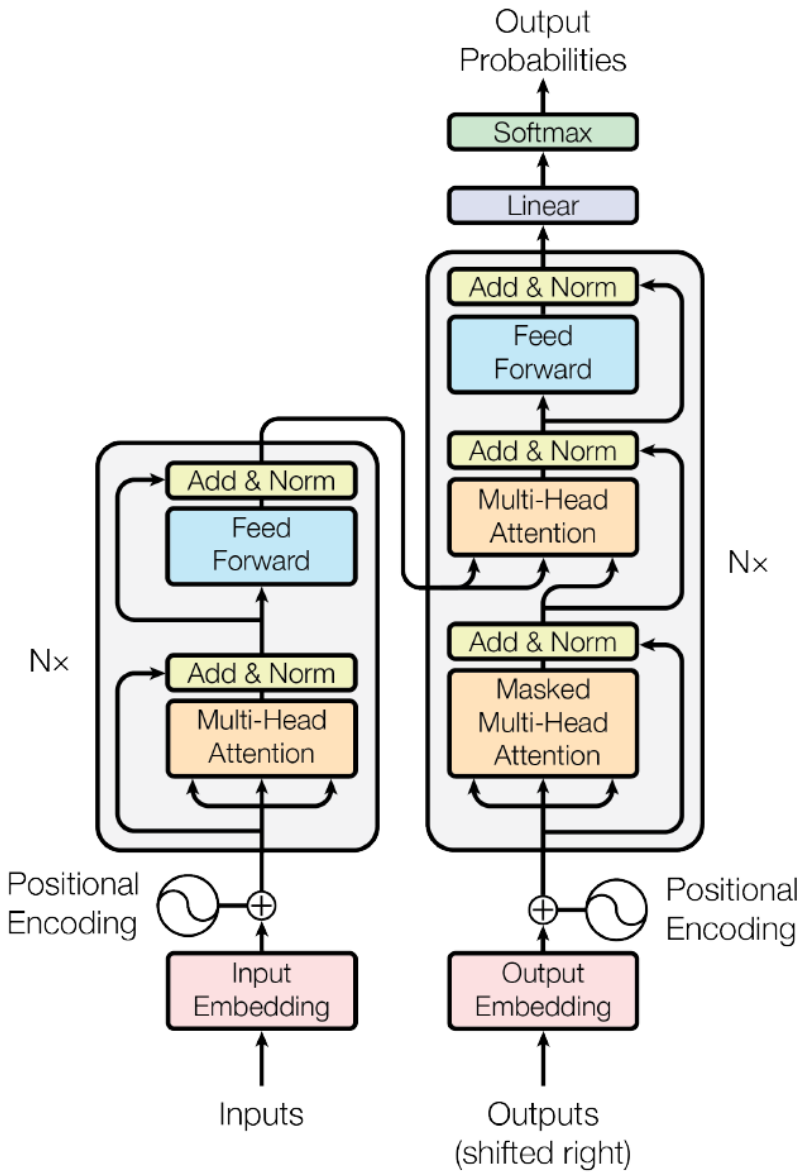
2.4.4 Transformers for Time Serie Forecasting

Transformers were first introduced for neural machine translation, a highly complex task within the field of Natural Language Processing (NLP) (VASWANI *et al.*, 2017). Recently, however, Transformers have been employed to address various challenges in the machine learning domain and have frequently achieved state-of-the-art results. Their applications extend beyond traditional NLP tasks to include image classification, object detection and segmentation, image, and language generation, sequential decision-making in reinforcement learning, multi-modal data processing (encompassing text, speech, and image), as well as the analysis of tabular and time-series data (AHMED *et al.*, 2023).

As introduced by Vaswani et al. in 2017, the original Transformer model adheres to the encoder-decoder structure common among many advanced neural sequence models. As depicted in Fig. 7, the encoder and decoder are built from identical units or blocks. Each block in the encoder is comprised of a multi-head self-attention module and a position-wise feed-forward network. Each block in the decoder is similarly composed

but with an additional cross-attention module inserted between the multi-head self-attention module and the position-wise feed-forward network.

Figure 7 – Transformer model architecture.



Source: Vaswani *et al.* (2017)

.

Unlike recurrent models like LSTMs or RNNs, Transformers do not have a recurrence but instead use positional encoding in the input embeddings to model sequence information (WEN *et al.*, 2022).

In the original Transformer model, absolute positional encoding is implemented. For each position index t, the encoding vector is given by:

$$PE(t)_i = \begin{cases} sin(w_i t) & i\%2 = 0 \\ cos(w_i t) & i\%2 = 1 \end{cases} \tag{18}$$

where w_i is the hand-crafted frequency for each dimension.

The Transformer employs a Query-Key-Value (QKV) model and utilizes the scaled dot-product attention mechanism:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{D_k}})V \tag{19}$$

where queries $Q \in \Re^{NxD_k}$, keys $K \in \Re^{MxD_k}$, values keys $V \in \Re^{MxD_v}$, N, M denote the lengths of queries and keys (or values), and D_k, D_v denote the dimensions of keys (or queries) and values.

It also applies multi-head attention with H different sets of learned projections:

$$MultiHeadAttn(Q,K,V) = Concat(head_1; ...; head_H)W^o \tag{20}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

In addition, the model includes a fully connected feed-forward network:

$$FFN(H') = ReLU(H'W^1 + b^1)W^2 + b^2 \tag{21}$$

where H' is outputs of previous layer, $W^1 \in \Re^{D_m xD_f}$, $W^2 \in \Re^{D_f xD_m}$, $b^1 \in \Re^{D_f}$, $b^2 \in \Re^{D_m}$ are trainable parameters.

In deeper modules, a residual connection module followed by a layer normalization module is inserted around each module:

$$H' = LayerNorm(SelfAttn(X) + X) \tag{22}$$

$$H = LayerNorm(FFN(H') + H') \tag{23}$$

In our study, we have adapted the Transformer architecture to our specific task of time-series forecasting by incorporating an Attention LSTM model. This approach, which has been used in many studies (LI, Y. *et al.*, 2019; KIM; KANG, 2019; ABBASIMEHR; PAKI, 2022), combines the strength of LSTM in capturing long-term dependencies and

the attention mechanism's ability to focus on relevant parts of the sequence.

Our model architecture, which will be describe in the next section, consists of an LSTM layer as the encoder, which returns the sequence and its state. An attention mechanism is then applied to the LSTM output, which is subsequently flattened. The softmax activation function is employed to compute the attention weights, which are repeated and permuted. The LSTM output and the attention weights are then multiplied, and the result is summed over time. The final output is obtained by passing through two dense layers.

The attention mechanism in our model allows it to focus on different parts of the input sequence when predicting a particular output, which can be especially useful when the input sequence is long and only certain parts of the sequence are relevant for the prediction. This is a significant advantage over traditional LSTM models, which do not have this capability.

3 MATERIAL AND METHODS

This chapter will delve into the methodologies employed to achieve our research objectives. It is broken down into significant sections covering aspects like data collection, data preprocessing, model development, training, and evaluation.

As this study involves data provided by an Original Equipment Manufacturer (OEM) under a confidentiality agreement, it is important to note that no confidential information has been included in this thesis. The data used has been carefully processed and anonymized to ensure the agreement's integrity. This has been done without compromising the research quality or the findings' validity. Therefore, the insights derived from this study can be considered reliable and applicable in a broader context while maintaining the confidentiality required by the OEM.

In the Data Collection section, we elaborate on the methods of compiling the data from a commercial 1.0l SCE 3-cylinder Engine, followed by a detailed overview of parameters recorded, testing conditions, and the volume of data collected. This foundational data will power the training of our proposed Artificial Neural Network (ANN) model.

The figures in this chapter illustrate the Knock Level and Noise signal distribution at different rotations per minute (rpm) for a full load and with the throttle partially closed. These distributions give us a statistical representation of our collected data under different engine conditions, allowing us to observe patterns, trends, and outliers.

Lastly, the correlation analysis tables outline the Spearman correlation coefficients between different parameters at various rpm during both full load and partially closed throttle conditions. This correlation analysis shows how tightly these parameters are related and can illustrate patterns that might have been missed in the graphical representations.

The Data Preprocessing section outlines the techniques used to clean, format, and partition the data to feed the ANN model. Here, the data is prepared to optimize the model's learning process.

In Model Development, we provide an in-depth description of the construction of our ANN model architecture, focusing on its structure, layers, nodes, and types of activation functions utilized.

The Model Training section lays down the methodologies for training our ANN

model, elaborating on the learning algorithms used, how the weight optimization was achieved, and the criteria employed to prevent overfitting.

The Model Evaluation section highlights the methods used to assess the performance of the trained model, focusing on the metrics used for performance evaluation and how the model's predictions are compared against actual data.

3.1 Data Collection

The data for the study was collected from a commercial 1.0l SCE 3-cylinder Engine, with the fuel used being low-octane gasoline.

We have collected parameters such as engine rotation (rpm), engine air load (wu), advance applied to each cylinder (°CRK), cylinder vibration signal (-), and peak detonation pressure inside each cylinder (bar). The measurements were executed on test benches and held at a constant engine rotation speed of 1700 rpm, 2700 rpm, 3600 rpm, and 4700 rpm, under operating conditions of full load (the throttle butterfly completely) as well as with the throttle partially closed.

In total, a vast database of 967201 data points was accumulated. This comprehensive dataset, capturing various operating conditions, constitutes the building block for training and validating the ANN model proposed in our research. The versatility of our dataset is instrumental in ensuring a robust model capable of efficiently predicting knock noises.

Let's move into a more detailed exploration in the Dataset Description section to offer more insight into the collected data. This section gives an in-depth look at the characteristics of the collected data, as recorded under varying engine rotations and operational conditions. We will also discuss the derived statistical parameters, such as the average, minimum, maximum, and standard deviation of peak detonation pressure and the cylinder vibration signals. Furthermore, we will investigate the data's distribution by observing trends and distinguishing outliers. Examining these attributes will provide a thorough understanding of the variables in our dataset and establish the context necessary for the subsequent modeling and analysis.

3.1.1 Dataset Description

Figure 8 illustrates the distribution of the peak detonation pressure inside each cylinder and the cylinder vibration signals, both recorded at an engine operation speed of 1700 rpm at Full load.

On average, the peak detonation pressure inside the cylinders sits around 0.18 bar. This pressure bottoms out at a minimum value of 0.03 bar and maxes out at values that range from 0.83 to 1.74 bar. These values suggest that the engine operation mostly resides within a lower to medium detonation pressure range, with a few instances where high-pressure spikes indicate severe knocking.

The cylinder vibration signals present a higher degree of variability. The mean average is between 15.00 to 18.96, with a standard deviation ranging from 7.59 to 14.26. The extreme ends of recorded vibrations range from a lower end of 4.00 to a significantly higher value of 197.00. This suggests that while the engine usually operates quietly, there are instances when a pronounced knocking noise is detected.

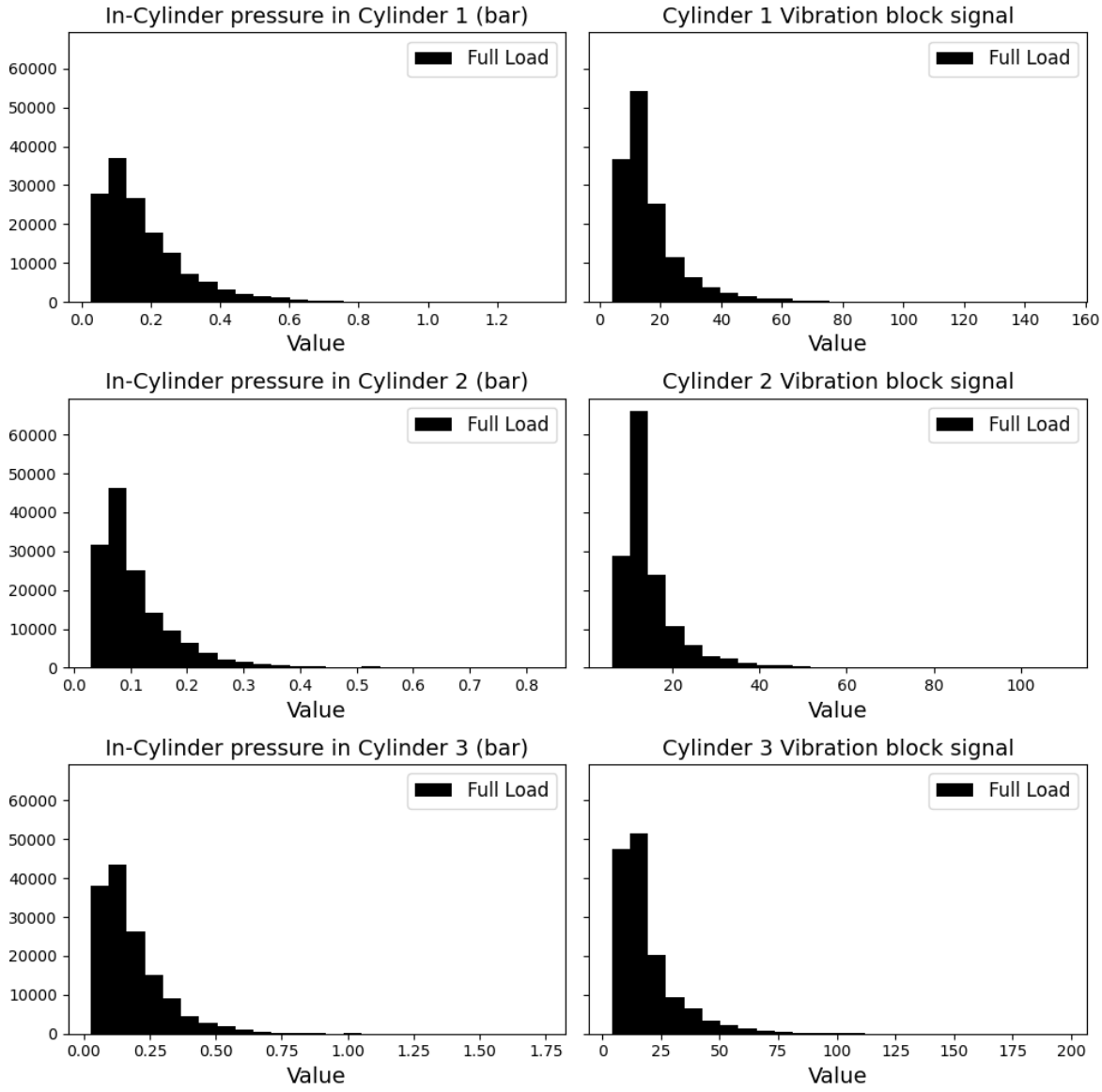
The histogram encapsulates a skewed distribution for both measurements with a few high outliers, suggesting a non-normal data distribution.

Figure 9 presents distributions of the peak detonation pressure inside each cylinder and the vibration signals at an engine speed of 2700 rpm. The data has been collected under two distinct operational loads: with the throttle partially closed and at full load.

When the throttle is partially closed at 2700 rpm, the peak detonation pressure inside the cylinders has a lower mean value of around 0.08 bar. It reaches minimum values of 0.03 bar while the maximum pressure varies from 0.62 to 1.74 bar, signifying light to moderately strong knocking instances. The cylinder vibration signals average from approximately 16.63 to 21.34, representing the noise generated from knocking. They showcase some variability, with standard deviations ranging from 4.93 to 7.37.

Contrastingly, at 2700 rpm with the throttle at full load, the peak detonation pressures have a slightly lower mean value of around 0.08 bar yet showcase higher variability, with standard deviations of approximately 0.05 to 0.07 bar. The maximum pressure values range from 1.03 to 1.60 bar, indicating occasional intense knocking under full load. Cylinder vibration signals present similar mean values, reflecting increased knocking and greater noise.

Figure 8 – Knock Level and Noises signal distribution at 1700 rpm.



Source: Own authorship (2023)

These observations show that the engine knock behavior exhibits differences when operated at different throttle positions - partially closed or at full load. While the mean values seem comparable, variations and extremes showcase that throttle position can significantly impact the intensity of knocking and its manifestation in noise.

In both operational conditions, the distributions of peak detonation pressures exhibit right skewness or positive skewness, meaning that the tail on the right side of the distribution is longer or fatter. The data extends more to the right. This skewness is indicated by the mean being greater than the median (50% percentile). For instance, under the partial throttle condition, the mean value is 0.08 bar, while the median is slightly lower at 0.07 or 0.08 bar.

The peak detonation pressure distributions indicate that most of the detonation pressure values are relatively low, with a few exceptions of higher pressures. These high-value exceptions cause the distribution to skew to the right.

The cylinder vibration signal distributions seem closer to a normal distribution, as indicated by nearer mean and median values. For example, under throttle partially closed operations, the mean vibration signal strength lies between 16.63 and 21.34, and the median ranges across 16.00 and 21.00, indicating a less skewed distribution.

However, high outliers (for instance, with a maximum of up to 153.00 or 166.00 depending on load conditions) extend the tail on the right side of these distributions, suggesting slight right skewness in this scenario as well.

In summary, both peak detonation pressures and vibration signal distributions seem to have positive skew (right-skewed) in their representations, although the latter depicts a histogram closer to normality

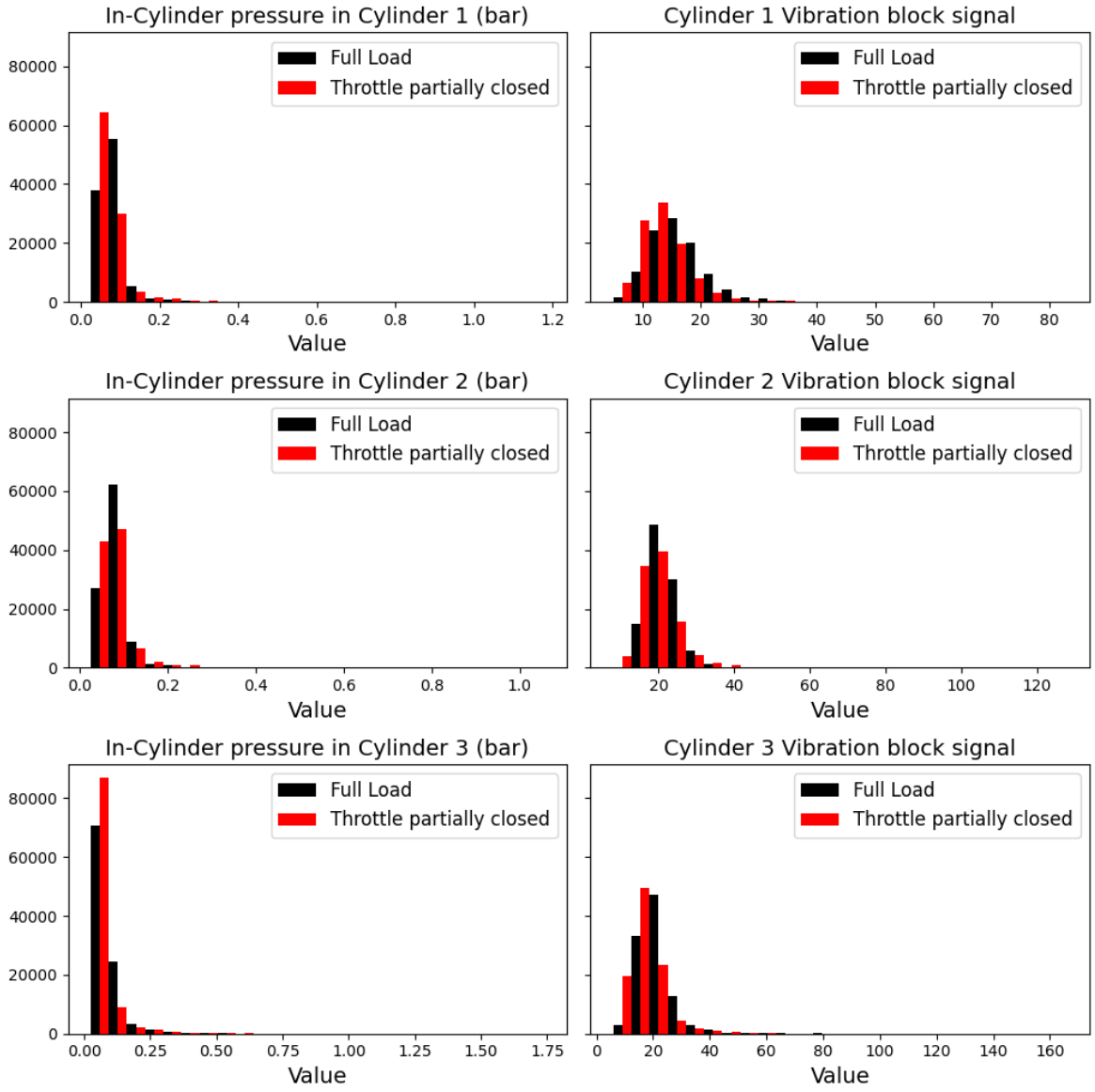
Figure 10 displays the distributions of the peak detonation pressure inside each cylinder and the cylinder vibration signals gathered at an operational speed of 3600 rpm. The data is presented under two unique operational conditions: when the throttle is partially closed and when the throttle is fully open (i.e., full load).

When the throttle is partially closed, the distribution of peak detonation pressure inside the cylinders shows a somewhat symmetric behavior, skewed slightly right, indicated by the slightly higher mean (0.09 to 0.10 bar) compared to the median (0.08 to 0.09 bar). The values mostly lie in the lower range, with a few high detonation pressures (max values ranging from 1.04 to 1.80 bar), suggesting sporadic intense knocking events. The cylinder vibration signals at this operational condition depict a mean ranging from 18.12 to 21.20. This distribution also leans slightly right, but the mean and median are almost similar, signifying a near-normal distribution.

During the full load at 3600 rpm, the peak detonation pressures depict a slightly higher mean of 0.10 bar, with a slightly lower median. This shows a positive skew (right skew); the maximum values lie between 1.77 and 2.28 bar. The vibration signals appear to be more normally distributed, with the mean and median very close. However, the maximum values (reaching up to 224.00), reveal occasional severe knocking instances resulting in high noise.

Both operational conditions present fairly normal distributions but with a subtle positive skew due to an extended right tail caused by rare, high-value outliers.

Figure 9 – Knock Level and Noises signal distribution at 2700 rpm.

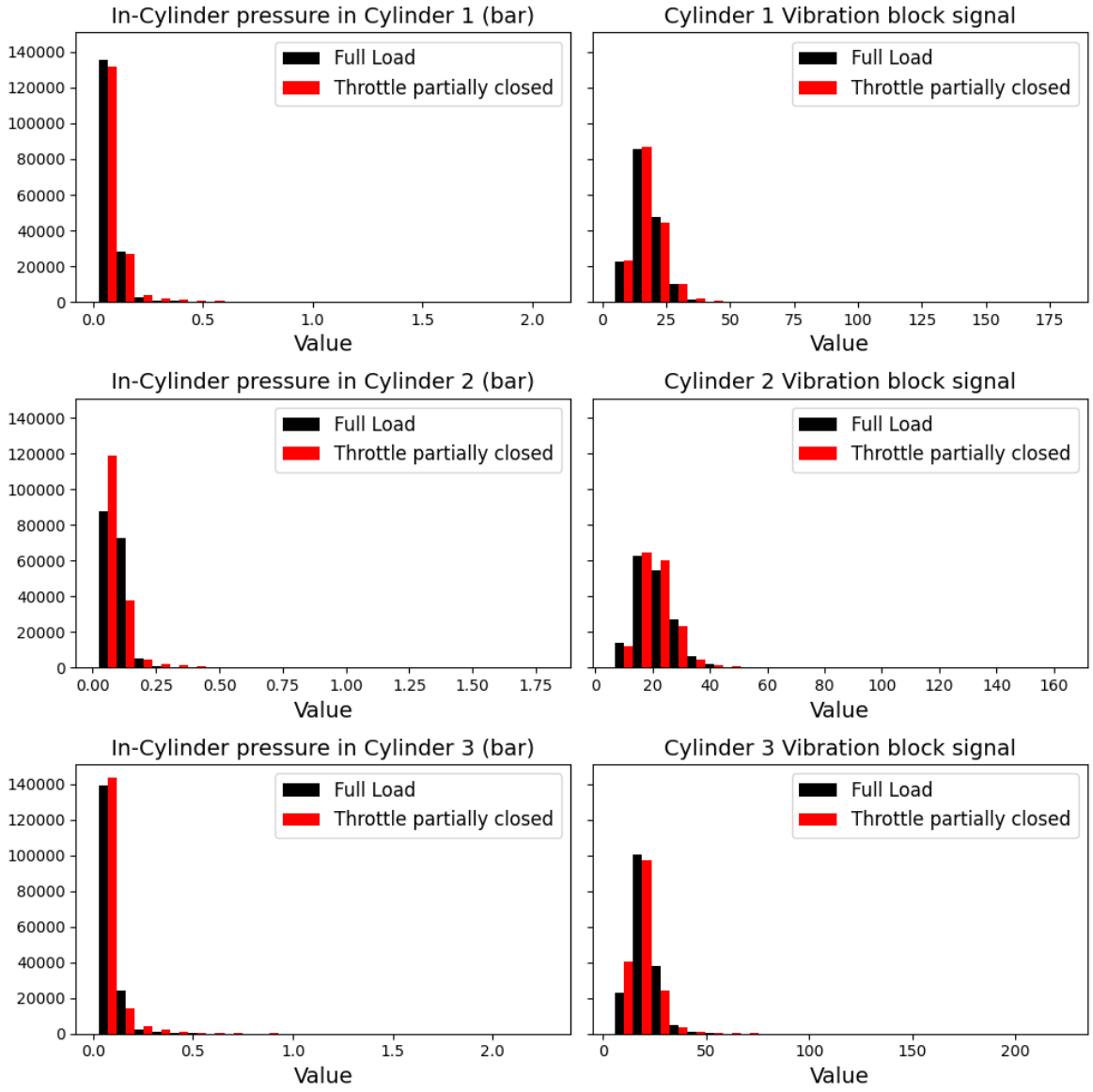


Source: Own authorship (2023)

Figure 11 illustrates the distributions of peak detonation pressure inside each cylinder and the vibration signals at an engine rotational speed of 4700 rpm. The data is analyzed under two different operational loads: with the throttle partially closed and at full load.

When the throttle is partially closed at 4700 rpm, the mean peak detonation pressure ranges from 0.12 to 0.14 bar, while the median settles around 0.11 bar, indicating a right skew or positive skew. This suggests that while detonation pressures tend to be relatively lower, there are higher pressures (evidenced by maximum values ranging from 1.5 to 2.87 bar). The cylinder vibration signals under this condition depict a mean value ranging between 20.12 to 24.92. The difference between the mean and median

Figure 10 – Knock Level and Noises signal distribution at 3600 rpm.



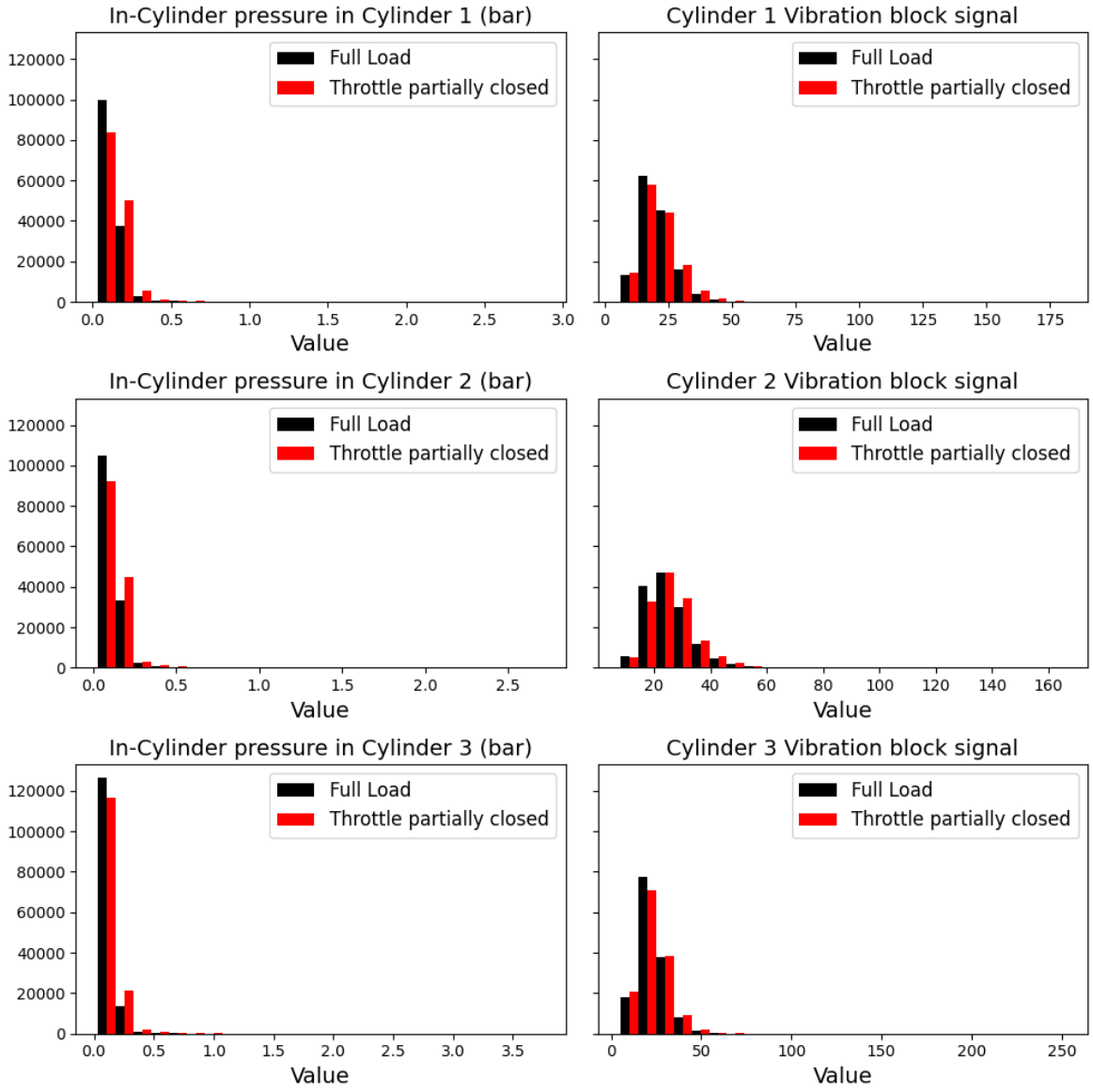
Source: Own authorship (2023)

suggests a slight positive skew.

The mean detonation pressures grow slightly at full load from 0.14 to 0.16 bar. This distribution exhibits a right skew, with maximum pressure values surging up to 2.40 to 3.75 bar, suggesting intense knocking when operating at full load. The vibration signals in this scenario also have a slight positive skew, with a mean value that ranges from 20.74 to 26.12.

To summarize, both operational conditions at 4700 rpm present distributions with a slight positive skew due to an elongated right tail caused by sporadic instances of high detonation pressures and subsequent noise.

Figure 11 – Knock Level and Noises signal distribution at 4700 rpm.



Source: Own authorship (2023)

3.1.2 Correlation Analysis

In this subsection, we perform a correlation analysis to understand the relationship between different engine parameters under varying operating conditions. Correlation is a statistical measure indicating the extent to which two or more variables fluctuate. A positive correlation indicates the extent to which those variables increase or decrease in parallel, while a negative correlation indicates the extent to which one variable increases as the other decreases.

In the context of the following tables, "Iga" stands for ignition advance, the measure of the timing of the spark in the combustion process in relation to the piston's

position. Each number after "Iga" indicates the cylinder from which the data comes from, for example, "Iga1" would indicate ignition advance for the first cylinder. "KP" represents the peak detonation pressure, which is the maximum pressure reached during the detonation process within the cylinder. Like "Iga", the numbers following "KP" refer to the corresponding cylinder. Lastly, "Noise" symbolizes the cylinder block vibration, which is the vibration caused by the engine's operation, measured at each cylinder block. As with both previous variables, the numerical suffix corresponds to the respective cylinder from which the data was collected.

We use Spearman's correlation coefficient, a non-parametric measure of correlation that assesses how well the relationship between two variables can be described using a monotonic function. We chose Spearman's over Pearson's correlation because it can capture both linear and monotonic relationships and is less sensitive to outliers (DE WINTER; GOSLING; POTTER, 2016).

The Spearman's correlation coefficient is given by:

$$r_s = \frac{\sum_{i=1}^{n}(r_i - \bar{r})(s_i - \bar{s})}{\sqrt{(\sum_{i=1}^{n}(r_i - \bar{r})^2)(\sum_{i=1}^{n}(s_i - \bar{s})^2)}} \tag{24}$$

where:

r_s is the Spearman's correlation coefficient.

n is the number of observations.

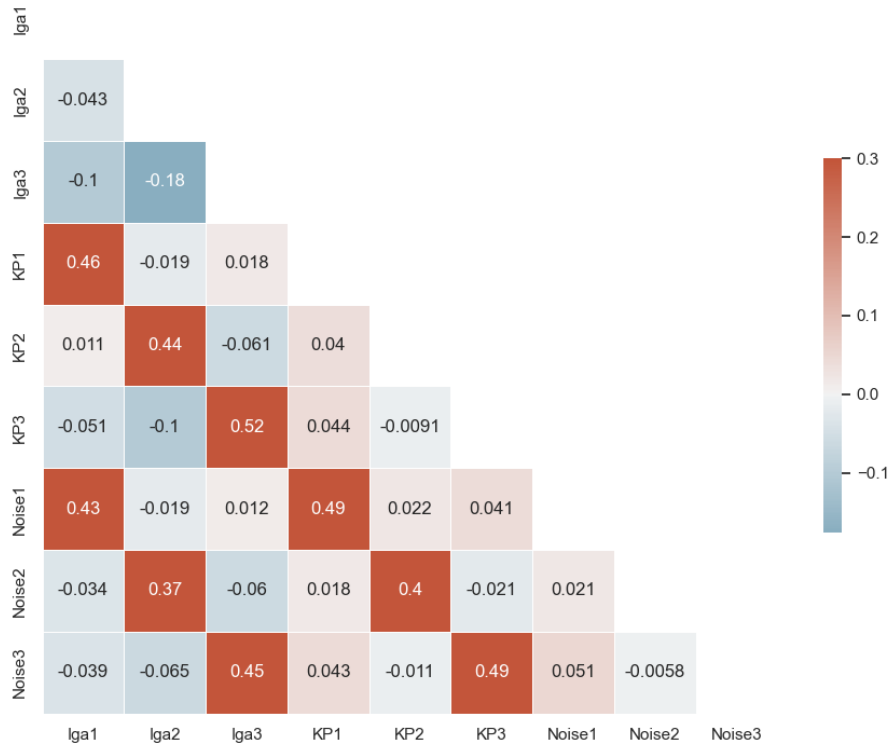
r_i and s_i are the ranks of x_i and y_i respectively.

$\bar{r}$ and $\bar{s}$ are the average ranks of x_i and y_i respectively.

The Figures 12 to 18 presented here show the correlation coefficients between the variables under different engine speeds, either in full load or with the throttle partially closed. Each cell in the table presents the correlation between two variables, and the values range between -1 and 1, inclusive. A value of 1 indicates perfect positive correlation, -1 indicates perfect negative correlation, and 0 suggests no correlation.

At 1700 rpm in full load, the highest positive correlation observed is 0.52 between Iga3 and KP3. This suggests that as the advance applied to the third cylinder increases, the peak detonation pressure inside the third cylinder also increases. Furthermore, a high positive correlation of 0.49 is noticed between KP1 and Noise1, inferring a direct relationship between the peak detonation pressure inside the first cylinder and its vibration signal.

Figure 12 – Spearman's Correlation Analysis at 1700 rpm in Full Load.



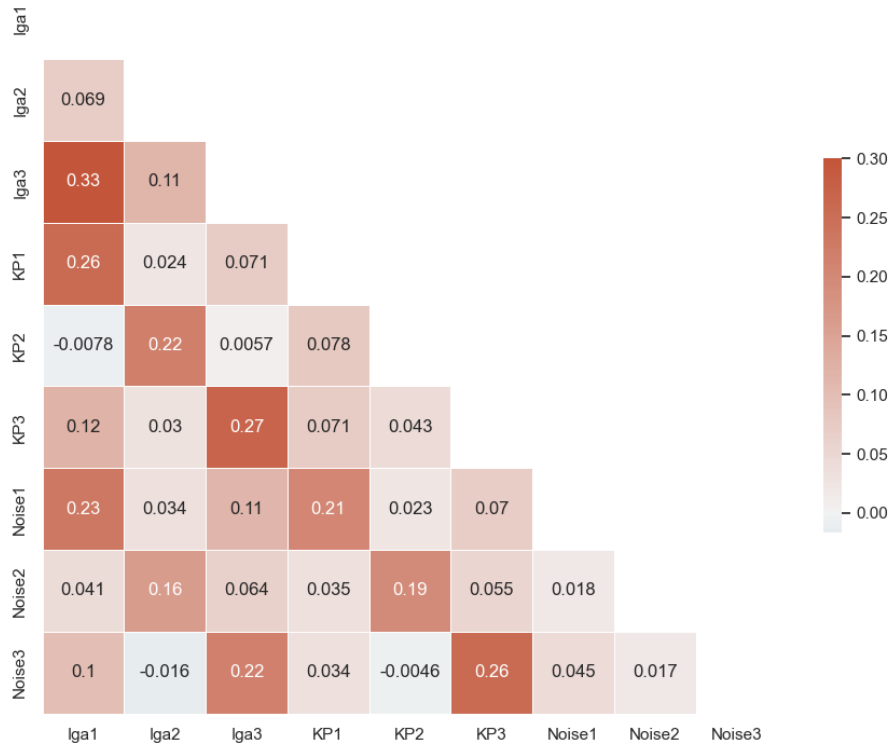
Source: Own authorship (2023)

At 2700 rpm in Full Load, a moderate positive correlation is observed between Iga1 and Iga3, suggesting that the advance applied to the third cylinder increases with the increase in advance applied to the first cylinder. Another noteworthy correlation between KP3 and Noise3 indicates a direct proportional relationship between peak pressure and vibration for the third cylinder.

At 3600 rpm in Full Load, correlations tend to decline, suggesting a less straight-forward relationship between variables. However, there's a relevant positive correlation between Iga1 and Iga2, and between Iga3 and KP3 at this speed. This suggests that increasing the advance applied to the first cylinder is likely to increase the advance applied to the second cylinder, and an increase in peak detonation pressure inside cylinder 3 is likely to result in higher vibration.

At 4700 rpm in Full Load, most correlation coefficients are weak, demonstrating that the relationship between variables is far less linear or monotonic at these higher

Figure 13 – Spearman's Correlation Analysis at 2700 rpm in Full Load.



Source: Own authorship (2023)

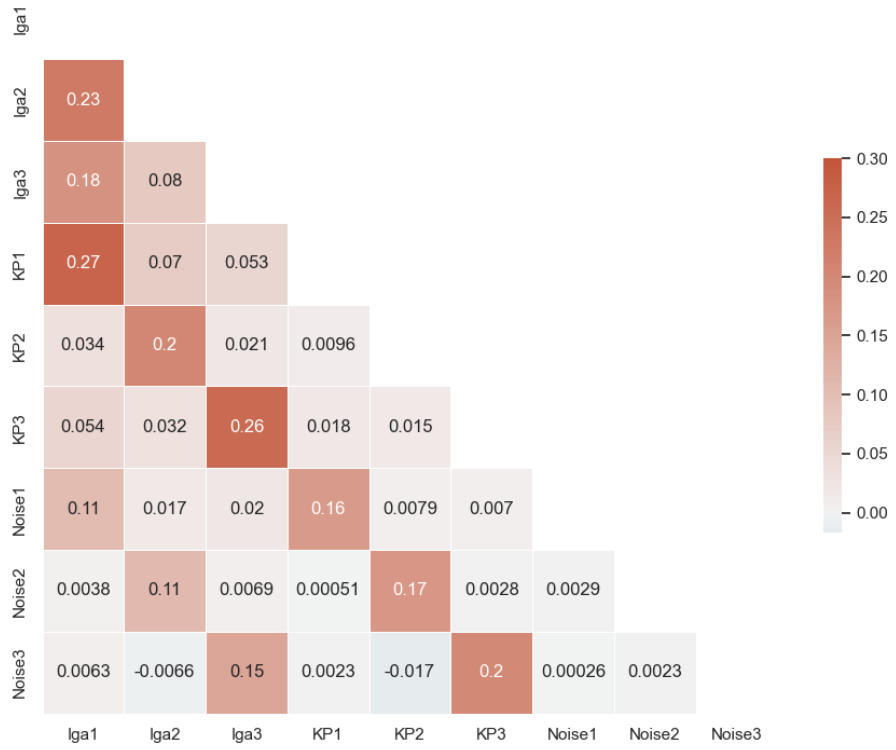
speeds. However, a moderate correlation between lga3 and KP1 suggests a relative increase in the advance applied to the third cylinder as the peak detonation pressure in the first cylinder increases.

At 2700 rpm with Throttle Partially Closed, a moderate positive correlation exists between lga2 and lga3 and a more significant one between lga3 and KP3. This indicates that with an increase in the advance applied to the second cylinder, the advance applied to the third correspondingly increases, as does the peak detonation pressure in the third cylinder.

At 3600 rpm with Throttle Partially Closed, a noticeable correlation exists between lga1 and lga3, and between lga3 and KP3. This suggests that with an increase in the advance applied to the first cylinder, the advance applied to the third cylinder also increases, as does the peak detonation pressure in the third cylinder.

At 4700 rpm with Throttle Partially Closed, there's a weak correlation between

Figure 14 – Spearman's Correlation Analysis at 3600 rpm in Full Load.



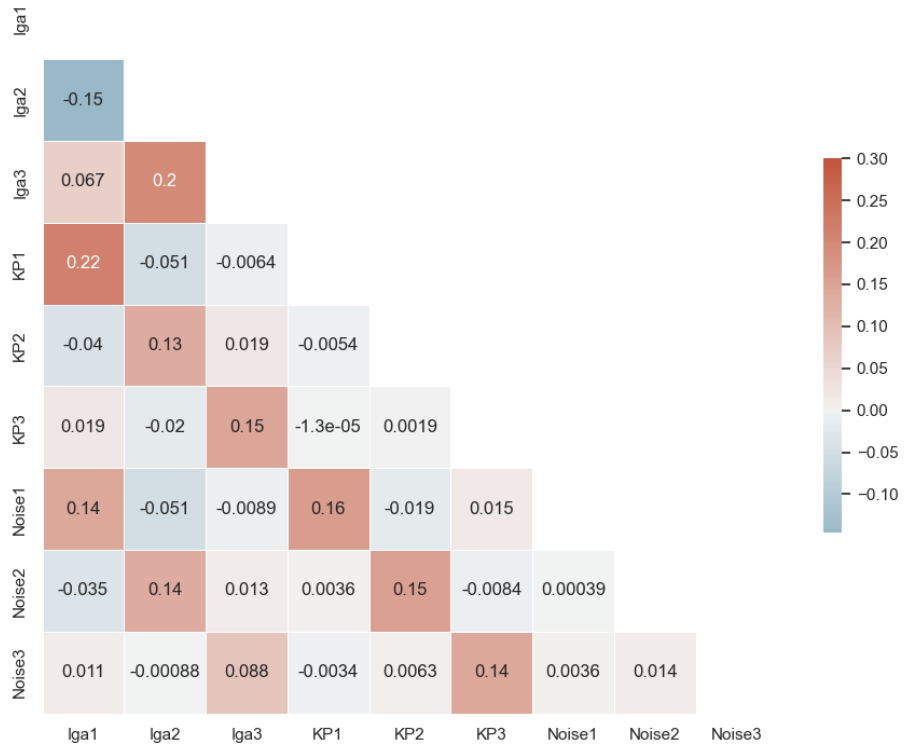
Source: Own authorship (2023)

most variables, signaling that the relationships between variables are far less obvious with the throttle partially closed at such high speeds. However, Iga3 and KP3 maintain a moderate positive correlation, similar to previous states. Thus, the peak detonation pressure in the third cylinder is expected to increase with the advance applied.

3.2 Models Architectures

In this section, we will delve into the development process of distinct types of deep learning architectures, namely, a 1-Dimensional Convolutional Neural Network (1-D CNN), a Bidirectional Long Short-Term Memory (Bi-LSTM) model, and an Attention LSTM model. These architectures have been chosen due to their prowess in handling and making predictions on time-series and sequence-based data, which perfectly align with our research problem of engine cylinder vibration prediction.

Figure 15 – Spearman's Correlation Analysis at 4700 rpm in Full Load.



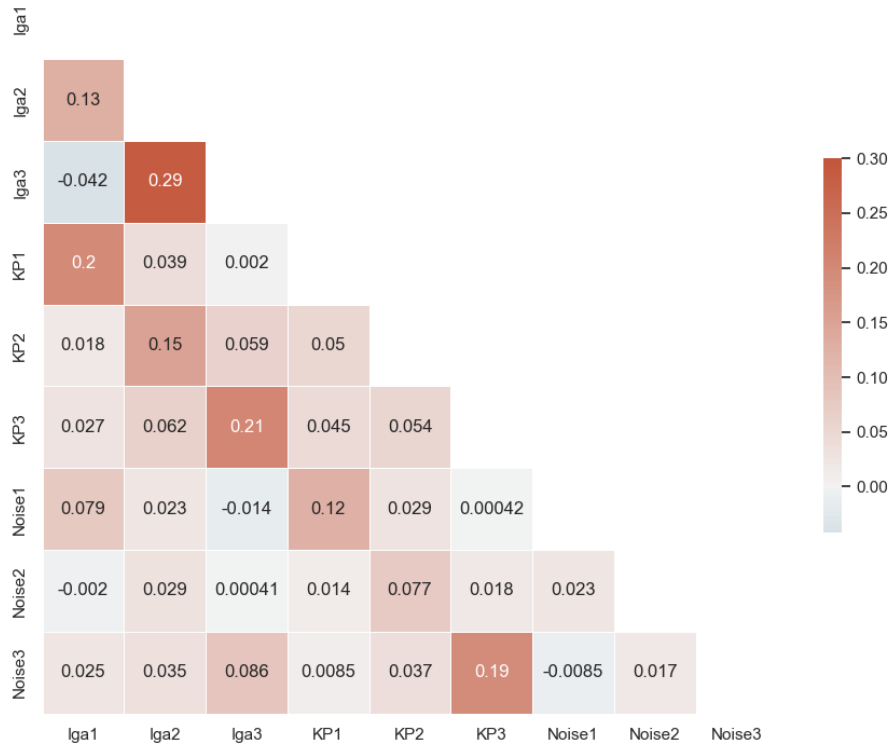
Source: Own authorship (2023)

The development process will encompass the design, training, validation, optimization, and evaluation of each individual model. Each model will be fine-tuned and evaluated on its predictive performance, focusing on how accurately it can predict cylinder vibration signals based on inputs of engine rotation, load, advance applied to each cylinder, and peak detonation pressure inside each cylinder.

The subsequent subsections will provide technical insights into the development and functioning of each model, elucidating how each model utilizes and learns from the data to produce accurate results.

The first subsection would deal with developing the 1-D CNN. 1-D CNNs are extensively used in sequence data handling due to their proficiency in learning spatial hierarchies from the input. They use filters/Kernels that scan through the sequence data and aggregate information to form a hierarchy of spatial features. The application of such models suits our scenario well, where spatial relationships in the form of sequence

Figure 16 – Spearman's Correlation Analysis at 2700 rpm with Throttle partially closed.



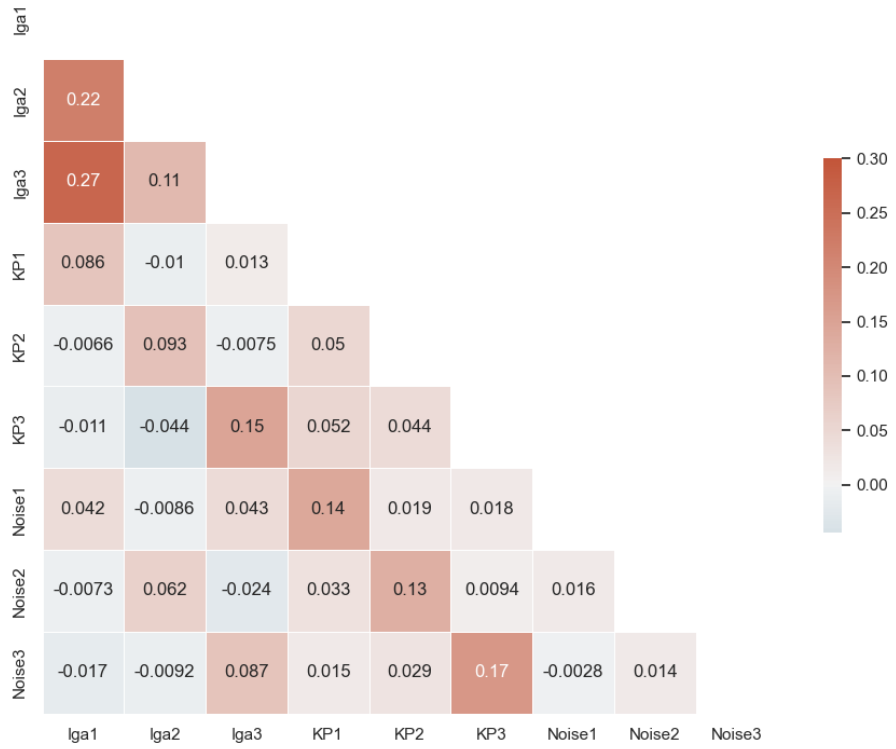
Source: Own authorship (2023)

lag values can play a vital role in determining cylinder vibrations.

The second and third subsections will present the development of two types of Recurrent Neural Networks (RNN) models - the Bidirectional Long Short-Term Memory (Bi-LSTM) model and the Attention LSTM model. As the engine's functioning is cyclical and temporal, the usage of bi-directionality and attention mechanism in LSTM-based models provides profound comprehension of the temporally spaced data. These RNN architectures preserve information from past data points and leverage future data points, granting them higher accuracy in capturing long-term dependencies and complexities in the data.

In summary, this model development section presents an in-depth exploration of various deep learning strategies and their tailored development catering to our problem of predicting engine cylinder vibrations based on past sequences of operations data. Careful observations from the development of these models will contribute significant

Figure 17 – Spearman's Correlation Analysis at 3600 rpm with Throttle partially closed.



Source: Own authorship (2023)

insights into the best-suited model for this data. The entire model was implemented using the Tensorflow machine learning framework.

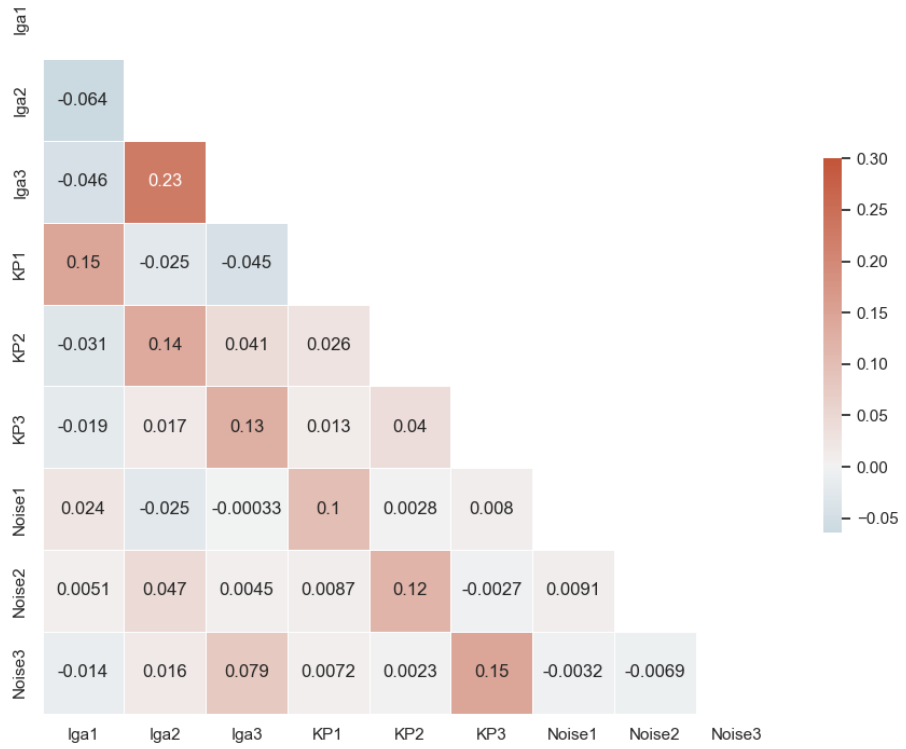
3.2.1 1-D CNN Model Development

The architecture of the 1-D CNN model has been presented in Figure 19. The model starts with a 1-dimensional Convolution layer, followed by a 1-dimensional Max Pooling layer, a Flatten layer, a Dense layer, and another Dense layer. Each layer has a specific function, and their combined interaction generates the model's output.

Before moving on, however, we need to establish some conventions. Let's denote:

- X as the input data tensor, having dimensions of (N, T, D), where N is the number of samples, T is the number sequence length, and D is the number

Figure 18 – Spearman's Correlation Analysis at 4700 rpm with Throttle partially closed.



Source: Own authorship (2023)

of features per sequence element.

- y as the output tensor, having the dimensions $(N,1)$.

1-Dimensional Convolution Layer: The first layer in the architecture. We utilize this layer to detect local features in the sequence data. During its operation, a kernel of a specified size convolves over the input data, producing multiple feature maps as output. The activation function ReLU (Rectified Linear Units) keep the nonlinearity intact in the network, which allows the model to learn complex patterns.

Let W_f and b_f denote the filter weights and bias, respectively. Then, the output, $y_{i,j}$, for i^{th} sample and j^{th} filter, is calculated as:

$$y_{ij} = \text{ReLU}\left(\sum_{k=1}^{D}\sum_{p=0}^{K-1} W_{j,k,p} \cdot x_{i,p+q,k} + b_j\right) \tag{25}$$

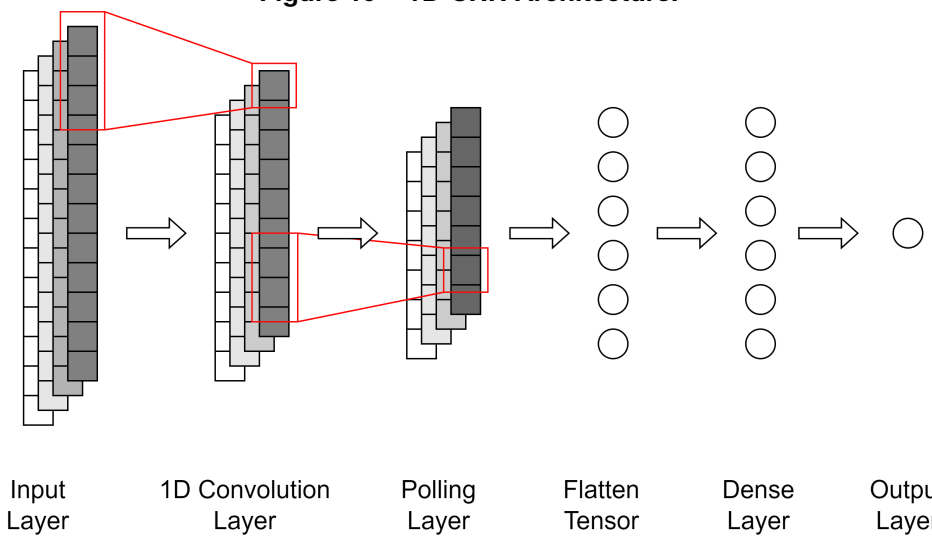
where K is the kernel size and q is the current position of the convolution.

Pooling Layer: The layer used here is a One Dimension Max Pooling with a max pooling window size equal to 2. It is used to down-sample the feature maps generated by the One Dimension Convolution Layer layer by taking the maximum value over a specified window size. This operation helps to reduce the spatial size, computationally making the model more manageable and providing a form of translational invariance.

Flatten Layer: It is a utility layer used to flatten the output of the One Dimension Max Pooling layer from a 2D tensor into a 1D tensor.

Dense Layers: These are fully connected layers with 50 nodes, designed to further process the features extracted by the convolution and pooling layers. Our network contains two such layers. The first dense layer with ReLU activation function comprises 50 units. Following that, we have another dense layer with a linear activation and 1 unit that produces the final prediction of the model.

Figure 19 – 1D-CNN Architecture.



Source: Own authorship (2023)

In the next steps of the process, this 1-D CNN model will be trained and validated with the engine operational data. This training will aid the model in learning effective feature representations and mappings between the input sequence and the target engine cylinder vibration signals.

3.2.2 Bidirectional LSTM Model Development

The architecture of the Bidirectional LSTM (Bi-LSTM) model is outlined in Figure 21. The model starts with an input layer, followed by a bidirectional LSTM layer, an

LSTM layer, a GlobalAveragePooling1D layer, and finally, a Dense layer. Each of this model's layers serves a unique purpose and constructs the model's output together.

Let's define:

- X as the input data tensor, having dimensions of (N, T, D), where N is the number of samples, T is the number sequence length, and D is the number of features per sequence element.
- y as the output tensor, having the dimensions $(N,1)$.

Input Layer: The model receives its sequence input in this layer. The shape of the input matches the dimensions of the sequence data.

Bidirectional LSTM Layer: This is the foundational component in our LSTM-based architecture. The significant feature of Bi-LSTM is that its LSTM layer runs in both forward and backward directions. The forward LSTM layer analyses the sequence from the past to the future time-step, and the backward LSTM layer runs from the future to the past time-step, thus providing a broader context. Each of the units in the LSTM layer can maintain its hidden state. A dropout of 0.2 has been implemented to the input sequence for model regularization, preventing overfitting of the model.

The output, y_t, for t^{th} time-step, is calculated as:

$$y_t = \text{bi-LSTM}(\vec{h}^f_{t-1}, \vec{h}^b_{t+1}, x_t) \tag{26}$$

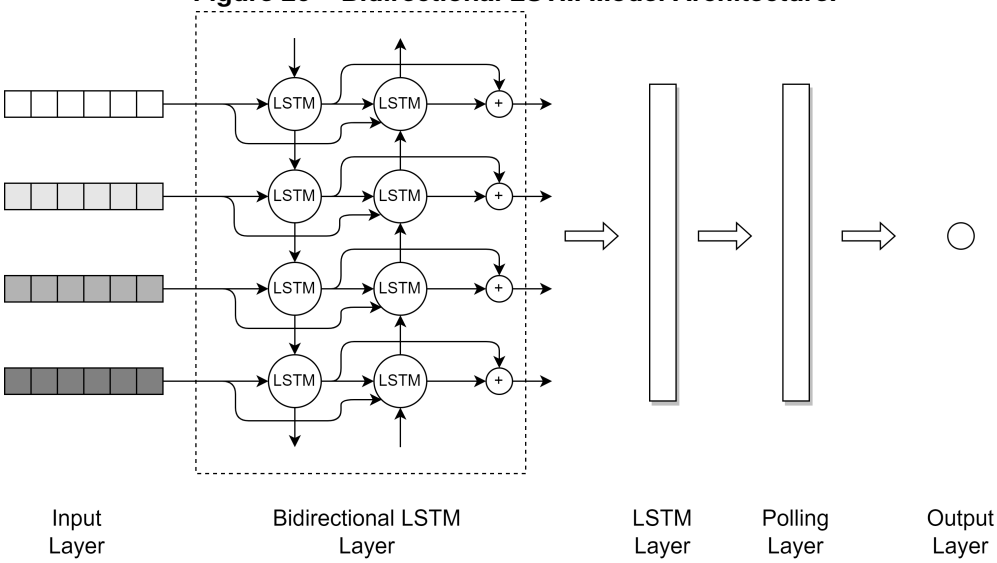
where h^f_{t-1} and h^b_{t+1} are the forward and backward hidden states respectively.

LSTM Layer: The bi-LSTM layer is followed by another LSTM layer. This layer is responsible for further processing the context-rich, forward and backward LSTM features, learning higher-level temporal representations.

Polling Layer: The layer used here is a 1D Global Average Pooling. This layer aims to reduce the dimensionality of the output tensor from the LSTM layer. It takes the average over the sequence dimension, thus preparing the output for the fully connected layer.

Dense Layer: This is a fully connected layer. In this layer, every node is connected to every other node in the preceding layer, and the output is transformed into a tensor that matches the target dimension. We finish with one output unit since we aim to have a single output predicting our target variable.

Figure 20 – Bidirectional LSTM Model Architecture.



Source: Own authorship (2023)

3.2.3 Attention LSTM Model Development

The architecture of the LSTM model with attention is illustrated in Figure 21. This model employs an LSTM layer, an attention mechanism, and a Dense layer. The attention mechanism aims to identify and focus on the most informative parts of the sequence, which is particularly beneficial when dealing with long sequences.

Input Layer: The input layer accepts the model's input sequence. The shape of this input corresponds to the dimensions of our sequence data.

LSTM Layer: LSTM (Long Short Term Memory) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is achieved through LSTM units within the network, which are designed to remember specific parts of the input between the steps of the sequences. In this LSTM layer, we carry state across the sequences with the number of LSTM units, learning important sequence features.

Attention Mechanism: Post LSTM layer, an attention mechanism is employed. The attention mechanism scores each time-step of the LSTM output sequences, identifies the most informative time steps, and accordingly allocates more weights. It consists of a Dense layer, Flatten layer, a $softmax$ Activation layer, Repeat Vector layer, and a Permute layer that collectively calculates attention weights for each time step of the sequence.

Multiply Layer: This layer multiplies the LSTM output sequence and the atten-

tion weights sequence, thus highlighting the most important time steps in the sequence.

Lambda Layer: The operation in this layer reduces the dimensionality by summing up the features across timesteps, effectively giving us a feature vector that has been attentively sum-pooled.

Dense Layer: Based on this feature vector, the final prediction is made using a Dense layer with a single output unit.

The Attention LSTM model described above can be related to Transformers, a type of model developed by Vaswani et al. (2017), which has recently revolutionized natural language processing tasks. These models are composed of an encoder and decoder part and employ a self-attention mechanism to focus on different parts of the input and output sequences, similar to the attention mechanism utilized in our model.

In the context of our Attention LSTM model:

Encoder: The encoder part of our model comprises of an input layer and an LSTM layer. It's operationally similar to an encoder in a Transformer model. The encoder mechanism aims to understand and transform the raw input data into a meaningful representation.

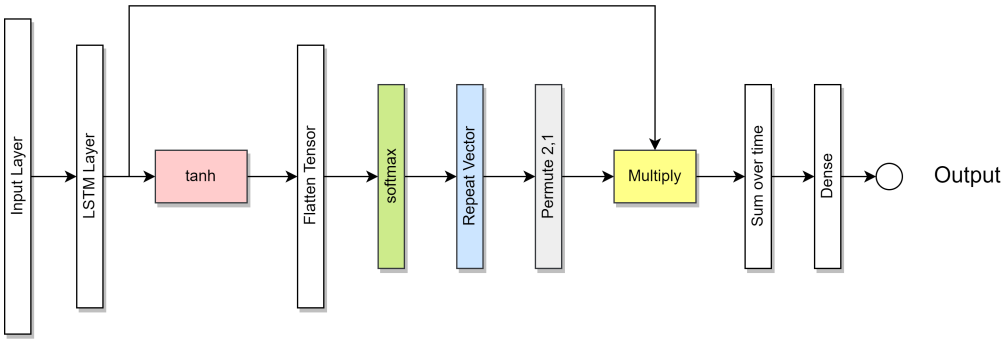
In the LSTM layer, which represents the majority of our encoding mechanism, the input sequence is processed in a time-distributed manner, creating a sequence of LSTM hidden states sensitive to temporal dependencies in the input sequence. This is similar to how the encoder part of a Transformer creates an internal representation of the input sequence by learning to pay attention to different parts of the sequence depending on the context.

Decoder: The decoder is the part of a model that transforms the encoded information into the final output. In the Transformer architecture, the decoder uses the output of the encoder along with its internal mechanisms to generate the sequence output. It's worth noting that a Transformer's decoder also utilizes attention mechanisms to iteratively generate an output sequence in an autoregressive manner.

In our Attention LSTM model, the attention mechanism and subsequent layers can be seen as pseudo-decoder components. Even though they don't explicitly generate an output sequence, they transform the encoded sequences (from the LSTM layer), into the final output. The attention mechanism garners the most information-rich parts of the sequence and the Multiply, Lambda and Dense layers process these parts to generate the final model prediction.

In conclusion, while the structure of our Attention LSTM model does not exactly replicate a Transformer's architecture, we can conceptually draw parallels between them. Both models utilize powerful sequence processing layers (LSTMs for our model and self-attention layers for Transformers) and attention mechanisms to capture and utilize sequence information for final output generation effectively. The LSTM layer in our model forms the encoding mechanism, and the components post the LSTM can be conceptually viewed as a pseudo-decoding mechanism, mirroring the foundational encoder-decoder structure of Transformer models.

Figure 21 – Attention LSTM Model Development Architecture.



Source: Own authorship (2023)

3.3 Experiment Setup

This section details the comprehensive setup of our experimentation designed to test exhaustively our deep learning models 1-D CNN, Bi-LSTM, and Attention LSTM and their efficiency in predicting the engine cylinder vibration signals. The process consists of a series of vital stages facilitating the identification of an optimized solution to our problem. These stages include the thoughtful segregation of the dataset into training, validation, and testing subsets; a meticulous hyperparameter grid search for model optimization; the strategic training of the models; and a stringent evaluation of the model performance.

Our first segment will explicate the division of our dataset into discrete subsets for training, validation, and testing. This trifurcation ensures a clear distinction between data used for learning, calibrating, and assessing the models.

Next, we dive into the exploration of the hyperparameters for each model. A dedicated Grid Search technique has been deployed to meticulously inspect different

combinations of hyperparameters, facilitating the discovery of the best set that enables our models to learn and predict accurately.

In the subsequent stage, we elucidate the model training methodology, which entails the specific techniques, parameters, and strategies applied to impart learning into the models and to adjust their internal parameters.

Finally, we impart a detailed account of our model evaluation scheme. This key step will signify the effectiveness and reliability of our models by scrutinizing their final prediction performance on unseen data, ensuring their validity and robustness. This strategic experiment setup provides a systematic approach to deep learning model development, tuning, and performance evaluation in the context of predicting engine cylinder vibration signals.

3.3.1 Data Prepocessing

Collecting data is integral, but making sense of it and preparing it for the model to consume is equally important. In this context, the raw inputs consist of engine rotation, load applied, advance applied to each cylinder, and peak detonation pressure inside each engine cylinder. The targeted output is the cylinder vibration signal, which is to be predicted.

To prepare these inputs for feeding to the sequence-to-point model, the algorithm 1 is employed. The purpose of this algorithm is to normalize the input features, extract them into a sequence of fixed length (lag), and pair each sequence with the corresponding target output. After these steps, we obtain tensors, which could be inputted into our model for training. Each instance in the tensor represents a sequence of historical data of fixed length, and each instance represents the corresponding target output.

As part of the preparation, data normalization is carried out using the MinMaxScaler transform, which rescales each feature or input value to the range [0,1]. The formula is as shown above where X represents the original feature value, X_{min} and X_{max} represent the minimum and maximum feature values, respectively, and X_{scaled} represents the normalized feature value. Essentially, this normalization step enhances the consistency and removes any potential bias in our dataset, ensuring each feature is considered on equal footing when inputted into the model.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{27}$$

After these preprocessing steps, the data is well-adapted to be fed into the sequence-to-point model for subsequent prediction of the cylinder vibration signal.

Algorithm 1 – Preprocess Data for Input and Target Sequences

Require: DataFrame df // Input DataFrame
Require: $inputs$ // List of input column names
Require: $target$ // List of target column names
Require: lag // Sequence length for creating time-series data
Require: $scaler$ // Scaler instance for normalizing input data
Ensure: X and y // Sequences for input features and target
1: Extract relevant columns from df into $data$
2: Normalize $inputs$ in $data$ using $scaler$
3: Set $sequence\_length$ equal to lag
4: Initiate empty lists X and y
5: **for** each index i from 0 to length of $data$ - $sequence\_length$ **do**
6: Append sequence of $inputs$ from i to $i + sequence\_length$ to X
7: Append $target$ at $i + sequence\_length$ to y
8: **end for**
9: **return** X, y as numpy arrays

Source: Own authorship (2023)

3.3.2 Hyper-parameters Grid Search

Hyperparameter tuning plays a critical role in obtaining a perfect fit for our predictive models. A systematic approach through a grid search was incorporated to probe for the optimal hyperparameters across a range of preset values.

The window context length, which defines the quantity of past data the models consider, ranged within the set [10, 15, 20, 25, 30], encapsulating a substantial span of prior engine operations data. For each span of window context, a distinct hyperparameter search was conducted for each model.

For the 1-D CNN model, we explored the impact of different quantities of trainable convolutional filters, which are integral for feature extraction, and distinct lengths of kernel size that confine the scope of temporal patterns the model acknowledges. This exploration manifested itself into a search over the quantity of convolutional filters within the set [16, 32, 64] and kernel sizes within the set [3, 5, 7].

When tuning the Bi-LSTM model, the alteration of unit numbers in the bidirectional LSTM layers was seen to impact the model's learning capacity of both short

and long-term dependencies. Additionally, modifying the LSTM layer's unit numbers influenced future predictions. A detailed search was embarked on for these parameters, with the number of units in bidirectional and LSTM layers ranging over [16, 32, 64].

Finally, for the Attention LSTM model, hyperparameter tuning was incorporated across the LSTM units and the densely connected layer's neural units. We endeavored to optimize the LSTM model with attention by searching over a range of LSTM units within [16, 32, 64] and neural units in the dense layer within [8, 16, 32].

Each model was subjected to numerous combinations of its corresponding hyperparameters, leading to the discernment of an optimal set of hyperparameters for each model, minimizing the loss function effectively. This elaborate Grid Search ensured our models were optimally configured to predict engine cylinder vibrations.

Given the many operating conditions in our study, including engine rotations per minute (rpm) at 1700, 2700, 3600, and 4700, both under full load and partially closed throttle, we faced a vast search space for model optimization. Given the intensive computational resources and cost required for such exhaustive experimentation under all operating conditions, an intelligent approach was devised. We decided to conduct the detailed grid search-based hyperparameter tuning process at one representative operating condition of 1700 rpm under full load in the first cylinder. This condition was chosen since it presents a reasonable balance of engine speed and load, showing a common operational state where engine efficiency is crucial. The optimal hyperparameters deduced from this experimental setup are assumed to generalize well across other operational states. Following the identification of the most performant hyperparameter set, the subsequent training step will be performed under all mentioned operating conditions using these optimized parameters. This strategy ensures the computational efficiency of our model tuning process while still promising a robust and optimum model configuration for effective predictive analysis across a range of operating conditions. It essentially offers a pragmatic balance between computational cost and model optimization scope, thereby ensuring the selection of a robust model configuration.

Moreover, this approach allows the models to take benefit from learning across diverse engine operating states in the training phase, thereby potentially improving the model's adaptability to different data scenarios. Consequently, this method helps ensure that our deep learning models are finely tuned and capable of generalizing their learning across various operating conditions, hence ensuring their effectiveness in manifold

real-world situations for predicting engine cylinder vibrations. Furthermore, carrying out forthcoming experiments under this approach enables us to provide insights into the comparative performance of the models across different operating conditions, assuring a comprehensive understanding of our model development process.

3.3.3 Model Training

During the model training process, the dataset was appropriately divided into training, validation, and testing sets with 65%, 20%, and 15% of total data, respectively. This division was made to ensure an unbiased evaluation of the training, tuning, and testing of the proposed ANN models.

All of the models were then diligently trained over a total of 100 epochs. The training phase of the models was designed to be robust but computationally efficient. We encapsulated this by integrating an Early Stopping mechanism. This effectively ceased the training process if the model's performance on the validation set didn't improve over 20 consecutive epochs, saving computational time and cost and precluding the model from overfitting (YING, 2019) on the training data.

The activation functions associated with various layers of each model, ReLU, play a crucial role in introducing non-linearity to the models (BANERJEE; MUKHERJEE; PASILIAO JR, 2019), conditioning them to learn complex mappings from the features to the target cylinder vibration signal. The ReLU activation is defined by:

$$\sigma(x) = max\{0,x\} \tag{28}$$

In the final nodes of each architecture, no activation function was used, permitting the models to freely predict continuous values that align with the regression nature of our task.

Further, the models were trained to reduce the Mean Squared Error (MSE) loss. Measure of this loss facilitated the understanding of how far the model predictions deviated from the actual data, to iteratively minimize this discrepancy during training.

The models were trained using the highly-regarded Adaptive Moment Estimation (Adam) optimizer to achieve this. As delineated in Algorithm 2, Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions predicated

on adaptive estimates of lower-order moments. This method is not only straightforward to implement but also boasts computational efficiency, minimal memory requirements, and invariance to diagonal rescaling of the gradients. These attributes make it an ideal choice for our problem, which is large in terms of data and/or parameters. Additionally, Adam's suitability extends to non-stationary objectives and problems with very noisy and/or sparse gradients, thus making it a widely preferred algorithm in training deep learning models (KINGMA; BA, 2014).

Together, these considerations ensure the training of our deep learning models while balancing performance, computational efficiency, and the prevention of overfitting. The successful execution of this training process equips our models perfectly for the subsequent evaluation phase, putting them through a series of robust tests to assess their effectiveness in predicting the cylinder vibration signal.

The fundamental aim of this approach is to allow the model to learn the complex underlying patterns in the training data and generalize this learning to unseen data. This process helps ensure that the final models are precise, reliable, and high-performing in predicting continuous values, fulfilling our research objective.

In conclusion, the model training process is critical in molding an adaptable model to learn the cylinder block vibration. Given our engine data's complexity and continuous nature, the training process was effectively designed to equip our models with a nuanced understanding of the data features, enabling the prediction.

Algorithm 2 – Adam Optimizer Algorithm

Require: α (step size)
Require: $\beta_1, \beta_2 \in [0, 1)$ (exponential decay rates for the moment estimates)
Require: $f(\theta)$ (objective function with parameters θ)
Require: θ_0 (initial parameter vector)
Ensure: θ_t (resulting parameters)
Initialize 1st and 2nd moment variables $s = 0, r = 0$
Initialize timestep $t = 0$
while not converged **do**
    $t = t + 1$
    Get gradients: $g_t = \nabla_\theta f(\theta_{t-1})$
    Update biased first moment estimate: $s = \beta_1 \cdot s + (1 - \beta_1) \cdot g_t$
    Update biased second raw moment estimate: $r = \beta_2 \cdot r + (1 - \beta_2) \cdot g_t^2$
    Correct bias in first moment: $\hat{s} = \frac{s}{1-\beta_1^t}$
    Correct bias in second moment: $\hat{r} = \frac{r}{1-\beta_2^t}$
    Update parameters: $\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{s}}{\sqrt{\hat{r}}+\varepsilon}$
end while
return θ_t

Source: (KINGMA; BA, 2014)

3.3.4 Models Evaluation

Model evaluation is essential in any machine learning pipeline to ensure effectiveness and reliability. This work utilized diverse metrics to evaluate the performance of the three described models comprehensively: 1-D CNN, Bi-LSTM, and Attention LSTM. Using these metrics, we could ascertain the models' predictive capability, reliability, and validity. In this study, we utilized three primary metrics to evaluate the performance of our models: Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Mean Absolute Error (MAE).

The MAPE measures prediction accuracy, expressing the average absolute error as a percentage of the actual values. It is calculated using the formula:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{29}$$

where

y_i represents the actual values, $\hat{y}_i$ denotes the predicted values, and n is the total number of data points.

The MSE is a commonly used regression loss function that calculates the average squared difference between the actual and predicted values. It is given by the formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{30}$$

The MAE is another measure of prediction accuracy that calculates the average absolute difference between the actual and predicted values. It is less sensitive to outliers compared to MSE and is computed using the formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{31}$$

These metrics comprehensively evaluate the model's performance, considering the magnitude of the error (through MAE and MSE) and the relative error (through MAPE). We can assess the models' predictive capability, reliability, and validity by analyzing these metrics.

The Coefficient of Determination, denoted as R^2, is a statistical measure that

shows the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It measures how well the regression predictions approximate the real data points. An R^2 of 1 indicates that all changes in the dependent variable are completely explained by changes in the independent variable(s). The R^2 score is calculated as:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{32}$$

where

SS_{res} is the sum of squares of the residual errors, and is SS_{tot} the total sum of squares.

The sum of squares of residuals, also known as the residual sum of squares (SSres), is the sum of the squares of the prediction errors. Prediction errors are the differences between the observed and predicted values. Mathematically, it can be represented as:

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{33}$$

where y_i is the observed value, $\hat{y}_i$ is the predicted value, and n is the total number of observations.

The total sum of squares SS_tot measures the total variance in the observed data. It can be calculated as the sum of the squares of the differences between the observed values and their mean. It is represented as:

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{34}$$

where y_i is the observed value, $\bar{y}$ is the mean of the observed values, and n is the total number of observations.

It is important to note that the choice of evaluation metrics should align with the objectives of the study and the nature of the data. In this case, these metrics were chosen for their ability to provide insights into the strength of the relationship between the predicted and actual values.

4 RESULTS AND ANALYSIS

In this chapter, we delve into the evaluation of our models, their comparative performance, and the conclusions drawn from the data. The evaluation of the models: 1D-CNN, Bi-LSTM and Attention LSTM were meticulously carried out with the primary focus of analyzing their comparative performance and drawing conclusions based on the gathered data. We employed the MAPE, MSE, and MAE as the metrics to evaluate the performance of these machine learning models.

The initial analysis as demonstrated in Figure 22 reveals that all the models performed optimally at a window length of 25, corroborating the robustness of our window selection strategy.

A performance comparison of 1D-CNN models with different combinations of Number of Filters and Kernel Size, as presented in Table 1, shows a consistent performance trend. The models consistently achieved a Spearman Correlation of 0.83, indicating a strong monotonic relationship between the actual and predicted values. The MAPE values varied between 21.03% and 22.12%, indicating a reasonable level of accuracy. Based on this results, a configuration of 16 filters with a kernel 03 of three was adopted during the model training across all operation conditions.

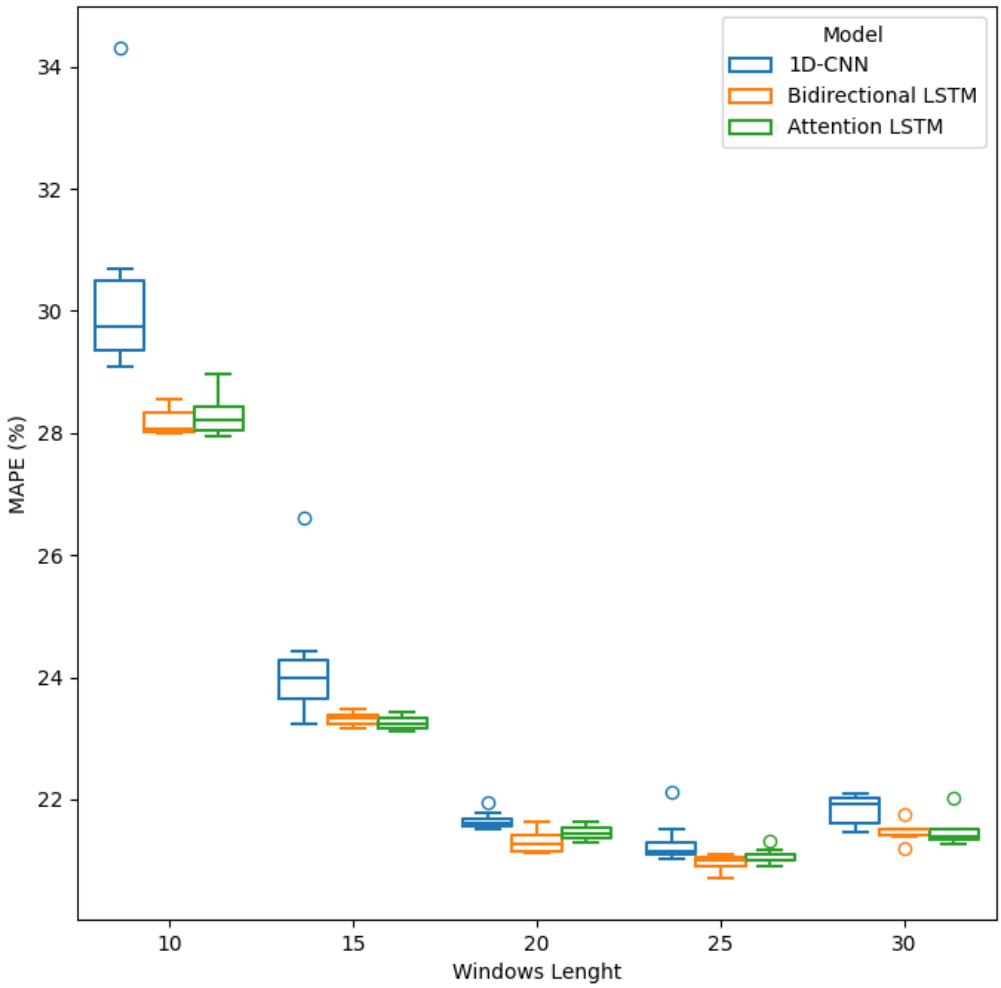
Table 1 – 1D-CNN performance per combination of Number of Filter and Kernel Size in 1700 rpm at Full Load with Windows length of 25.

Number of Filter	Kernel Size	Sperman Correlation	R2	MAPE (%)	MSE	MAE
16	3	0.83	0.64	21.03	35.12	3.82
16	5	0.83	0.64	21.07	34.82	3.83
64	7	0.83	0.62	21.11	36.08	3.87
32	5	0.83	0.62	21.14	34.99	3.83
64	5	0.83	0.61	21.15	35.81	3.88
16	7	0.83	0.61	21.18	36.10	3.88
32	7	0.83	0.65	21.30	34.59	3.79
32	3	0.83	0.62	21.50	35.32	3.86
64	3	0.83	0.64	22.12	34.30	3.80

Source: Own authorship (2023).

For Bi-LSTM model, the best combination is 32 Bi-LSTM units and 64 LSTM layer units was adopted, as outlined in Table 2. However, across all permutations, the Spearman correlation remained stable at 0.83, underscoring the robust reliability of these models. The MAPE also remained consistent, oscillating around the 21% mark, indicating a strong level of precision. Therefore, the results indicate that the Bi-LSTM

Figure 22 – Box Plot of MAPE per Windows Length for each model in 1700 rpm at Full Load.



Source: Own authorship (2023)

models, regardless of the arrangement of layer units, were capable of maintaining a reliable monotonic relationship between true and estimated values, signifying high performance and precision.

The Attention LSTM model, as presented in Table 3, show a Spearman Correlation of 0.83, the same as the Bi-LSTM and 1D-CNN models, exemplifying exquisite model reliability. Likewise, the MAPE values hovered around 21%, in line with the precision levels manifested by the other models. This indicates a relatively reasonable discrepancy between the actual and predicted values, solidifying the model's significant predictive accuracy. These results reinforce the understanding that the Attention LSTM

Table 2 – Bi-LSTM performance per combination of each Layers units in 1700 rpm at Full Load with Windows length of 25.

Bi-LSTM Units	LSTM Units	Sperman Correlation	R2	MAPE (%)	MSE	MAE
32	64	0.83	0.66	20.71	31.75	3.67
32	32	0.83	0.66	20.86	32.49	3.72
32	16	0.83	0.66	21.00	33.69	3.79
16	64	0.83	0.61	21.01	35.51	3.83
16	32	0.83	0.63	21.07	35.37	3.82
16	16	0.83	0.62	21.09	36.46	3.86

Source: Own authorship (2023).

model, with varying layer units, could provide reliable and highly accurate predictions. Based on this results, we make use of 32 Units in LSTM Layer and 8 Units Dense Layer for the model training across all operation conditions.

Table 3 – Attention LSTM performance per combination of each Layers units in 1700 rpm at Full Load with Windows length of 25.

LSTM Layer Units	Dense Layer Units	Sperman Correlation	R2	MAPE (%)	MSE	MAE
32	8	0.84	0.64	20.90	34.47	3.78
64	32	0.84	0.62	20.98	34.90	3.80
64	8	0.83	0.62	20.99	34.99	3.82
16	16	0.83	0.64	21.00	35.10	3.81
64	16	0.83	0.62	21.00	36.24	3.84
32	16	0.83	0.60	21.04	36.32	3.85
16	32	0.83	0.62	21.10	35.12	3.85
16	8	0.83	0.60	21.16	36.44	3.89
32	32	0.83	0.62	21.32	35.04	3.79

Source: Own authorship (2023).

Looking at the model evaluation tables 4, 5 and 6, we see a competitive performance across the three models for different engine conditions.

The 1D-CNN Model Evaluation Table (4) reflects the trained 1D-CNN model's performance against speed, load, and different cylinders. For Cylinder 1, under full load conditions, MAPE ranged from 19.44% at 2700 rpm to 25.19% at 4700 rpm, indicating that the model's performance dropped slightly as the speed increased. For Cylinder 2, under full load, the best MAPE was obtained at 1700 rpm with 16.51%. However, when the Throttle is partially closed at the same rpm (1700), the observed MAPE increased, showing a possible nuanced throttle impact on model performance. Cylinder 3 showed a similar trend, with the highest errors occurring at the highest rpm at both full load and partially closed throttle.

The Bi-LSTM Model Evaluation Table (5) paints a similar picture as the 1D-CNN. For Cylinder 1, the model got a 19.41% MAPE at 2700 rpm at full load and a 25.13%

Table 4 – 1D-CNN Model Evaluation

Cylinder	Load	Speed	MAPE (%)	MSE	MAE
1	Full Load	1700	21.71	35.33	3.85
		2700	19.44	14.12	2.77
		3600	21.62	26.78	3.98
		4700	25.19	47.45	5.34
	Throttle partially closed	2700	19.87	17.86	3.31
		3600	21.87	24.72	3.93
		4700	24.10	39.29	4.86
	Full Load	1700	16.51	16.67	2.67
		2700	17.55	20.92	3.45
		3600	19.51	29.66	4.12
		4700	22.75	60.96	5.96
2	Throttle partially closed	2700	14.43	16.39	3.20
		3600	22.12	37.22	4.80
		4700	23.03	56.44	5.75
	Full Load	1700	24.49	60.81	4.97
		2700	21.23	28.38	3.68
		3600	21.58	31.38	4.12
		4700	26.26	62.54	6.21
3	Throttle partially closed	2700	17.51	26.16	3.88
		3600	20.19	28.16	4.16
		4700	24.09	48.02	5.41

Source: Own authorship (2023).

MAPE at 4700 rpm at full load. Furthermore, as the throttle position changed from full to partially closed, the performance dropped slightly across most rpm ranges. The performance for Cylinder 2 and 3 also followed a similar trend as for Cylinder 1.

The Attention LSTM Model Evaluation Table (6) reflects the performance of the Attention LSTM model under various conditions. The results show that under full load, the model delivered the best performance at 2700 rpm with an MAPE of 19.16% for Cylinder 1. At partial throttle, the model's performance shows a tangible decrease across all the rpm values for the same cylinder. For Cylinders 2 and 3, a similar trend can be observed, with performance decreasing as the engine rpm increases.

In general, all three models demonstrated varying performance levels depending on cylinder, load, and speed. However, the common trend among all was the lack of performance improvement as the engine's speed increased, suggesting that model performance might be affected by high-speed, high load engine operation conditions.

Nevertheless, it can be observed from tables that under lower speed operation, we are consistently obtaining lower MAPE for all the model types and cylinder configurations. This points to the fact that given ideal operational conditions these models

Table 5 – Bi-LSTM Model Evaluation

Cylinder	Load	Speed	MAPE (%)	MSE	MAE
1	Full Load	1700	20.71	32.43	3.72
		2700	19.41	14.20	2.74
		3600	21.83	27.11	4.00
		4700	25.13	47.27	5.35
	Throttle partially closed	2700	19.68	17.62	3.30
		3600	21.92	25.13	3.96
		4700	24.11	39.06	4.87
2	Full Load	1700	16.07	14.78	2.56
		2700	17.52	20.95	3.46
		3600	19.57	30.51	4.15
		4700	22.78	62.22	6.02
	Throttle partially closed	2700	14.35	16.26	3.15
		3600	22.17	37.19	4.77
		4700	23.04	55.90	5.72
3	Full Load	1700	24.31	55.39	4.84
		2700	20.85	29.19	3.69
		3600	21.25	30.24	4.10
		4700	26.70	65.90	6.24
	Throttle partially closed	2700	17.94	27.71	3.84
		3600	20.36	28.38	4.16
		4700	24.25	49.11	5.46

Source: Own authorship (2023).

Table 6 – Attention LSTM Model Evaluation

Cylinder	Load	Speed	MAPE (%)	MSE	MAE
1	Full Load	1700	20.83	32.48	3.73
		2700	19.16	13.37	2.70
		3600	21.85	27.14	3.97
		4700	25.19	46.65	5.33
	Throttle partially closed	2700	19.74	17.60	3.29
		3600	21.83	24.81	3.93
		4700	24.08	38.94	4.86
2	Full Load	1700	16.31	16.21	2.64
		2700	17.34	20.09	3.41
		3600	19.32	28.71	4.12
		4700	22.70	60.28	5.94
	Throttle partially closed	2700	14.41	16.56	3.21
		3600	22.09	36.76	4.77
		4700	23.05	55.73	5.71
3	Full Load	1700	24.22	56.75	4.87
		2700	21.34	28.06	3.65
		3600	21.32	31.71	4.11
		4700	26.45	63.44	6.19
	Throttle partially closed	2700	17.64	25.73	3.81
		3600	20.34	28.67	4.17
		4700	24.04	48.24	5.42

Source: Own authorship (2023).

provide robust predictions with relatively low error rates.

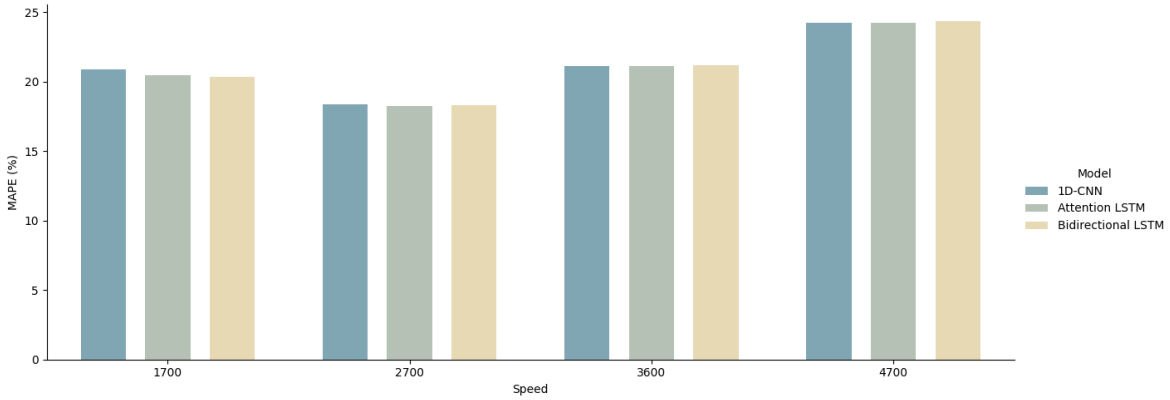
It's also essential to take note that the change in the throttle, from full load to partially closed, doesn't manifest a drastic shift in the model's performance. Although there's a marginal increase in MAPE with throttle adjustments, the predictability of our models remains satisfactory and within an acceptable range under different throttle conditions.

In the case of model performance across different cylinders, the dataset doesn't exhibit a discernible variance. Each cylinder, under the operation conditions, registered similar ranges of MAPE, suggesting our models have the potential to generalize well across different cylinders.

Overall, while the evaluation of our models does show slight differences in performance between different operation conditions, on the whole, the models have demonstrated confidence in their predictive abilities, proving themselves to be reliable tools for engine performance prediction.

The overall model performance comparison, depicted in Figure 23, illustrates that the models exhibit a near equivalent performance across different speeds. This consistency is a testament to the models' ability to generalize and predict effectively across different engine conditions.

Figure 23 – Comparing Models Performance per Speed.



Source: Own authorship (2023)

Lastly, the MAPE values for each model across different ranges of SI Engine cylinder vibrations. The evaluation Table 7 displaying the Mean Absolute Percentage Error (MAPE) for each of the models per engine vibration range offers some valuable insights into the predictive capability of our models under varying engine conditions.

Table 7 – MAPE (%) of each model per SI Engine cylinder vibration range in all operation condition.

Range	1D-CNN	bi-LSTM	Attention LSTM
4 - 14	42.43	42.17	42.09
14 - 24	17.98	17.74	17.83
24 - 34	19.40	19.58	19.37
34 - 44	30.57	30.97	30.63
44 - 54	32.04	32.48	32.20
54 - 64	29.62	29.89	29.76
64 - 74	29.34	29.01	28.81
74 - 84	26.46	26.70	26.15
84 - 94	24.53	22.77	23.18
94 - 104	26.73	26.57	24.68
104 - 114	29.04	24.06	22.50
114 - 124	14.19	9.90	5.18
124 - 134	22.01	26.28	26.61
134 - 144	35.59	39.41	32.03
144 - 154	28.17	16.77	18.90
154 - 164	23.64	19.46	15.81
164 - 174	25.65	28.75	18.75
174 - 184	28.09	27.54	16.85
184 - 194	-	-	-
194 - 204	32.51	21.57	18.61
204 - 214	-	-	-
214 - 224	-	-	-
224 - 234	33.58	22.06	26.49
234 - 244	-	-	-
244 - 254	24.67	32.78	8.11

Source: Own authorship (2023).

The 1D-CNN, Bi-LSTM, and Attention LSTM models have shown varied performance in prediction accuracy across different vibration ranges. The trend represented in the table indicates that the smallest prediction error (or, in other words, the highest accuracy) is obtained in the 114.00 - 124.00 vibration range, specifically for the Attention LSTM model, with an impressive MAPE of 5.18%, followed by the Bi-LSTM model with a MAPE of 9.90%. Similar performance for the 1D-CNN model appeared in the 194.00 - 204.00 and 224.00 - 234.00 ranges with MAPE values of 32.51% and 33.58%, respectively.

As evident from the MAPE values, the performance plateau is observed in the mid-vibration ranges for all the models. For instance, the ranges from 84.00-124.00 and 154.00-184.00 exhibited a more stable and relatively lower error rate for all three models. This could indicate the favorable operating ranges for these models where the prediction accuracy becomes relatively robust.

Contrarily, at the extreme ends of the vibration spectrum, the performance of all

the models tends to fluctuate with higher MAPE values. This behavior can be attributed to the inherent complexity associated with predicting engine performance under such high vibration ranges, which often comes with increased noise and greater variance in the data.

Interestingly, this tabulated comparison also provides insight into the models' comparative performance. Noticeably, the Attention LSTM model tends to outperform the other two models in the most vibration ranges. For instance, it demonstrated notably lower MAPE values in the 194 - 204 and 224 - 234 ranges, indicating higher predictive accuracy.

These findings indicate that, although all three models demonstrate commendable predictive capabilities throughout the central engine vibration ranges, the Attention LSTM model garners a slight edge over the others for these particular ranges of engine operation. Importantly though, the ability of each model to predict accurately across the entire spectrum of operation conditions suggests strong generalization capabilities and validates the chosen model architectures and configuration parameters. It emphasizes the efficacy of using these models for predicting engine performance across a wide range of operating conditions.

5 FINAL CONSIDERATIONS

The overall essence of this study revolves around the development, training, and evaluation of Artificial Neural Network architectures. More specifically, the focal models include 1D-CNN, Bi-LSTM, and Attention LSTM, which are employed in the prediction of Spark Ignition Engine Cylinder Block Vibration. The primary motivation behind this initiative is the broad vision to enhance the efficiency of engine calibration procedures by introducing a more streamlined and cost-effective process. The intricate correlation between knock pressure and engine block vibrations forms the core basis for our research.

The structure of this research comprises an organized sequence of steps that includes data collection, preprocessing, model development, model training, and performance evaluation – all aimed at implementing systematic improvements. This framework ensures the creation of an all-encompassing model capable of predicting knock intensities under varying operational conditions with a symbiosis of precision and accuracy.

The performance metrics achieved by each of our models provided significant insights into their relative capacities. The MAPE values for the 1D-CNN, Bi-LSTM and Attention LSTM models varied within a relatively narrow range, around 21%. The consistency in these values underlines the fact that the models can be equally efficient under most of the engine operation conditions. However, the models also demonstrated some variance in MAPE across different engine vibration ranges, suggesting that certain operational ranges prove to be more challenging for the models. Despite these nuances in their predictive capabilities, the overall performance of the models was not overly impacted.

Analyzing the performance results further, the Attention LSTM model seemed to have a slight edge in the model comparison, outperforming the other two models when examining the results from the full range of engine vibration conditions. This not only indicates the robustness and reliability of the Attention LSTM model but also emphasizes the beneficial application of attention mechanisms in machine learning models for such complex prediction tasks.

However, it's important to keep in mind that the models, in their current status, still reflect an average MAPE of around 21%. While this shows promising potential for improvement and optimization, it remains not ideal for real-world implementation

in terms of current engineering tolerance levels. This constraint indicates that further refinement and tuning of these models are required before they can be fully integrated into the practical arena of engine calibration and testing.

This research, though it might appear as a niche domain, is an integral constituent of the broader narrative around improving engine efficiency and performance. The potential avenues for future work could encompass numerous directions. It could involve refining the proposed models, exploring alternative machine learning architectures, or devising methodologies for the inclusion of these models directly within the ECU for an automated and more efficient calibration process.

Consequently, though the current suite of models may not yet be ripe for implementation, they still represent a significant progression in this complex field. We view these results as an encouraging signal for the potential of machine learning applications in the realm of engine calibration, and we look forward to the research advancements that this work could stimulate.

REFERENCES

ABBASIMEHR, H.; PAKI, R. Improving time series forecasting using LSTM and attention models. **Journal of Ambient Intelligence and Humanized Computing**, Springer, p. 1–19, 2022.

AHMED, S. *et al.* Transformers in time-series analysis: A tutorial. **Circuits, Systems, and Signal Processing**, Springer, v. 42, n. 12, p. 7433–7466, 2023.

ASHOK, B.; ASHOK, S. D.; KUMAR, C. R. A review on control system architecture of a SI engine management system. **Annual Reviews in Control**, Elsevier, v. 41, p. 94–118, 2016.

BANERJEE, C.; MUKHERJEE, T.; PASILIAO JR, E. An empirical study on generalizations of the ReLU activation function. *In*: **PROCEEDINGS** of the 2019 ACM Southeast Conference. [*S. l.: s. n.*], 2019. P. 164–167.

BARES, P. *et al.* A new knock event definition for knock detection and control optimization. **Applied Thermal Engineering**, Elsevier, v. 131, p. 80–88, 2018.

BHATT, A. N.; SHRIVASTAVA, N. Application of artificial neural network for internal combustion engines: A state of the art review. **Archives of Computational Methods in Engineering**, Springer, v. 29, n. 2, p. 897–919, 2022.

BOROVYKH, A.; BOHTE, S.; OOSTERLEE, C. W. Conditional time series forecasting with convolutional neural networks. arXiv 2018. **arXiv preprint arXiv:1703.04691**.

CHUN, K. M.; HEYWOOD, J. B. **Characterization of knock in a spark-ignition engine**. [*S. l.*], 1989.

CORTI, E.; MORO, D. Knock indexes thresholds setting methodology. **SAE Transactions**, JSTOR, p. 1016–1024, 2007.

DE WINTER, J. C.; GOSLING, S. D.; POTTER, J. Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. **Psychological methods**, American Psychological Association, v. 21, n. 3, p. 273, 2016.

DIAS, B. M. D. A. *et al.* Model-based development of an engine control module for a spark ignition engine. **IEEE Access**, IEEE, v. 6, p. 53638–53649, 2018.

DONAHUE, J. *et al.* Long-term recurrent convolutional networks for visual recognition and description. *In*: **PROCEEDINGS** of the IEEE conference on computer vision and pattern recognition. [*S. l.: s. n.*], 2015. P. 2625–2634.

DREW, P. J.; MONSON, J. R. Artificial neural networks. **Surgery**, Elsevier, v. 127, n. 1, p. 3–11, 2000.

FU, J. *et al.* Application of artificial neural network to forecast engine performance and emissions of a spark ignition engine. **Applied Thermal Engineering**, Elsevier, v. 201, p. 117749, 2022.

GÖLCÜ, M. *et al.* Artificial neural-network based modeling of variable valve-timing in a spark-ignition engine. **Applied Energy**, Elsevier, v. 81, n. 2, p. 187–197, 2005.

GUZZELLA, L.; ONDER, C. **Introduction to modeling and control of internal combustion engine systems**. [*S. l.*]: Springer Science & Business Media, 2009.

ISLAM, A. R. Machine learning in computer vision. *In*: **APPLICATIONS** of Machine Learning and Artificial Intelligence in Education. [*S. l.*]: IGI Global, 2022. P. 48–72.

JAFARI, M. *et al.* In-cylinder pressure reconstruction by engine acoustic emission. **Mechanical Systems and Signal Processing**, Elsevier, v. 152, p. 107490, 2021.

KALGHATGI, G. Knock onset, knock intensity, superknock and preignition in spark ignition engines. **International Journal of Engine Research**, SAGE Publications Sage UK: London, England, v. 19, n. 1, p. 7–20, 2018.

KARIM, F. *et al.* Multivariate LSTM-FCNs for time series classification. **Neural networks**, Elsevier, v. 116, p. 237–245, 2019.

KEFALAS, A. *et al.* Estimation of Combustion Parameters from Engine Vibrations Based on Discrete Wavelet Transform and Gradient Boosting. **Sensors**, MDPI, v. 22, n. 11, p. 4235, 2022.

KIM, S.; KANG, M. Financial series prediction using Attention LSTM. **arXiv preprint arXiv:1902.10877**, 2019.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

LAGANA, A. A. *et al.* Identification of combustion and detonation in spark ignition engines using ion current signal. **Fuel**, Elsevier, v. 227, p. 469–477, 2018.

LI, Y. *et al.* EA-LSTM: Evolutionary attention-based LSTM for time series prediction. **Knowledge-Based Systems**, Elsevier, v. 181, p. 104785, 2019.

LI, Z. *et al.* A survey of convolutional neural networks: analysis, applications, and prospects. **IEEE transactions on neural networks and learning systems**, IEEE, 2021.

MALAQUIAS, A. C. T. *et al.* The misleading total replacement of internal combustion engines by electric motors and a study of the Brazilian ethanol importance for the sustainable future of mobility: a review. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, Springer, v. 41, p. 1–23, 2019.

MAURYA, R. K.; MAURYA, R. K. Knocking and combustion noise analysis. **Reciprocating Engine Combustion Diagnostics: In-Cylinder Pressure Measurement and Analysis**, Springer, p. 461–542, 2019.

MITTELMAN, R. Time-series modeling with undecimated fully convolutional neural networks. **arXiv preprint arXiv:1508.00317**, 2015.

OFNER, A. B. *et al.* Knock detection in combustion engine time series using a theory-guided 1-D convolutional neural network approach. **IEEE/ASME Transactions on Mechatronics**, IEEE, v. 27, n. 5, p. 4101–4111, 2022.

PEYTON JONES, J. C.; SHAYESTEHMANESH, S.; FREY, J. Parametric modelling of knock intensity data using a dual log-normal model. **International Journal of Engine Research**, SAGE Publications Sage UK: London, England, v. 21, n. 6, p. 1026–1036, 2020.

PEYTON JONES, J. C.; SPELINA, J. M.; FREY, J. Optimizing knock thresholds for improved knock control. **International Journal of Engine Research**, SAGE Publications Sage UK: London, England, v. 15, n. 1, p. 123–132, 2014.

REITZ, R. D. *et al.* **IJER editorial: The future of the internal combustion engine**. v. 21. [*S. l.*]: SAGE Publications Sage UK: London, England, 2020. P. 3–10.

RICCI, F. *et al.* **Engine knock evaluation using a machine learning approach**. [*S. l.*], 2020.

RICHARD, S. *et al.* On the reduction of a 3D CFD combustion model to build a physical 0D model for simulating heat release, knock and pollutants in SI engines. **Oil & Gas Science and Technology-Revue de l'IFP**, EDP Sciences, v. 64, n. 3, p. 223–242, 2009.

SHAHLARI, A. J.; GHANDHI, J. B. **A comparison of engine knock metrics**. [*S. l.*], 2012.

SHAW, G. B. **Progress is impossible without change, and those who cannot change their minds cannot change anything.**, 1856 - 1950.

SIAMI-NAMINI, S.; TAVAKOLI, N.; NAMIN, A. S. The performance of LSTM and BiLSTM in forecasting time series. *In*: **IEEE**. 2019 IEEE International conference on big data (Big Data). [*S. l.: s. n.*], 2019. P. 3285–3292.

SIANO, D.; BOZZA, F. **Knock detection in a turbocharged SI engine based on ARMA technique and chemical kinetics**. [*S. l.*], 2013.

SIANO, D.; PANZA, M. A.; D'AGOSTINO, D. Knock detection based on MAPO analysis, AR model and discrete wavelet transform applied to the in-cylinder pressure data: results and comparison. **SAE International Journal of Engines**, JSTOR, v. 8, n. 1, p. 1–13, 2015.

TOSSO, H. G. *et al.* Spark Ignition Engine Modeling Using Optimized Artificial Neural Network. **Energies**, MDPI, v. 15, n. 18, p. 6587, 2022.

TOWERS, J. M.; HOEKSTRA, R. L. Engine knock, a renewed concern in motorsports-a literature review. **SAE transactions**, JSTOR, p. 2329–2343, 1998.

TRIMBY, S. *et al.* Unified approach to engine cylinder pressure reconstruction using time-delay neural networks with crank kinematics or block vibration measurements. **International Journal of Engine Research**, SAGE Publications Sage UK: London, England, v. 18, n. 3, p. 256–272, 2017.

TURKSON, R. F. *et al.* Artificial neural network applications in the calibration of spark-ignition engines: An overview. **Engineering science and technology, an international journal**, Elsevier, v. 19, n. 3, p. 1346–1359, 2016.

VASWANI, A. *et al.* Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

WANG, Z.; LIU, H.; REITZ, R. D. Knocking combustion in spark-ignition engines. **Progress in Energy and Combustion Science**, Elsevier, v. 61, p. 78–112, 2017.

WEN, Q. *et al.* Transformers in time series: A survey. **arXiv preprint arXiv:2202.07125**, 2022.

XIAO, B.; WANG, S.; PRUCKA, R. G. **A semi-physical artificial neural network for feed forward ignition timing control of multi-fuel SI engines**. [*S. l.*], 2013.

XIE, M. *et al.* Multi methods for combustion instability analysis of Atkinson cycle engine based on time series analysis. **Applied Thermal Engineering**, Elsevier, v. 226, p. 120298, 2023.

YING, X. An overview of overfitting and its solutions. *In*: **IOP PUBLISHING**. Journal of physics: Conference series. [*S. l.: s. n.*], 2019. v. 1168, p. 022022.

YU, X. *et al.* Internal combustion engine calibration using optimization algorithms. **Applied Energy**, Elsevier, v. 305, p. 117894, 2022.

ZENG, A. *et al.* Are transformers effective for time series forecasting? *In*: 9. **PROCEEDINGS** of the AAAI conference on artificial intelligence. [*S. l.: s. n.*], 2023. v. 37, p. 11121–11128.

ZHEN, X. *et al.* The engine knock analysis–An overview. **Applied Energy**, Elsevier, v. 92, p. 628–636, 2012.

ZOU, J.; HAN, Y.; SO, S.-S. Overview of artificial neural networks. **Artificial neural networks: methods and applications**, Springer, p. 14–22, 2009.