# Enhancing Knock Detection and Combustion Diagnostics

## A Machine Learning Approach with Ion Current Sensors

Björnsson, Ola

2025

# Enhancing Knock Detection and Combustion Diagnostics

## A Machine Learning Approach with Ion Current Sensors

OLA BJÖRNSSON
DEPARTMENT OF ENERGY SCIENCES | FACULTY OF ENGINEERING | LUND UNIVERSITY

# Enhancing Knock Detection and Combustion Diagnostics

# Enhancing Knock Detection and Combustion Diagnostics

## A Machine Learning Approach with Ion Current Sensors

Ola Björnsson


LUND
UNIVERSITY

DOCTORAL DISSERTATION
Doctoral dissertation for the degree of Doctor of Philosophy (PhD) at the Faculty of Engineering at Lund University
Thesis advisors: Prof. Per Tunestål, Prof. Andreas Jakobsson, Dr. Marcus Lundgren
Faculty opponent: Prof. Yudai Yamasaki

To be presented, with the permission of the Faculty of Engineering of Lund University, for public criticism in the M:B lecture hall at the Department of Energy Science on Friday, the 28th of February 2025 at 10:00.

Organization
**LUND UNIVERSITY**

Department of Energy Science
Box 118
SE–221 00 LUND
Sweden

Document name
**DOCTORAL DISSERTATION**

Date of disputation
2025-02-28

Sponsoring organization

Author(s)
Ola Björnsson

Title and subtitle
Enhancing Knock Detection and Combustion Diagnostics: A Machine Learning Approach with Ion Current Sensors

Abstract
Achieving robust and efficient control of internal combustion engines is critical for meeting stringent environmental regulations and reducing greenhouse gas emissions. Natural gas engines, in particular, offer significant potential for cleaner energy but present unique challenges due to variations in fuel composition and combustion dynamics. This thesis investigates novel methods for improving combustion diagnostics and control, focusing on ion current measurements as a cost-effective and versatile tool. The research encompasses three main areas: (1) closed-loop control of the combustion phase using ion current-based diagnostics, (2) development of a virtual pressure sensor leveraging artificial neural networks, and (3) knock detection using ion current as a standalone sensor and in combination with vibration sensors. All experiments were conducted on a 13-liter, six-cylinder, heavy-duty natural gas engine.

The first part of this work examines ion current-based closed-loop control of the combustion phase. By estimating the peak pressure location, the system dynamically adjusts spark timing to compensate for offsets caused by variations in fuel composition. This approach effectively restored performance when the combustion phasing deviated from nominal conditions and improved combustion stability.

Next, a virtual pressure sensor was developed using artificial neural networks to predict in-cylinder pressure from ion current signals. This method accurately predicted in-cylinder pressure, enabling precise estimation of combustion parameters such as peak pressure location, gross indicated mean effective pressure, and heat release. These findings demonstrate the feasibility of replacing costly in-cylinder pressure sensors with robust and affordable ion current sensors.

The research also focused on knock classification using convolutional neural networks trained on ion current data. These models achieved high accuracy in identifying knock events and were further improved by incorporating knock indicators from vibration sensors into a dual-input convolutional neural network architecture. This combined approach reduced cylinder-to-cylinder variability and enhanced overall detection reliability.

The findings highlight the potential of ion current measurements, combined with machine learning, to provide cost-effective and robust solutions for combustion diagnostics and control. By enabling engines to operate closer to the knock limit, these methods contribute to enhanced fuel efficiency and reduced emissions.

Key words
ion current, machine learning, knock, classification, spark-ignition

Supplementary bibliographical information

Language
English

ISSN and key title
0282-1990

ISBN
978-91-8104-372-3 (print)
978-91-8104-373-0 (pdf)

Recipient's notes

Number of pages
142

Price

Security classification



Signature _____ Date 2025-01-16

# Enhancing Knock Detection and Combustion Diagnostics

## A Machine Learning Approach with Ion Current Sensors

by Ola Björnsson


LUND
UNIVERSITY

**Funding information:** The thesis work was financially supported by KCFP Engine Research Center Swedish Energy Agency (project number 22485 – 4).



Faculty of Engineering, Department of Energy Science

ISBN: 978-91-8104-372-3 (print)
ISBN: 978-91-8104-373-0 (electronic)
ISRN: LUTMDN/TMHP-25/1179-SE
ISSN: 0282-1990

Printed in Sweden by Media-Tryck, Lund University, Lund 2025

# Contents

# List of publications

This thesis is based on the following publications, referred to by their Roman numerals:

I **Evaluation of Closed-Loop Combustion Phase Optimization for Varying Fuel Compensation and Cylinder Balancing in a HD SI-ICE**
**O. Björnsson**, P. Tunestål
SAE Technical Paper 2024-01-2837, 2024, doi: 10.4271/2024-01-2837.

II **Virtual Pressure Sensor Based on Ion Current Measurements using Artificial Neural Networks**
**O. Björnsson**, P. Tunestål
*ASME* 2024 *ICE Forward Conference*, San Antonio, Texas, USA: American Society of Mechanical Engineers. Oct. 2024, ISBN 978-0-7918-8852-0, p. V001T04A002. doi: 10.1115/ICEF2024-140748.

III **Ion Current-Based Knock Detection using Convolutional Neural Networks**
**O. Björnsson**, P. Tunestål
*IFAC-PapersOnLine*, Jan. 2024, vol. 58, no. 29, p. 261–266, ISSN 2405-8963. doi: 10.1016/j.ifacol.2024.11.154.

IV **Knock Detection with Ion Current and Vibration Sensor: A Comparative Study of Logistic Regression and Neural Networks**
**O. Björnsson**, P. Tunestål
*Energies* Nov. 2024, vol. 17, no. 22, p. 5693, ISSN 19964-1073. doi: 10.3390/en17225693.

## Author contributions

Co-authors are abbreviated as follows:
Per Tunestål (PT).

### Paper I: Evaluation of Closed-Loop Combustion Phase Optimization for Varying Fuel Compensation and Cylinder Balancing in a HD SI-ICE

P.T. and I designed the study. I performed the experiments, analyzed the data, authored the draft, and revised the paper after peer review. P.T. provided feedback and proofread all submitted paper revisions

### Paper II: Virtual Pressure Sensor Based on Ion Current Measurements using Artificial Neural Networks

I designed the study, performed the experiments, built the models, analyzed the results, authored the draft, and revised the paper after peer review. P.T. provided feedback and proofread all submitted paper revisions.

### Paper III: Ion Current-Based Knock Detection using Convolutional Neural Networks

I designed the study, performed the experiments, built the models, analyzed the results, authored the draft, and revised the paper after peer review. P.T. supervised part of the experimental campaign, provided feedback, and proofread all submitted paper revisions.

### Paper IV: Knock Detection with Ion Current and Vibration Sensor: A Comparative Study of Logistic Regression and Neural Networks

I designed the study, performed the experiments, built the models, analyzed the results, authored the draft, and revised the paper after peer review. P.T. supervised part of the experimental campaign, provided feedback, and proofread all submitted manuscript revisions.

# Publications not included in this thesis

**Artificial neural networks improve early outcome prediction and risk classification in out-of-hospital cardiac arrest patients admitted to intensive care**
J. Johansson, **O. Björnsson**, P. Andersson *et. al*
*Crit Care* **24**, 474 (2020). doi: 10.1186/s13054-020-03103-1.

**Predicting neurological outcome after out-of-hospital cardiac arrest with cumulative information; development and internal validation of an artificial neural network algorithm**
P. Andersson, J. Johansson, **O. Björnsson** *et. al*
*Crit Care* **25**, 83 (2021). doi: 10.1186/s13054-021-03505-9.

**Spark Ignition-Searching for the Optimal Spark Profile**
J. Ängeby, A. Saha, M. Lundgren, **O. Björnsson**
Heintzel, A. (eds) Internationaler Motorenkongress 2022. AUDR 2022. Proceedings. Springer Vieweg, Wiesbaden. doi: 10.1007/978-3-658-44740-3_26.

**Predicting intensive care need in women with preeclampsia using machine learning – a pilot study**
C. Edvinsson, **O. Björnsson**, L. Erlandsson, S. R. Hansson
*Hypertension in Pregnancy*, 43(1), 2024. doi: 10.1080/10641955.2024.2312165.

# Acknowledgements

Pursuing a PhD is an exciting experience, filled with both challenges and rewards. It is a journey that would not have been possible without the guidance, support, encouragement, or the occasional friendly chat over a Swedish fika with many individuals.

First and foremost, I would like to express my sincere gratitude to my supervisor, Per Tunestål, for his mentorship and for providing me with the opportunity to pursue this PhD. My gratitude also extends to my co-supervisors, Marcus Lundgren and Andreas Jakobsson, the latter of whom played a pivotal role in inspiring me to even consider a PhD.

I would also like to acknowledge the other seniors at the division for their support. Öivind Andersson, Martin Tunér, and Sebastian Verhelst, thank you.

This project represents a blend of experimental and numerical efforts, made possible through the collaboration and support of several industry partners who ensured the engine's successful operation. I thank Jakob Ängeby and Anders Johnsson from SEM for their assistance and engaging discussions. Additionally, Alexander Ölén from Volvo Penta who provided swift support whenever the engine required additional parts or documentation. Giuseppe Miletto from Metatron S.p.A who offered invaluable expertise and many hours of dedicated support during the calibration of the ECU.

I would like to acknowledge the administrative staff, Anders Schyllert, Isabella Frej, Andrea Frydenlund, Sissela Silvmarker and in particular Catarina Lindén, for their assistance and always ready to answer any questions.

I am deeply thankful to Patrik Johansson, and especially Anders Olsson, whose technical support has been truly invaluable.

To my colleagues, both current and former PhD students and postdoctoral researchers, I am deeply grateful for their camaraderie and support. Thank you, Alexandra, Amir, André, Anumpan, Axel, Beyza, Brian, Carlos, Christoffer, Luis, Magnus, Maja, Menno, Miaoxin, Nikolaos, Nika, Peter, Vikram, Xiufei, Xinda, Yachao, and Yannick for fostering a friendly and collaborative research environment.

Lastly, I would like to express my profound gratitude to my family for their unwavering support throughout this journey. I am especially thankful to my partner, Rachele Anderson, whose patience, support, and understanding have been a pillar of strength throughout this journey. Having experienced the challenges of completing her own PhD, she has been a constant source of encouragement.

## Populärvetenskaplig sammanfattning

Det är allmänt känt att utan åtgärder kommer utsläppen av växthusgaser att leda till katastrofala konsekvenser i framtiden. Transportsektorn står för en betydande del av dessa utsläpp. Trots att andelen elektriska fordon ökar, är det inom en överskådlig framtid nästintill omöjligt att helt elektrifiera den tunga fordonsflottan. Därför pågår intensiv forskning kring hur man kan förbättra och minska utsläppen från tunga fordon.

Min forskning syftar till att minska utsläppen av växthusgas från gnistantända lastbilsmotorer genom att utveckla förbättrade verktyg för förbränningsdiagnostik och motorstyrning. Detta leder i sin tur till minskad bränsleförbrukning tack vare en ökad motorverkningsgrad. För att uppnå detta använder jag mig av jonström, där tändstiftet fungerar som en sensor som samlar in information om förbränningen direkt inuti cylindern.

I en förbränningsmotor, under förutsättning att inget annat förändras, påverkas verkningsgraden av när förbränningen sker — även kallad förbränningsfasen. Verkningsgraden minskar om förbränningen avviker från den optimala tidpunkten. Det finns flera orsaker till att förbränningsfasen kan förskjutas. För naturgasdrivna motorer, som min forskning fokuserar på, finns en unik utmaning: naturgas består av flera olika komponenter. Även om metan utgör majoriteten, varierar gassammansättningen beroende på ursprung, vilket påverkar förbränningens hastighet.

I den första delen av mitt arbete visar jag att genom att använda jonströmsdata för att uppskatta förbränningsfasen och justera tändtidpunkten i realtid, kan systemet anpassa sig till förändringar i gasens sammansättning. På så sätt kan motorns verkningsgrad behållas istället för att försämras.

Anledningen till att jonströmsmetoden är användbar beror på dess beroende av tryck och temperatur inuti cylindern. Inom motorforskning och motorutveckling används oftast trycksensorer inne i cylindrarna. Det uppmätta trycket är en viktig referens för att förstå förbränningen och optimera motorn. Men trycksensorer är dyra och används därför inte i vanliga fordon. Genom att träna ett artificiellt neuralt nätverk på jonströmsdata har jag visat att trycket kan uppskattas. Från det uppskattade trycket kan viktiga förbränningsparametrar beräknas. Med hjälp av jonström kan man alltså få liknande information som annars kräver trycksensorer, men till en bråkdel av kostnaden.

I den sista delen av mitt arbete undersöks hur jonströmsdata och avancerade datadrivna metoder kan komplettera traditionella strategier för knackdetektering. Knack är ett skadligt fenomen där luft-bränsleblandningen i cylindern antänds för tidigt, vilket är en stor utmaning för gnistantända motorer. Om knack inte åtgärdas kan det orsaka motorslitage och försämrad verkningsgrad. Traditionellt används vibrationssensorer, som fästs på motorblocket, för att upptäcka knack. Men dessa sensorer är känsliga för störningar från mekaniska

vibrationer, vilket påverkar noggrannheten. Dessutom kan deras prestanda variera mellan cylindrar beroende på placering.

Genom att kombinera jonströmsdata med traditionella vibrationssensorer visar jag, med hjälp av maskininlärningsmetoder som logistisk regression och neurala nätverk, att noggrannheten i knackdetektering kan förbättras samtidigt som variationen mellan cylindrar minskar. Dessa modeller gör det möjligt att köra motorn närmare knackgränsen, det vill säga den punkt där knack börjar uppstå. Genom att ligga så nära denna gräns som möjligt utan att knack uppstår kan bränsleekonomin förbättras och risken för motorslitage minskas.

## Abbreviations

| | |
|---|---|
| AFR | Air to Fuel Ratio |
| ANN | Artificial Neural Network |
| CAD | Crank Angle Degree |
| CNN | Convolutional Neural Network |
| ECU | Engine Control Unit |
| EGR | Exhaust Gas Recirculation |
| FFNN | Feed-Forward Neural Network |
| GHG | GreenHouse Gas |
| HD | Heavy Duty |
| ICE | Internal Combustion Engine |
| ICM | Ignition Control Module |
| LR | Logistic Regression |
| MAPO | Maximum Amplitude of Pressure Oscillations |
| ML | Machine Learning |
| NG | Natural Gas |
| PPL | Peak Pressure Location |
| RNN | Recurrent Neural Network |
| SGD | Stochastic Gradient Descent |
| SI | Spark Ignited |
| TPE | Tree-structured Parzen Estimator |

# Enhancing Knock Detection and Combustion Diagnostics: A Machine Learning Approach with Ion Current Sensors

*They're funny things, accidents.*
*You never have them till you're having them.*
— Eeyore

# Introduction

Barely half a century after the inception of the Internal Combustion Engine (ICE) as we know it today, it came under scrutiny for its impact on the environment. The first observed problem was the smog in the Los Angeles basin in the 1940s, which was concluded to be caused by emissions from ICEs [1]. Since then, the scientific community has agreed that human-made climate change is driven by the excessive release of GreenHouse Gases (GHGs), and reducing these emissions is of vital importance for mitigating the impact of climate change. In recent years, with the release of the Intergovernmental Panel on Climate Change (IPCC) report, the call for action has increased. The IPCC emphasized the urgency of limiting global warming to 1.5°C above pre-industrial levels, requiring global efforts to achieve net-zero emissions by 2050 [2].

Around the world, multiple nations and associations are working towards this goal by introducing initiatives, strategies, and commitments aimed at reducing greenhouse gas emissions. These efforts include broad frameworks, such as the European Green Deal, outlining a strategy to achieve net-zero GHG emissions by 2050 [3]. Part of the roadmap detailed in the Green Deal has since been legislated through the "Fit For 55" legislative package, where the ambitious target to decrease GHG emissions from cars and vans by 55% by 2030 compared to 1990 levels has been explicitly written into EU law, making it a legal obligation [4]. Furthermore, by 2035, the EU aims for all newly registered vehicles to be zero-emission, a goal aligned with the broader transition towards sustainable energy systems [5].

As evident from the actions of the European Union, reducing emissions from vehicles is a primary focus. Fossil fuel combustion remains a dominant source of GHG emissions, with the transportation sector accounting for approximately one-quarter of these emissions in Europe in 2022, as shown in Figure 1.1. Greenhouse gases are a group of gases in the Earth's atmosphere that trap heat, the primary one being carbon dioxide ($CO_2$). $CO_2$ is particularly relevant to combustion in internal combustion engines, as it is a GHG that cannot be removed by aftertreatment systems. It is, therefore, an unavoidable byproduct of the combustion of carbon-based fuels and a key driver for so-called decarbonization.

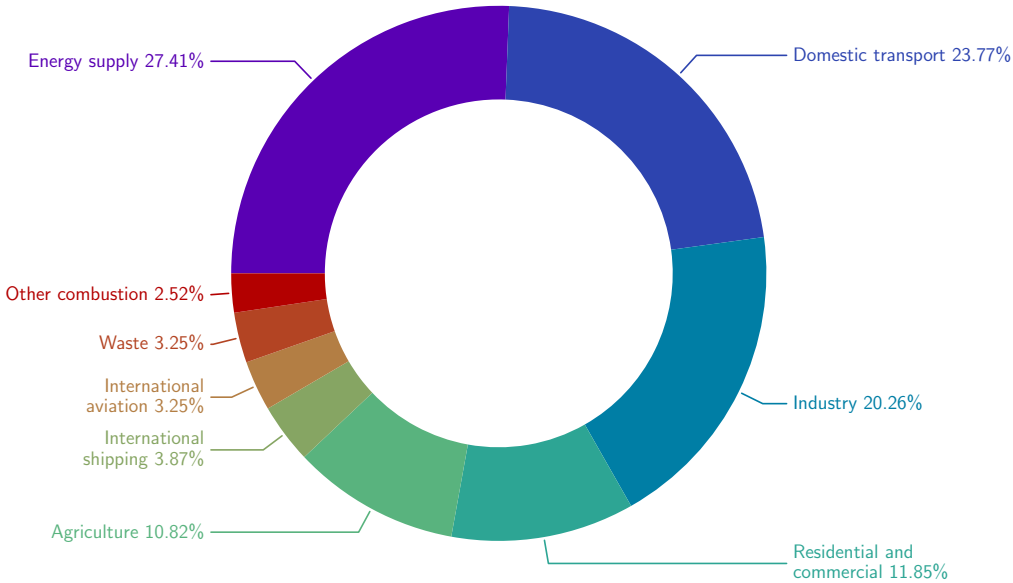



**Figure 1.1:** Distribution of greenhouse gas emissions by sector in Europe for the year 2022. Data source: [6].

Among the many solutions to reduce emissions, the adoption of alternative fuels plays a crucial role during the transition to fully electrified transportation. Natural Gas (NG) and biogas offer viable alternatives to conventional fossil fuels, contributing to reduced carbon dioxide emissions [7]. Furthermore, hydrogen can be added to NG and biogas to create Hythane, which has overall improved emissions characteristics compared to NG [8, 9].

Other technical solutions include enhancing engine control systems to allow the engine to operate more efficiently. This becomes even more important when using NG, as it is typically an unrefined fuel. Its composition, and therefore its combustion characteristics, can vary between refueling. Calibrating the engine control system to adapt to any fuel mixture is nearly impossible. Instead, adopting adaptive control systems that can handle such variability is necessary to fully harness the potential of these fuels. One feasible approach is the use of ion current measurements, a technology commonly referred to as ion sensing. Ion sensing provides direct feedback on the combustion process by utilizing the spark plug in a Spark-Ignited (SI) engine as a sensor. This technology can also be applied to other combustion types by using a dedicated ion current sensor.

Direct feedback from the combustion process is a significant step forward in developing more efficient control systems. However, while ion sensing technology is not new, it has never gained widespread adoption. This is most likely because of the stochastic nature of the measurement, which poses challenges for obtaining robust methods. Nonetheless, robustness remains critical and cannot be compromised.

In recent years, advances in computational power and the development of sophisticated Machine Learning (ML) algorithms have enabled significant progress in a variety of fields, including automotive engineering. Machine learning models, ranging from simpler approaches like linear regression to complex Artificial Neural Networks (ANNs), are particularly well-suited for analyzing large datasets and identifying patterns that may not be readily apparent through traditional methods. This capability is especially valuable when dealing with the stochastic nature of the ion current signal which makes it challenging to derive robust insights using conventional techniques.

Machine learning can address these challenges by learning patterns within the data and developing predictive models that are not only robust but also adaptive to variability in the measurements, provided it is trained on diverse and representative data, including that from different fuels. By leveraging the strengths of machine learning, it becomes possible to extract meaningful information from ion current signals, which can, in turn, be used to optimize engine control systems. This integration of advanced computational tools into engine development marks a step forward in achieving higher efficiency and lower emissions. It further contributes to the transition toward sustainable transportation.

## 1.1 Research Scope and Contributions

The initial scope of this PhD project was broad: to develop combustion diagnostics and control for natural and bio-gas engines using ion current. Subsequently, the project focused on three main components:

1. Evaluating the benefits of using closed-loop control of the combustion phase by estimating the combustion phase from the ion current.
2. Developing a virtual pressure sensor using artificial neural networks trained on ion current data.
3. Investigating knock detection using the ion current as a standalone sensor and in combination with a vibration sensor.

The research is presented through four published papers. The first paper developed a closed-loop control system using ion current-based Peak Pressure Location (PPL) estimations. Prior research has demonstrated the successful application of this type of control scheme [10]. Similarly, [11, 12] investigated controlling the combustion phase by adapting the spark timing based on CA50 estimations derived from the ion current. However, in [10, 12], the combustion phasing is adjusted by applying an offset to the open-loop spark timing, which the closed-loop control detects and begins to revert. In contrast, [11] evaluates the method by directly enabling closed-loop control without such modifications. In

our study, the goal was to evaluate the benefits of the closed-loop control scheme under variations in natural gas composition. To simulate the effect of varying composition, the initial offset in combustion phasing was achieved by adding Exhaust Gas Recirculation (EGR). Furthermore, prior research, with the exception of [11], did not report the effects of the control scheme on engine power or combustion stability. The main novelty of our study was the quantification of the benefits of the control scheme on produced engine power. The results demonstrated that the closed-loop control effectively adapted to changes in gas composition, maintaining combustion phasing and recovering lost engine power caused by the delayed combustion phase. Additionally, the closed-loop control improved combustion stability.

The primary attraction of ion sensing lies in the well-established correlation between ion current, in-cylinder pressure, and temperature. Building on this potential, the second paper introduced a virtual pressure sensor utilizing ANNs. The primary aim was to demonstrate the feasibility of replacing in-cylinder pressure sensors with ion sensing by developing an ANN capable of predicting in-cylinder pressure based on ion current data.

Given the exciting possibilities of such an approach, prior research has explored the use of ion current to predict in-cylinder pressure, with varying degrees of success. Early methods relied on parameterized functions to relate the ion current signal to the pressure trace. For instance, in [13, 14], the models were based on the ionization equilibrium theory. Although conceptually straightforward, these methods are inherently limited by the significant cycle-to-cycle variability of the ion current signal.

With advancements in computational power and machine learning, neural network-based approaches have gained traction [15–17]. For example, [15, 16] used the Adaptive Linear Neuron (ADALINE) model, a single-layer ANN with a linear activation function (Chapter 4 provides the background necessary to understand this concept). These studies showed promising results but were limited by small datasets and low sampling rates, with one sample collected per 2 crank angle degrees (CAD). In contrast, Gao et al. [17] employed a hybrid model combining a Recurrent Neural Network (RNN) and a Convolutional Neural Network (CNN). Their model predicted either the peak pressure or pressure changes with improved performance. Among these studies, only [16] incorporated additional engine operating conditions as inputs, such as throttle position, engine speed, engine load, air-to-fuel ratio (AFR), ignition timing, and compression ratio.

Our approach builds on these prior works, particularly [16], by combining ion current data with additional engine parameters. In our study, we utilized a Feed-Forward Neural Network (FFNN) and included engine speed and expected torque as input features, chosen because they map the highest level of the engine calibration. We trained the model on data collected from all six cylinders, employing a high sampling rate of 10 samples per CAD. Using the predicted pressure, we further calculated combustion parameters and heat release

rates to evaluate model accuracy comprehensively.

In summary, this study demonstrated that combining ion current data with engine speed and expected torque enables the accurate reconstruction of in-cylinder pressure traces, from which critical combustion metrics can be estimated. This approach highlights the feasibility of using ion current measurements alongside machine learning to replace expensive physical sensors, offering robust and cost-effective solutions for engine diagnostics and control.

The last two papers focused on knock detection using CNN models. During knock events, the resulting pressure waves also affect the ion current measurements, as explained in more detail in chapter 5. Several studies have demonstrated the use of ion sensing for knock detection. The vast majority of these approaches are based on the same methodology as conventional vibration sensors, namely calculating a metric from the bandpass filtered signal [18–20]. An alternative approach was proposed by Zhang et al. [21], where knock detection was performed using the wavelet transform of the ion current.

In the third paper, a CNN model was developed to classify knock events into no-knock, medium-knock, and high-knock categories using ion current measurements. To the best of our knowledge, no prior research has directly applied CNNs to raw ion current data for knock detection. The CNN achieved strong classification accuracy, particularly in distinguishing between no-knock and heavy-knock events. This work demonstrated the potential of combining ion current data with deep learning to enhance robustness and precision in knock detection.

The final paper compared logistic regression and neural networks for knock detection using both ion current and vibration sensors. This comparison provided insights into the strengths and limitations of traditional and advanced machine learning models for leveraging data from different sensors. The idea of combining information from ion current and vibration sensors was inspired by Ängeby et al. [22], who demonstrated improved performance by using metrics derived from the bandpass filtered signals of both sensors. Our findings highlighted the benefits of combining knock indicators from both sensors in a dual-input CNN model, leading to improved classification accuracy and consistency across cylinders. This study further demonstrated the effectiveness of integrating traditional and novel sensing techniques with machine learning to create robust solutions for knock detection.

## 1.2 Thesis Outline

This thesis comprises the following seven chapters.

**Chapter 1: Introduction.** The first chapter provides the background and motivation for the research.

**Chapter 2: Experimental Facilities and Numerical Tools.** Details the experimental setup and numerical tools used throughout the research. It includes details on the test engine, instrumentation, numerical tools and pressure post-processing.

**Chapter 3: Ion Sensing.** Provides an overview of ion sensing technology and its applications in internal combustion engines.

**Chapter 4: Machine Learning Models.** Focuses on the machine learning techniques applied in the research.

**Chapter 5: Engine knock.** Explores the phenomenon of engine knock, how it is measured, the signal characteristics, knock classification and typical control strategies.

**Chapter 6: Discussion.** Synthesizes the findings from the research, expanding on the discussions included in the papers.

**Chapter 7: Conclusion and Future Work.** The final chapter summarizes the key contributions of the thesis and discusses their significance. It also outlines potential avenues for future research.

# Experimental Facilities and Numerical Tools

## 2.1 Experimental Facilities

All experimental data for this thesis was gathered using a 13-liter, natural gas-fueled six-cylinder Heavy-Duty (HD) SI Volvo engine. Table 2.1 outlines the engine specifications. The engine platform is based on the familiar Volvo D13 engine, adapted for port-fueled injection of natural gas. The engine is equipped with a production Engine Control Unit (ECU) provided by Metatron S.p.A., which we have been allowed complete control over the calibration. Furthermore, the Ignition Control Module (ICM) provided by SEM AB also measures the ion current.

Table 2.1: Engine Specification

| | |
|---|---|
| Number of Cylinders | 6 |
| Arrangement | Inline |
| Fuel | CNG |
| Injection Type | Port injected |
| AF/Ratio | Stoichiometric |
| Compression Ratio | 12.4 : 1 |
| Bore | 131 mm |
| Stroke | 158 mm |
| Displaced Volume | 12.8 liters |

### 2.1.1 Instrumentation

This section provides an overview of the instrumentation used for test measurements. As mentioned earlier, the engine is controlled by a production ECU, which operates with its own array of sensors that are independent of those discussed here. The sensors integrated with the ECU will not be covered in this section.

A table summarizing key information about the sensors is provided in Table 2.2, while a simplified schematic illustrating the placement of sensors and instrumentation is shown in Figure 2.1.

The engine was connected to an ABB M3BP 355SMC three-phase electric motor, rated at 355 kW and served as a dynamometer. Brake torque was measured using an HBM T40B torque sensor that was installed between the engine and the electric motor. In-cylinder pressure was measured using AVL GU24D transducers connected to AVL microIFEM amplifiers. The crankshaft position was determined using a Leine & Linde model 520026011 incremental encoder, which was also utilized to synchronize data logging at 0.1 CADs. Air and fuel flow measurements were taken using thermal mass flow sensors, specifically the Bronkhorst F-106DI for air and the Bronkhorst F-106BI for natural gas. Temperatures were measured with standard K-type thermocouples. Low pressure measurements were taken using an array of pressure sensors manufactured by Keller. Gaseous emissions were sampled using an AVL AMA i60 emissions bench. Although this thesis does not evaluate gaseous emissions, the emissions bench estimates the EGR rate by comparing the $CO_2$ content in the intake runner and the exhaust. All sensors were connected to a LabVIEW-based program for logging the test measurements.

**Table 2.2:** Instrumentation details in the test cell.

| Function | Sensor Model | Measurement Range | Accuracy |
|---|---|---|---|
| In-cylinder pressure | Transducer: AVL GU24D<br>Amplifier: AVL microIFEM | 0 – 250 bar | – |
| Crankshaft Position | Leine & Linde 520026011 | – | 0.1 CA[a] |
| Brake Torque | HBM T40B | 0 – 10000 Nm | ±0.05% FS[b] |
| Intake air flow rate | Bronkhorst F-106DI | 0 – 1800 $m^3$n/h | ±0.1% FS |
| Fuel gas flow rate | Bronkhorst F-106BI | 0 – 300 kg/h | ±0.1% FS |
| Lambda sensor | ETAS LA3 + LSU 4.9 | – | – |
| $CO/CO_2$ | AVL AMA I60 | 0 – 1/16 % | ±0.1% FS |
| $NO_x$ | " | 0 – 25 % | " |
| HC | " | 0 – 10000 ppm | " |
| $O_2$ | " | 0 – 25 % | " |

[a] CA refers to Crank Angle.
[b] FS refers to Full Scale.

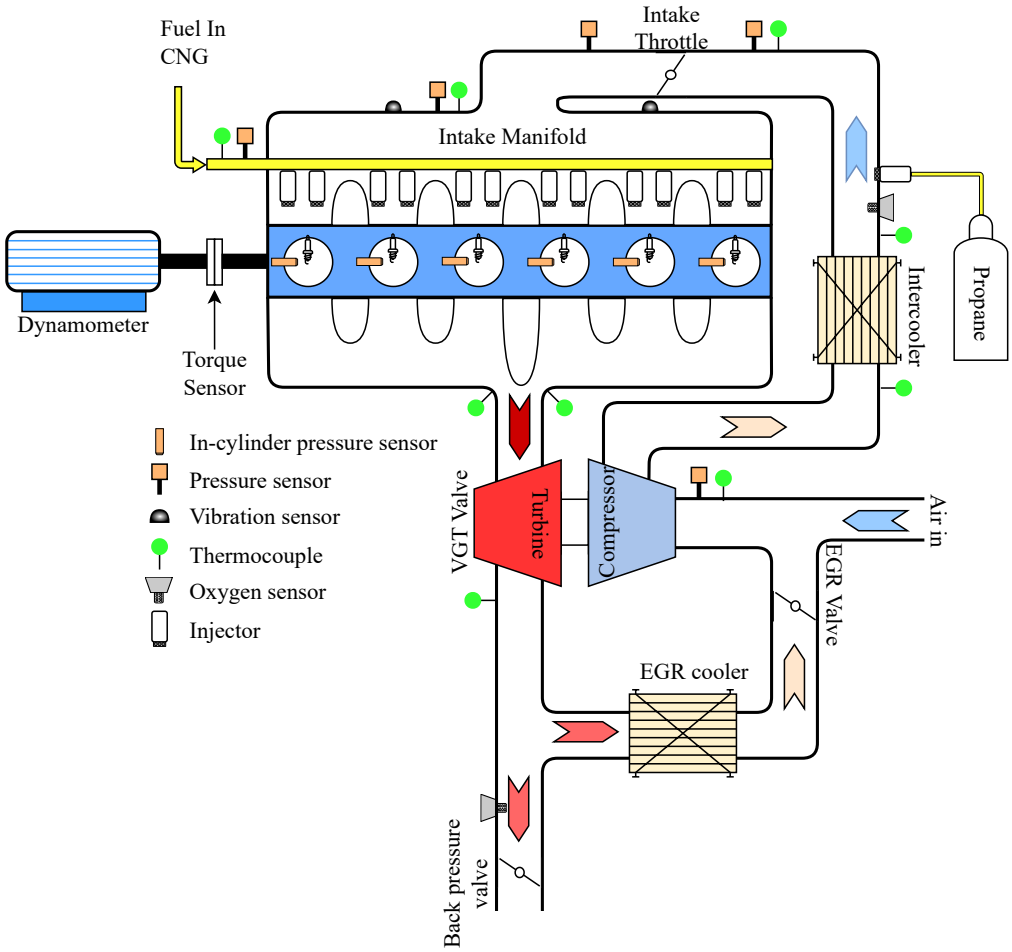



Figure 2.1: Schematic of the experimental setup.

## 2.2 Numerical Tools and Calculations

All post-processing of the experimental data for this study was carried out using Python. The neural network models used in paper II were developed using TensorFlow [23, 24], an open-source machine learning framework developed by Google. In papers III and IV, the models were implemented using PyTorch [25], another open-source framework developed by Facebook.

### 2.2.1 Pressure Pegging

Since the in-cylinder pressure is measured using a piezoelectric pressure transducer, the recorded signal lacks an absolute reference point. The first step, therefore, is to reference or "peg" the pressure to a known value to ensure the measurements align with physical reality. The most common method is to peg the in-cylinder pressure to the intake manifold pressure at the end of the intake stroke, i.e., when the piston is positioned at the bottom dead center.

However, when using natural gas, there is an additional challenge regarding calculating the heat release, specifically concerning the specific heat ratio used in the heat release calculations. Since the engine operates on natural gas supplied from the municipality's pipeline, the gas composition is uncontrolled and subject to variation. This variability affects the specific heat ratio and other fuel properties, making it unsuitable to use a fixed value, as is commonly done with refined fuels. To address this, we employ an algorithm that estimates the polytropic exponent and cylinder pressure offset both before and after combustion. A linear interpolation of these values is performed, with the interpolated pressure offset used to peg the pressure and the interpolated polytropic exponent applied to the heat release calculation. This algorithm, developed by Per Tunestål, the supervisor of this thesis, is detailed in [26]. Below, we provide a brief overview of the method but will not delve into the details of the estimation algorithm.

The method makes the following assumptions: During the high-pressure phase of the engine cycle, when the valves are closed, it is assumed that any mass transfer through the combustion chamber boundaries is negligible. This allows the combustion chamber to be thermodynamically modeled as a closed system. For further simplification, the cylinder contents are assumed to remain constant. As a result, the enthalpy of formation changes due to combustion are modeled as heat, as described by Gatowski et al. [27]. It is also assumed that the only work interaction with the environment is the $p\,dV$ work exerted on the piston.

The first law of thermodynamics states the following:

$$\mathrm{d}U = \mathrm{d}Q - \mathrm{d}W, \tag{2.1}$$

where $\mathrm{d}U$ represents the change in internal energy, $\mathrm{d}Q$ is the heat transfer to the system, and $\mathrm{d}W$ is the work performed by the system.

The instantaneous volume of the combustion chamber, $V(\alpha)$, is given as a function of the crank angle $\alpha$:

$$V(\alpha) = V_c + \frac{\pi B^2}{4}\left[l + a(1-\cos(\alpha)) - \sqrt{l^2 - a^2\sin^2(\alpha)}\right], \tag{2.2}$$

where $V_c$ is the clearance volume, $B$ is the bore, $l$ is the connecting rod length, and $a$ is the crank radius.

Using the ideal gas law, the change in internal energy can be related to pressure and volume. Substituting into the energy equation gives:

$$\mathrm{d}Q = \frac{C_v}{R}\mathrm{d}(pV) + p\,\mathrm{d}V, \tag{2.3}$$

where $C_v$ is the specific heat at constant volume, $p$ is the pressure, $V$ is the volume, and $R$ is the specific gas constant.

When neither heat transfer nor combustion occurs, $\mathrm{d}Q$ equals zero. It is also assumed that any heat transfer can be modeled as a correction to $C_v$. This correction can be derived by rewriting (2.3):

$$0 = \frac{C_v}{R}\mathrm{d}(pV) - \mathrm{d}Q + p\,\mathrm{d}V = \frac{C_v - R\frac{\mathrm{d}Q}{\mathrm{d}(pV)}}{R}\mathrm{d}(pV) + p\,\mathrm{d}V. \tag{2.4}$$

The corrected $C_v$ can then be identified as follows:

$$\tilde{C_v} = C_v - R\frac{\mathrm{d}Q}{\mathrm{d}(pV)}. \tag{2.5}$$

The algorithm aims to identify the "corrected" $C_v$ prior to and after the combustion, using these values to perform interpolation throughout the combustion process. By introducing the specific heat ratio $\gamma = \frac{\tilde{C_p}}{\tilde{C_v}}$ and the relation $R = \tilde{C_p} - \tilde{C_v}$, (2.3) can be rewritten assuming $\mathrm{d}Q = 0$ (no combustion) as:

$$0 = \left(\frac{\tilde{C_v}}{R} + 1\right)p\,\mathrm{d}V + \frac{\tilde{C_v}}{R}V\mathrm{d}p = \frac{\gamma}{\gamma - 1}p\,\mathrm{d}V + \frac{1}{\gamma - 1}V\mathrm{d}p. \tag{2.6}$$

This equation can be further simplified by eliminating the denominators and combining the differentials, leading to the polytropic relation:

$$\mathrm{d}(pV^\gamma) = 0. \tag{2.7}$$

This indicates that when $\mathrm{d}Q = 0$, $pV^\gamma$ remains constant, and identification of $C_v$ before and after combustion is therefore equivalent to identifying $\gamma$ before and after combustion.

Now, we will derive the expressions required to estimate the pressure offset needed to properly peg the pressure. The pressure measurements from the piezoelectric pressure sensor are offset by a drift term $\Delta p$, so the measured pressure $p_m$ is expressed as:

$$p_m = p + \Delta p, \tag{2.8}$$

where $p$ is the true cylinder pressure. Substituting this into the polytropic relation (2.7) yields:

$$(p_m - \Delta p) V^{\kappa} = (p_{m0} - \Delta p) V_0^{\kappa}, \tag{2.9}$$

where $p_{m0}$ and $V_0$ are the pressure and volume at a reference crank angle. Note that $\gamma$ has been replaced with $\kappa$, as the identified exponent is not necessarily the true thermodynamic specific heat ratio. This equation is rearranged into a dimensionless form:

$$\frac{p_m}{p_{m0}} - \left(\frac{V_0}{V}\right)^{\kappa} = \left[1 - \left(\frac{V_0}{V}\right)^{\kappa}\right] \frac{\Delta p}{p_{m0}}. \tag{2.10}$$

This can be expressed as:

$$y(\kappa) = \phi(\kappa)\theta, \tag{2.11}$$

with

$$y(\kappa) = \frac{p_m}{p_{m0}} - \left(\frac{V_0}{V}\right)^{\kappa}, \tag{2.12}$$

$$\phi(\kappa) = 1 - \left(\frac{V_0}{V}\right)^{\kappa}, \tag{2.13}$$

$$\theta = \frac{\Delta p}{p_{m0}}. \tag{2.14}$$

The unknown parameters that need to be estimated are the pressure offset, $\theta$, and the polytropic exponent, $\kappa$. The parameters are estimated using a nonlinear least squares estimation algorithm solved by Newton's method. However, we direct the interested reader to the original paper [26] for the details.

# Ion Sensing

Proper engine operation depends on a wide range of sensors, each playing a crucial role in ensuring optimal performance, efficiency, and emissions control. Despite the advancements in sensor technology, there remains a notable lack of sensors capable of directly monitoring the combustion process inside the engine cylinder. This limitation is particularly significant because combustion is the heart of engine operation, influencing key parameters like power output, efficiency, and emissions.

In laboratory settings—such as during research or engine calibration in manufacturing—combustion is typically monitored using in-cylinder pressure sensors. These sensors are usually piezoelectric, offering high sensitivity and fast response, which enable precise measurements of in-cylinder pressure. Such measurements are widely regarded as the gold standard for combustion diagnostics, as they provide direct insights into the combustion process. However, these sensors are both expensive and fragile, making them unsuitable for use in most commercial applications outside controlled laboratory settings.

This is where ion current sensing emerges as a promising alternative. By applying a voltage across the spark gap after the spark, the spark plug in SI engines can be used as a sensor to measure ion current. These measurements offer a cost-effective solution for directly monitoring combustion. Moreover, ion current has been shown to correlate strongly with in-cylinder pressure and temperature [28, 29]. This correlation is so significant that in-cylinder pressure can be accurately predicted using machine-learning models trained on ion current data which is demonstrated in paper II [30].

Figure 3.1 illustrates the in-cylinder pressure alongside the corresponding ion current measurement. The ion current signal is typically divided into three distinct phases, each named after the primary process responsible for generating the signal: the ignition phase, the chemical phase, and the thermal phase.

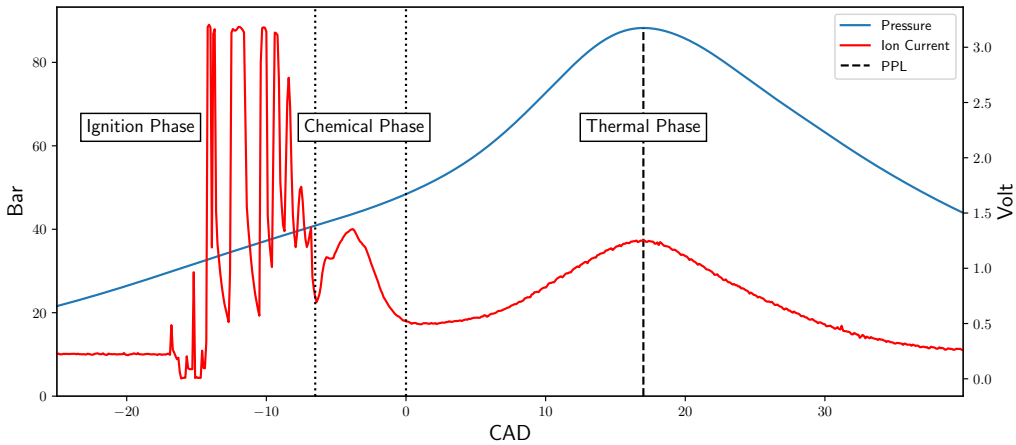



**Figure 3.1:** Comparison between the in-cylinder pressure and ion current for a single cycle. The three distinct phases of the ion current are marked.

During the *ignition phase*, the ion current is influenced by the residual electrical charge in the ignition system, resulting in a highly volatile, rapidly fluctuating and decaying signal.

In the *chemical phase*, the ion current is primarily driven by chemi-ionization occurring in the flame front surrounding the spark plug. This phase is characterized by a sharp peak that rises quickly followed by a rapid decline.

In the *thermal phase*, the ion current is dominated by thermal ionization processes occurring after the flame front has moved away from the spark plug. This phase features a broader and more stable peak that gradually decreases as the combustion chamber cools down. As the pressure increases, the temperature rises, accelerating the rate of ionization. Consequently, the thermal phase exhibits a strong correlation with in-cylinder pressure and temperature, making it the most extensively studied phase in ion current research.

## 3.1 Ion Current Formation

The phenomenon of electrical conductivity in flames has been recognized and studied for over two centuries, with the first record from Erman in 1802 [31]. In his experiment, Erman observed electrical currents in flames by inserting two wires into the flame. The notion of chemi-ionization was first introduced in 1906, when Tufts proposed that the ions responsible for this conductivity originated from chemical reactions occurring during the combustion process [32]. This proposal marked a shift in understanding, as it linked flame conductivity directly to the underlying chemistry of combustion. What these processes have in common is the creation of ions, which generate free electrons and enable a current

to flow across the spark gap, making it possible to measure the ion current (see Section 3.2 for details).

In the following subsections, the processes of chemi-ionization and thermal ionization will be described.

### 3.1.1 Chemi-ionization

Chemi-ionization occurs when reactive species, such as radicals or excited molecules, collide in a high-energy environment and undergo a chemical reaction that leads to the formation of ions and free electrons. This process is a key mechanism in flames, where the energy from the collision facilitates ionization by overcoming the activation energy barrier. The fundamental nature of chemi-ionization can be described by

$$A + B \rightarrow C + D^{+} + e^{-}, \tag{3.1}$$

which illustrates how the reactants form an ionized product and a free electron.

The chemi-ionization process can be explained using the potential energy curves depicted in Figure 3.2, adapted from [33]. These curves illustrate the variation in potential energy as the reaction progresses from the reactants ($A + B$) to the products ($C + D$) or the ionized products ($C + D^{+} + e^{-}$). The lower curve represents the ground state, which includes the lowest energy levels of the reactants and products, while the upper curve corresponds to the potential energy of the ionized products.

For ionization to occur, sufficient energy must be available to overcome the energy barrier between the reactants and the ionized products. This energy comes from three components:

1. The *heat of reaction* ($\Delta H$), which represents the energy released or absorbed during the chemical reaction.
2. The *activation energy* ($E$), which is the energy required to reach the transition state from the reactants.
3. The *ionization potential* ($V_i$) of the species being ionized, which represents the minimum energy required to remove the most loosely bound electron from a neutral atom or molecule, creating a positively charged ion.

Ionization is possible when the combined energy of $\Delta H$ and $E$ is greater than or equal to the ionization potential $V_i$,

$$V_i \leq \Delta H + E. \tag{3.2}$$

If the potential energy curve of the ionized product approaches or crosses the ground-state energy curve of the reactants, ionization can occur.

This process is more favorable when the reactants are in an excited electronic or vibrational state, as these states provide additional energy to meet the ionization threshold. The diagram also shows that for successful chemi-ionization, the potential energy surfaces of the ionized products and reactants must come close enough to allow a transition between them.

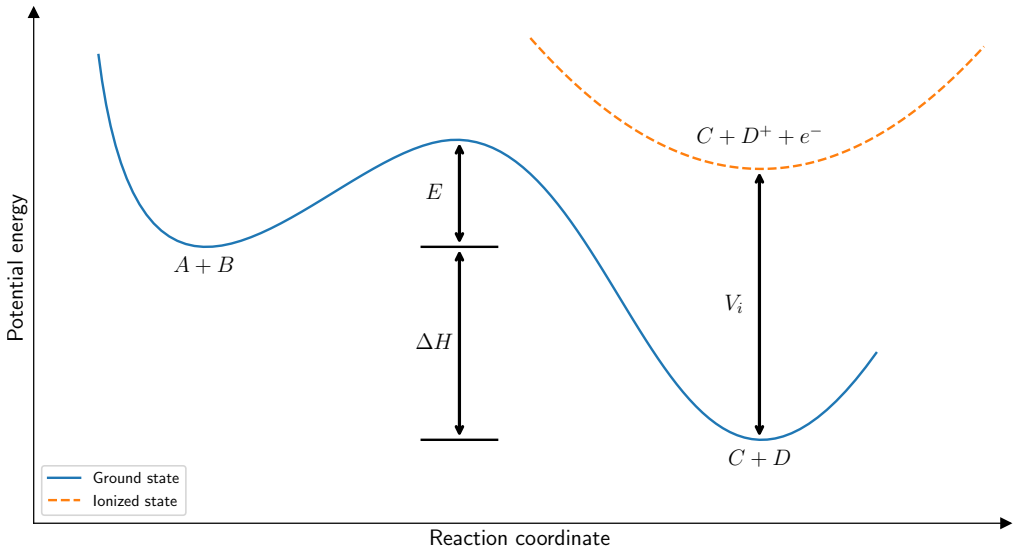



**Figure 3.2:** Potential energy diagram illustrating the reaction $A + B \rightarrow C + D^+ + e^-$, showing the energy levels for the ground state and ionized state as a function of the reaction coordinate. Adapted from [33].

As one can imagine, there is a vast number of reactions that may undergo chemi-ionization, making it very difficult to identify all of them. However, the literature generally agrees that the following reactions are the main mechanism [34–41]:

$$CH + O \rightarrow CHO^+ + e^-, \tag{3.3}$$

$$CHO^+ + H_2O \rightarrow CO + H_3O^+, \tag{3.4}$$

$$H_3O^+ + e^- \rightarrow H_2O + H \quad \text{(recombination)}. \tag{3.5}$$

Equations (3.3)–(3.5) represent the primary reactions initially identified as the dominant mechanisms for chemi-ionization in flames. These reactions involve CH and oxygen species, leading to the formation of ions such as $CHO^+$ and $H_3O^+$, which are key intermediates in the ionization process. These mechanisms were among the earliest understood and widely studied pathways of chemi-ionization in hydrocarbon combustion.

Later research expanded on these findings and revealed additional reactions contributing to chemi-ionization, particularly in flames containing hydrocarbon impurities. One such reaction involves the interaction of CH with acetylene ($C_2H_2$), resulting in the formation of $C_3H_3^+$ and a free electron,

$$CH + C_2H_2 \rightarrow C_3H_3^+ + e^-. \tag{3.6}$$

### 3.1.2 Thermal Ionization

Thermal ionization is a process in which high temperatures provide sufficient energy to remove electrons from neutral atoms or molecules, creating positively charged ions. Unlike chemi-ionization, which relies on chemical reactions and collisions between reactive species, thermal ionization is purely driven by thermal energy. This process becomes significant in environments where the temperature is high enough to overcome the ionization potential of the species involved.

At elevated temperatures, the thermal energy of particles can exceed the ionization potential ($V_i$) of an atom or molecule, allowing the most loosely bound electron to be ejected. This leads to the formation of a positive ion and a free electron, a process that can be expressed as:

$$M \leftrightarrow M^+ + e^-. \tag{3.7}$$

Here, M represents the neutral atom or molecule, $M^+$ is the resulting ion, and $e^-$ is the ejected electron. The $\leftrightarrow$ arrow indicates that the process is reversible, meaning that ionized species can recombine with free electrons to return to their neutral state. The extent to which this ionization occurs depends on the temperature, the ionization energy of the species, and the equilibrium conditions of the system.

#### Saha Equation

The degree of ionization at a given temperature can be quantitatively described using the Saha equation (3.8), [42, 43]. This equation relates the ionization fraction to the temperature, pressure, and ionization energy of the species.

$$\frac{n_{i+1} n_e}{n_i} = 2 \left( \frac{2\pi m_e kT}{h^2} \right)^{3/2} \frac{B_{i+1}}{B_i} \exp\left( -\frac{V_i}{kT} \right) \tag{3.8}$$

In this equation, $n_i$, $n_{i+1}$, and $n_e$ are the number densities of the neutral, ionized species, and free electrons, respectively. The term $m_e$ is the mass of an electron, $k$ is the Boltzmann constant, $h$ is Planck's constant, and $T$ is the temperature. The partition functions $B_i$ and $B_{i+1}$ account for the statistical weights of the energy levels for the neutral and ionized states, respectively. Finally, $V_i$ represents the ionization potential of the species, which is the minimum energy required to remove the most loosely bound electron.

The Saha equation reveals that ionization strongly depends on temperature, with higher temperatures leading to a greater degree of ionization. This dependency is mathematically expressed through the term $\left(\frac{2\pi m_e kT}{h^2}\right)^{3/2}$, which increases significantly with temperature, and the exponential term, which decreases the ionization barrier as temperature rises. In high-temperature environments, such as the post-flame zone in combustion systems, thermal ionization plays a significant role in sustaining ionization after chemi-ionization processes in the flame front have subsided.

### NO Dependency

As temperature is the main driver for thermal ionization, the composition of the post-flame gases has a significant impact on determining the degree of ionization, since ionization potential varies across species. Among the various species present, nitric oxide (NO) has been shown to contribute significantly to the free electrons due to thermal ionization in ICEs. In the post-flame region, NO is identified as a dominant contributor to the ionization process due to its low ionization potential of 9.25 eV, compared to other combustion species. This low energy threshold enables NO to produce free electrons more readily during combustion. Using thermal equilibrium analysis and Saha's equation (3.8), [13, 14] calculated that under conditions where NO concentration is approximately 1%, it is responsible for up to 95% of the free electrons.

The formation of NO predominantly occurs through the extended Zeldovich mechanism (3.9)–(3.11), which is governed by high temperatures and the availability of oxygen in the combustion gases [44]. In [45], it was suggested to add (3.12)–(3.14) to the extended Zeldovich mechanism as possible contributors to NO formation. While NO is primarily formed in the post-flame region, some contributions arise from overshoots in oxygen (O) and hydroxyl radical (OH) concentrations near the flame front [14].

$$N_2 + O \leftrightarrow NO + N \tag{3.9}$$
$$N + O_2 \leftrightarrow NO + O \tag{3.10}$$
$$N + OH \leftrightarrow NO + H \tag{3.11}$$
$$N_2O + H \leftrightarrow N_2 + OH \tag{3.12}$$
$$N_2O + O \leftrightarrow N_2 + O_2 \tag{3.13}$$
$$N_2O + O \leftrightarrow NO + NO \tag{3.14}$$

## 3.2 How to Measure Ion Current

The measurement of ion current in an SI ICE relies on utilizing the spark plug as a sensor to capture the electrical conductivity of the gas in the combustion chamber. This conductivity results from the ionization processes during combustion (as described in the previous section), where ions and free electrons are produced, enabling the gas to conduct electricity and allowing a current to flow when a voltage is applied across the spark gap.

After the spark has occurred, a bias voltage is applied across the spark gap, typically in the range of 80–150 volts. This voltage creates an electric field that drives the movement of free electrons and ions, generating a current. While both ions and electrons contribute to this current, electrons, due to their much smaller mass, dominate the flow of charge. This current, referred to as the ion current, is proportional to the ion density, which depends on various combustion parameters such as temperature, pressure, and air-fuel ratio.

To measure the ion current, a resistor is connected in series with the spark plug. The voltage drop across this resistor is amplified and is ready to be processed.

## 3.3 Applications of Ion Sensing

As emissions regulations continue to tighten, the need for a precise control of ICEs has become increasingly critical. Advancements in modern electronic systems have facilitated the integration of a growing number of sensors and actuators in vehicles. However, the addition of new sensors directly increases production costs, ultimately affecting the vehicle's price. To address this, reducing the number of sensors through intelligent control strategies, combining sensor inputs, or utilizing existing onboard hardware offers a promising solution. Ion sensing technology has shown potential in fulfilling the roles traditionally performed by separate sensors, making it an attractive alternative. The following sections provide an overview of some of the various applications using ion current.

## Pre-Ignition

Pre-ignition is an abnormal combustion phenomenon where the air-fuel mixture ignites prematurely, before the spark event. The less harmful results of pre-ignition is simply a reduction in produced work as the ignition timing will deviate from the calibration. But pre-ignition is also a precursor for super knock, which results in rapid pressure rises and potential mechanical damage.

Since the ion current is a local measurement, it may not be inherently suitable for detecting pre-ignition, which is typically initiated by hot spots located far from the sensor. However, recent studies have demonstrated promising results in this area. Both [18] and [46] observed that the ion current signal exhibited an abnormal rise in amplitude before the spark ignition signal during pre-ignition cycles. Leveraging this characteristic, both studies demonstrated that analyzing the rising edge of the ion current signal allowed for accurate detection of pre-ignition events.

## Misfire Detection

Misfiring engine cycles will naturally result in low temperatures resulting in an environment with very low levels of ionization if any. It is therefore intuitive that misfire can be easily detected using ion current. Its success in this application has been demonstrated in multiple studies across a wide range of engines and various methods for sensing ion current [20, 47–49].

Typical methods for detecting misfire involve comparing the ion current level to a predefined threshold. One of the simplest approaches is to integrate the ion current signal and compare the resulting value to this threshold. If the integrated value falls below the threshold, a misfire is detected. However, the threshold must be chosen carefully. When no combustion occurs, the ion current reading is close to zero, but setting the threshold at zero is impractical, as noise in the measurements could falsely indicate a misfire. Additionally, low speeds and loads result in lower temperatures and reduced levels of ionization, which could easily be misclassified as misfire. A good threshold should therefore be adaptive and account for variations in speed and load.

## Knock Detection

Knock in an ICE occurs when the air-fuel mixture auto-ignites prematurely or uncontrollably, leading to the formation of abnormal high pressure waves within the combustion chamber. These pressure waves can may result in reduced efficiency, increased emissions, and, in severe cases, mechanical damage to engine components. Detecting knock accurately

is critical for maintaining engine performance and preventing long-term damage. Ion current has demonstrated reliability as a method for detecting knock, and when combined with vibration sensors, the accuracy of knock detection is further enhanced. A comprehensive introduction of knock and the methods used for its detection is presented in Chapter 5.

### Air-Fuel Ratio Estimation

The perception of the feasibility of using ion current to estimate the air-fuel ratio has evolved significantly since the early years of its application. In a paper from 1986 [50], the authors concluded that, due to the large cycle-to-cycle variation in ion current measurements and their poor correlation with peak cylinder pressure and the mass fraction burned (90%), ion current would not be suitable for controlling the air-fuel ratio. The correlation was particularly weak at lean air-fuel ratios. A decade later, multiple studies demonstrated promising results by employing neural networks to estimate the air-fuel ratio [20, 51, 52]. These studies showed that ion current could be used for AFR estimation with a high degree of accuracy, however, none of them tested under very lean conditions, focusing instead on the range $0.9 \leq \lambda \leq 1.1$.

### Peak Pressure Location

As illustrated in Figure 3.1 in the introduction to this chapter, the close alignment between the peak of the thermal phase of the ion current and the peak cylinder pressure suggests that estimating the PPL from the thermal phase of the ion current is feasible.

Various methods for estimating PPL have been proposed, most of which involve fitting a parametric function to the ion current signal and determining the location of the maximum. For instance, Eriksson et al. [53] presented a method for estimating PPL by fitting the ion current signal to a parametric model comprising two Gaussian curves. ANNs have also been utilized for PPL estimation, as demonstrated by Wickström et al. [52].

The ICM used in the experimental work for this thesis employs an algorithm to estimate the PPL, with the results communicated over the CAN network. This algorithm, which forms the foundation for the closed-loop control system described in Paper 1, also relies on a parametric-based method. The complete details of this algorithm can be found in [54].

## 3.4 Ion Current Fuel Dependencies

While fuel is not the focus of this thesis, natural gas presents a unique challenge due to its inherent variation in composition. Unlike refined fuels, which are produced to meet

specific standards, natural gas is not chemically uniform. Since ion current is generated by chemical reactions, its signal is influenced by the composition of the fuel and the air-fuel mixture. Variations in these parameters can significantly affect the ion current signal as the concentrations of the ionized species varies.

Another complication stems from the fact that the part of the ion current that is of most interest for combustion diagnostic, the thermal phase, are mainly a result of thermal ionization which is dependent on the temperature. The temperature is in large depending on the equivalence ratio, and most research on application of ion current are performed on engines running with a fuel/air ratio close to stoichiometric or rich, reaching high temperatures. However, many of the currently "emerging" fuels of interest are lean-burning, such as methanol and hydrogen. Lean mixtures typically burn at lower temperatures, which do not promote ionization as effectively as stoichiometric mixtures that achieve higher combustion temperatures. Especially hydrogen is challenging as it can be used in a very broad span of air to fuel ratios, allowing for very lean combustion. The resulting reduction in ionization can be to the extent that it becomes impossible to measure any ion current during the combustion without modification to the typically used hardware. However, while little publicly available research supports the following, it is suggested that reliable measurements of ion current from lean hydrogen combustion in ICE can be obtained by modifying the measurement circuitry by increasing the signal amplification.

# Machine Learning Models

Machine learning has seen remarkable advancements in recent years, fueled by exponential increases in computational power and the availability of vast datasets. These advancements have enabled the development of increasingly sophisticated models capable of solving complex problems, including applications like image recognition to autonomous vehicles. The explosion of machine learning's popularity among the general public has been catalyzed by the appearance of large language models which have made AI-driven tools accessible to millions. These tools are now integral to everyday life, powering features like personalized social media feeds, recommendation systems, and virtual assistants.

While public attention often gravitates toward highly advanced applications, machine learning encompasses a wide spectrum of models and techniques. This includes not only cutting-edge neural networks or large-scale artificial intelligence but also fundamental approaches like logistic regression. These simpler methods are widely used for their interpretability and robustness in solving many real-world problems.

The growing interest in machine learning has also influenced research on the application of ion current in ICEs, with most recent papers focusing on machine learning models.

In this chapter, we will explore specific types of machine learning models used in this thesis, focusing on feed-forward neural networks and convolutional neural networks. Additionally, we will discuss how data is partitioned to train and evaluate these models, as well as the role of hyperparameter optimization in improving their performance. Most of the information presented in this chapter is based on the foundational concepts and explanations provided in the books: Deep Learning by Goodfellow et al. [55] and Pattern Recognition and Machine learning by Christopher M. Bishop [56]. This chapter is intended to provide a clear and accessible explanation of the methodologies used in this research, while avoiding many excessive technical details.

## 4.1 Feed-Forward Neural Networks

Feed-forward neural networks are one of the simplest and most widely used types of artificial neural networks. They consist of layers of interconnected nodes, where information flows in a single direction—from the input layer, through one or more hidden layers, to the output layer. FFNNs are designed to approximate complex functions and are commonly used for tasks such as regression and classification. Despite their simplicity compared to more advanced architectures, they form the foundation for many machine learning applications and remain an important tool.

### 4.1.1 FFNN Architecture

The FFNN consists of sequential layers of interconnected nodes, organized into three main types: the input layer, one or more hidden layers, and the output layer. Figure 4.1 illustrates a fully connected FFNN, meaning that each node in one layer is connected to every node in the next layer. The input layer receives raw data, with each node representing a specific feature and passing its values to the subsequent layer. The hidden layers transform the data, capturing patterns and relationships through computations involving weights, biases, and activation functions. Finally, the output layer produces the network's predictions, tailored to the specific task, such as classification or regression.

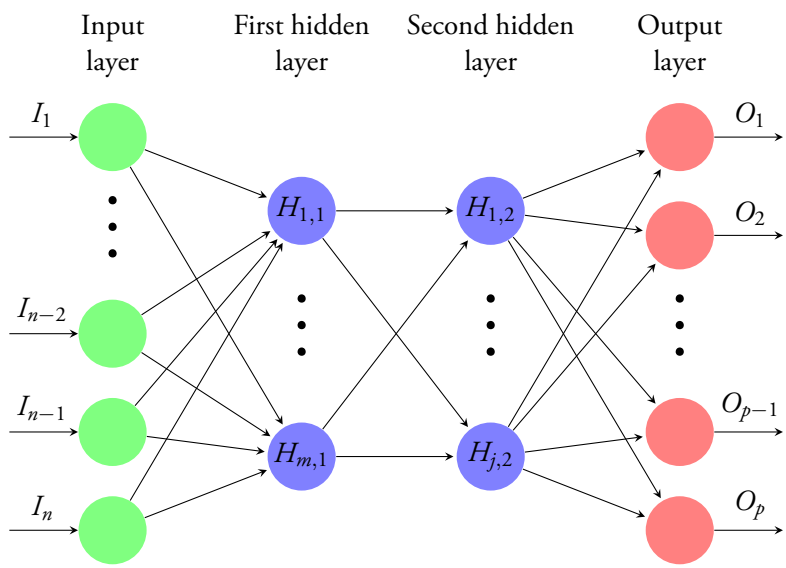



**Figure 4.1:** Structure of a feed-forward neural network with an input layer, two hidden layers, and an output layer. Each node in the input layer ($I_1$ to $I_n$) represents an input feature. The hidden layers, containing $m$ nodes ($H_{1,1}$ to $H_{m,1}$) in the first hidden layer, $j$ nodes in the second ($H_{1,2}$ to $H_{j,2}$), process the input data through weighted connections and activation functions. The output layer generates the final output of the network.

Each node in a hidden layer computes its output, or activation, by applying a non-linear activation function to the weighted sum of its inputs, combined with a bias term. The input to a layer $\ell$ is given by:

$$\mathbf{z}^{(\ell)} = \mathbf{W}^{(\ell)T}\mathbf{a}^{(\ell-1)} + \mathbf{b}^{(\ell)}, \tag{4.1}$$

where $\mathbf{W}^{(\ell)}$ is the weight matrix for layer $\ell$, $\mathbf{a}^{(\ell-1)}$ is the activation vector from the previous layer, and $\mathbf{b}^{(\ell)}$ is the bias vector for layer $\ell$. The activations of the current layer are computed as:

$$\mathbf{a}^{(\ell)} = g^{(\ell)}(\mathbf{z}^{(\ell)}) = g^{(\ell)}\left(\mathbf{W}^{(\ell)T}\mathbf{a}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right), \tag{4.2}$$

where $g^{(\ell)}(\cdot)$ represents the activation function for layer $\ell$.

The weight matrix $\mathbf{W}^{(\ell)}$ is structured such that the element $w_{ij}$ represents the weight of the connection between the $i$-th node in layer $\ell - 1$ and the $j$-th node in layer $\ell$. For instance, the activation of the $j$-th node in layer $\ell$, assuming layer $\ell - 1$ has $n$ nodes, is given by:

$$a_j^{(\ell)} = g^{(\ell)}\left(\sum_{i=0}^{n-1} w_{ij}^{(\ell)} a_i^{(\ell-1)} + b_j^{(\ell)}\right). \tag{4.3}$$

The input layer does not apply any activation function or perform transformations. Its role is to pass the raw input data $\mathbf{x}$ to the first hidden layer, where the computation begins. Activations for the first hidden layer are then computed as:

$$\mathbf{a}^{(1)} = g^{(1)}\left(\mathbf{W}^{(1)T}\mathbf{x} + \mathbf{b}^{(1)}\right). \tag{4.4}$$

The hidden layers process the data in stages, capturing progressively complex patterns as it moves deeper into the network. As shown in Figure 4.1, the first hidden layer ($H_{1,1}$ to $H_{m,1}$) extracts features from the input and sends its outputs to the second hidden layer ($H_{1,2}$ to $H_{j,2}$). Arrows connecting nodes between layers indicate both the flow of information and the associated weights. At the output layer ($O_1$ to $O_p$), this information is combined to produce the network's final predictions, such as probabilities for classification or numerical values for regression tasks.

The FFNN's capacity to model complex relationships is determined by its architecture, which includes the depth (number of hidden layers) and the width (number of nodes per layer). The input layer's width corresponds to the number of input features, while the output layer's structure depends on the specific task. For example, binary classification typically uses a one-node output layer with a sigmoid activation function, which maps the output to the range $[0, 1]$. This enables the prediction of probabilities, from which a binary decision can be produced by applying a threshold to the predicted probabilities.

During training, the weights and biases are adjusted iteratively to minimize the error between the predicted output and the real target.

### 4.1.2 Activation Functions

Activation functions play a crucial role in feed-forward neural networks by introducing non-linearity into the model. Without activation functions, the network would be limited to learning only linear relationships, regardless of the number of hidden layers. By applying a non-linear transformation to the input of each node, activation functions enable the network to capture and model complex patterns in the data.

Several activation functions are commonly used in neural networks. During the model development for the papers in this thesis four activation functions have been used: Rectified Linear Unit (ReLU), Leaky ReLU, sigmoid, and tanh. Figure 4.2 visualizes these activation functions, while their mathematical definitions are provided in (4.5)–(4.8).

The rectified linear unit function is defined as:

$$\text{ReLU}(x) = \max(0, x), \tag{4.5}$$

where it outputs the input directly if it is positive and zero otherwise which helps to maintain the gradient for positive values. This simplicity makes ReLU efficient and widely used in hidden layers.

The leaky ReLU function addresses the issue of "dead" nodes that may occur while using the ReLU function by allowing small gradients for negative inputs. A dead node refers to a node that outputs zero for all inputs, effectively becoming inactive and no longer contributing to learning during training. It is defined as:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } x > 0, \\ \alpha x & \text{if } x \leq 0, \end{cases} \tag{4.6}$$

where $\alpha$ is a small constant (e.g., 0.01) that determines the slope for negative values.

The sigmoid function maps inputs to a range between 0 and 1:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \tag{4.7}$$

The tanh function maps inputs to a range between $-1$ and 1, often providing better gradient flow during training compared to sigmoid:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{4.8}$$

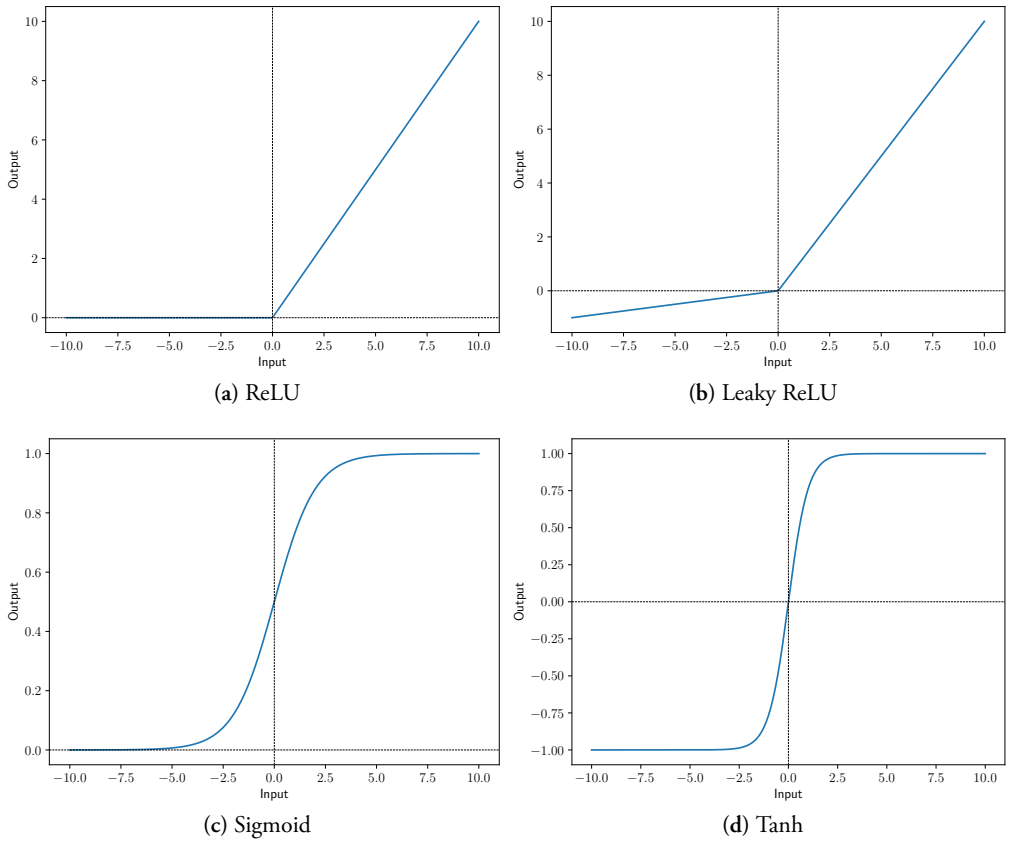
The choice of activation function depends on the network's structure and the task it is designed to perform. In hidden layers, ReLU and its variant, leaky ReLU, are often effective starting points, but experimentation during model development is essential to determine the most suitable option for a given problem. For the output layer, the choice of activation function is dictated by the task.

Binary classification typically uses the sigmoid function to map outputs to probabilities between 0 and 1, while multi-class classification relies on the softmax function, defined as:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}, \quad i = 1, 2, \ldots, n, \tag{4.9}$$

to convert the outputs of the network into probabilities that sum to 1, ensuring they form a valid probability distribution across $n$ classes. In regression tasks, a linear activation function is typically used, enabling the network to produce continuous outputs without bounds.

Non-linearity is essential for solving complex problems, as it allows the network to approximate any continuous function, given sufficient capacity. This property enables neural networks to model intricate relationships in data, making them powerful tools for a wide range of tasks.




(a) ReLU (b) Leaky ReLU

(c) Sigmoid (d) Tanh

**Figure 4.2:** Figure 4.2: Plots of common activation functions. (a) ReLU: Outputs the input if positive, otherwise zero. (b) Leaky ReLU: Allows small gradients for negative inputs. (c) Sigmoid: Maps inputs to [0, 1]. (d) Tanh: Maps inputs to [-1, 1].

### 4.1.3 Training Process

The training process of a neural network is based on supervised learning, where the network learns from labeled data to make accurate predictions. During training, the network iteratively updates its parameters to minimize the error between its predictions and the actual target values. This process consists of several steps that enable the network to learn effectively.

#### Forward Propagation

Forward propagation is the first step in the training process, where the input data is passed through the network layer by layer to calculate the output. At each layer, the data is trans-

formed using the weights, biases, and activation functions, ultimately producing an output that represents the network's prediction. The network can be seen as applying a function to our input data as:

$$\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta}), \tag{4.10}$$

where $\mathbf{y}$ is the predicted output, $\mathbf{x}$ represents the input data, and $\boldsymbol{\theta}$ encapsulates all trainable parameters in the network. The function $f$ represents the series of transformations performed by the network, including linear operations and activation functions, across its layers.

**Loss Function**

The loss function measures the difference between the predicted outputs and the actual target values. This difference, often referred to as the error, provides a quantitative assessment of how well the network is performing. In this thesis, different loss functions were used depending on the task at hand. Mean squared error (MSE), defined in (4.11), was used in Paper II for predicting the in-cylinder pressure trace as it is a regression task. The MSE quantifies the average squared difference between predicted and actual values. For multi-class classification in Paper III, the categorical cross-entropy loss, shown in (4.12), was employed. Finally, binary cross-entropy loss, detailed in (4.13), was used in Paper IV for binary classification.

$$\text{MSE Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.11}$$

$$\text{Categorical Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\hat{y}_{i,c} \tag{4.12}$$

$$\text{Binary Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{4.13}$$

In these equations:

- $N$ represents the number of samples.
- $C$ represents the number of classes in multi-class classification.
- $y_i$ is the actual target value or label for sample $i$, and $y_{i,c}$ is the one-hot encoded label for class $c$ in multi-class classification.

- $\hat{y}_i$ is the predicted value or probability for sample $i$, and $\hat{y}_{i,c}$ is the predicted probability for class $c$.

Note that the MSE loss (4.11) assumes a single output. This can naturally be generalized to handle multiple outputs by averaging the squared errors across all output dimensions, but we refrain from doing so here to simplify the notation.

### Backpropagation

Backpropagation is the process of calculating gradients for each parameter in the network based on the error computed by the loss function. These gradients are determined using the chain rule of calculus, propagating the error backward through the network. This step identifies how each weight and bias contributes to the error, allowing the network to adjust them accordingly.

One of the key advantages of backpropagation is its efficiency in computing gradients for all parameters simultaneously using the chain rule. This efficiency arises because backpropagation reuses intermediate computations from the forward pass, avoiding redundant calculations. As a result, the computational complexity scales linearly with the number of parameters, making backpropagation feasible for training even large networks with millions of parameters.

### Optimization

Optimization involves updating the network's weights and biases to minimize the loss function. This is typically done using algorithms such as Stochastic Gradient Descent (SGD) or its variants, such as Adam [57], which is the optimization algorithm used to train the models for the papers in this thesis. Adam is particularly advantageous because it combines the benefits of momentum and adaptive learning rates. The optimizer uses the gradients calculated during backpropagation to adjust the parameters, gradually improving the network's predictions over successive iterations.

Training neural networks on large datasets involves dividing the data into smaller subsets called batches. Rather than computing gradients over the entire dataset, which is computationally expensive, gradients are calculated for each batch, and the network parameters are updated iteratively based on the gradients computed for each batch. This approach, known as mini-batch gradient descent, is the most commonly used variant of gradient descent in modern deep learning.

To keep the terminology correct, SGD updates the parameters using a single data point at

each iteration, whereas batch gradient descent computes gradients over the entire dataset before updating the parameters. Mini-batch gradient descent strikes a practical balance by computing gradients over smaller subsets of the data, offering efficiency without the high memory demands of batch gradient descent or the noisiness of true SGD.

The iterations during training a machine learning model are typically referred to as epochs. An epoch corresponds to one complete pass through the entire training dataset, where the network sees all training samples exactly once. For large datasets, using mini-batches within each epoch makes training computationally feasible by reducing memory usage and enabling efficient parallel computation on modern hardware, such as GPUs.

### 4.1.4 Advantages and Limitations

FFNNs offer a simple yet powerful framework for solving a wide range of tasks, including regression, classification, and basic time-series prediction. Their main advantage lies in their simplicity and flexibility, making them easy to implement and often less computationally demanding compared to more complex architectures.

However, FFNNs also have notable limitations. Their fully connected nature can result in inefficiency and a tendency to overfit (see Section 4.3.1), particularly when handling high-dimensional input data such as images or structured signals. Moreover, FFNNs assume all inputs are independent and do not include any components explicitly designed to capture relationships between neighboring features, making them less suited for processing structured inputs.

In contrast, convolutional neural networks are specifically designed to address these shortcomings. By introducing sparsity and weight sharing through convolutional layers, CNNs efficiently capture local correlations and spatial patterns in the data. This specialization makes CNNs particularly effective for tasks like image processing and structured data analysis, where patterns among neighboring features are crucial. Furthermore, the architectural design of CNNs significantly reduces the number of trainable parameters compared to FFNNs, lowering computational complexity and often improving performance and generalization for structured data tasks.

Despite these advantages of CNNs, FFNNs remain a valuable tool for problems where input data does not exhibit clear spatial or sequential structure. For tasks with limited data or lower computational resources, FFNNs are often preferred due to their simplicity and efficiency. Furthermore, as shown in [30] (Paper II), FFNNs can still achieve strong results even when applied to signals with a sequential structure.

## 4.2 Convolutional Neural Networks

Convolutional neural networks are a class of deep learning models that excel in capturing spatial and temporal patterns. The foundational ideas behind CNNs can be traced back to Fukushima's neocognitron [58], which introduced the concept of hierarchical feature extraction inspired by the human visual system. However, the neocognitron lacked trainable parameters and the supervised learning capability that defines modern CNNs. It was LeCun [59] who formalized CNNs as we know them today, introducing convolutional layers with trainable weights and the use of backpropagation for supervised learning. Although CNNs were not initially named as such, LeCun's work laid the groundwork for their development into one of the most widely used architectures in machine learning. Subsequent work by LeCun et al. demonstrated the practical potential of CNNs in document recognition tasks [60], while the introduction of AlexNet by Krizhevsky et al. [61] revolutionized the field, showcasing the scalability and efficacy of CNNs in large-scale datasets.

The strength of CNNs lies in their ability to efficiently process structured data, such as images or time-series signals, by leveraging convolutional layers to detect features like trends, edges, or periodicities. These features are then distilled through pooling operations, which reduce dimensionality while preserving essential information, making CNNs particularly robust to noise and minor variations in the input data.

The details regarding the activation functions (see Section 4.1.2) and the training process (see Section 4.1.3) discussed in the FFNN section also apply to CNNs, albeit with some differences due to the convolutional layers. These details will therefore be omitted here for brevity.

### 4.2.1 CNN Architecture

CNNs are composed of two main components: feature extraction and classification. The feature extraction component is responsible for identifying local patterns in the input data and combining them to form increasingly complex representations as the data propagates through the network. The classification component utilizes these extracted features to make predictions, typically through fully connected layers that integrate the learned information into a final decision.

#### Feature Extraction

The feature extraction component of a CNN consists of multiple convolutional layers, each designed to transform the input data by detecting meaningful patterns. A typical convolutional layer, as illustrated in Figure 4.3, is composed of three key operations: a *convolution*

*operation* that applies filters to the input, a *detector (activation) layer* that introduces non-linearity, and an optional *pooling layer* that reduces the dimensionality of the feature maps.

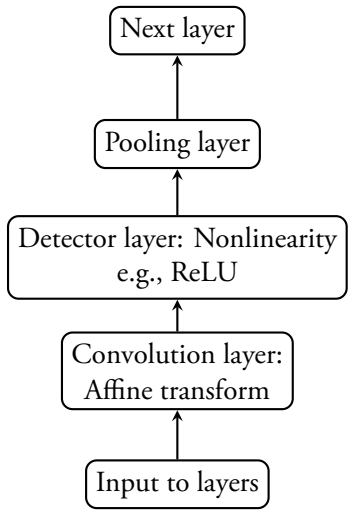



**Figure 4.3:** Illustration of a CNN layer pipeline including convolution, detector (activation function), and pooling layers. Adapted from [55].

The convolution operation applies filters, also referred to as kernels, that slide across the input data. These filters are meant to detect localized patterns, such as edges, trends, or periodicities, depending on the data. Each filter has a defined size, typically smaller than the input data, such as $3 \times 3$ for images or 3 for one-dimensional signals. The stride determines the step size by which the filter moves across the input. A stride of 1 ensures that the filter overlaps adjacent regions, while a larger stride reduces overlap and results in a smaller output feature map. The output of the convolution operation is a set of feature maps, to which an activation function is applied to introduce non-linearity, similarly to FFNNs.

As an example, consider a one-dimensional input signal $x$ convolved with a filter $w$, as shown in Figure 4.4. In this example, the input has a length of 6 ($x_0, x_1, \ldots, x_5$), the filter has a size of 3, and the stride is 1. The filter slides across the input signal, computing weighted sums of the overlapping regions, which results in an output feature map. The output values $y_0, y_1, y_2, y_3$ are given by:

$$y_0 = x_0 w_0 + x_1 w_1 + x_2 w_2, \tag{4.14}$$
$$y_1 = x_1 w_0 + x_2 w_1 + x_3 w_2, \tag{4.15}$$
$$y_2 = x_2 w_0 + x_3 w_1 + x_4 w_2, \tag{4.16}$$
$$y_3 = x_3 w_0 + x_4 w_1 + x_5 w_2. \tag{4.17}$$

Here, the filter starts at the first position and moves one step at a time (stride 1) until it reaches the end of the input. With a filter size of 3 and an input length of 6, this produces an output feature map of length 4 ($y_0, y_1, y_2, y_3$). Note that the activation part is left out from the example to reduce complexity and focus on the convolution operation itself.

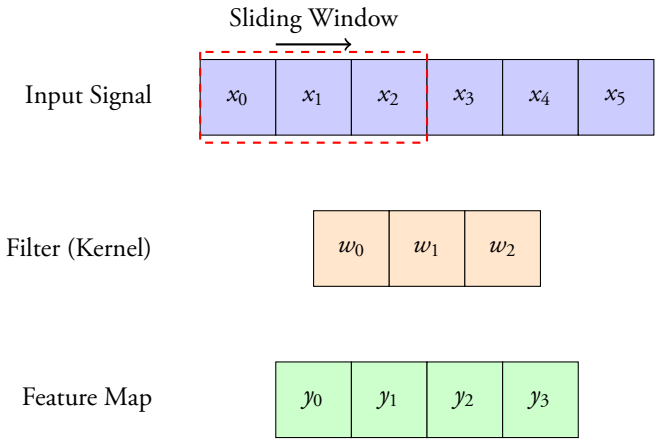



**Figure 4.4:** Illustration of 1D convolution. A 3-element filter slides horizontally over a 1D input signal, computing a weighted sum of overlapping regions to produce the feature map.

In the previous example only one filter is applied. In practice, multiple filters are applied simultaneously in each convolutional layer. This allows the network to extract a variety of features from the input data within the same layer. The resulting set of feature maps is then passed to the next layer for further transformations. Each filter has its own set of weights, which are trainable parameters updated during the optimization process. Through training, the network learns to adjust these weights to detect the most relevant patterns in the data.

To further reduce the computational cost and make the network more robust to small shifts or distortions in the data, pooling layers are often applied following the convolution and activation operations. Pooling summarizes the feature maps by aggregating information within small regions, typically selecting the maximum (max pooling) or average (average pooling) value.

Additional components such as batch normalization are often incorporated to improve the training process. Batch normalization is commonly applied after convolutional layers to standardize the output, accelerating convergence and reducing sensitivity to initialization [62].

### Classification

The classification component of a CNN consists of fully connected layers, which function exactly like the feed-forward neural network described earlier. The output from the feature extraction part is flattened into a one-dimensional vector and passed to these layers, where the extracted features are aggregated and combined to produce the final predictions.

## 4.2.2 Dual-Input CNN

In the papers comprising this thesis, two versions of the CNN architecture were used. The first is a single-input model, which corresponds to a typical CNN as described in the previous section. The second is a dual-input model, which extends the single-input architecture by incorporating additional information alongside the ion current signal.

The dual-input CNN includes knock indicators ($KI_v$ and $KI_i$) alongside the ion current signal (see Section 5.1 for the definition of $KI_v$ and $KI_i$). In this architecture, the ion current signal is first processed through convolutional layers to extract features. The resulting feature maps are then flattened and combined with the knock indicators ($KI_v$ and $KI_i$) before being passed into the fully connected layers. By including knock indicators, the network gains access to additional high-level features that complement the features learned from the ion current signal.

The conceptual flowchart illustrating the dual-input CNN architecture is provided in Figure 4.5. This diagram visualizes how knock indicators are integrated into the fully connected layers, while convolutional features are extracted from the ion current.

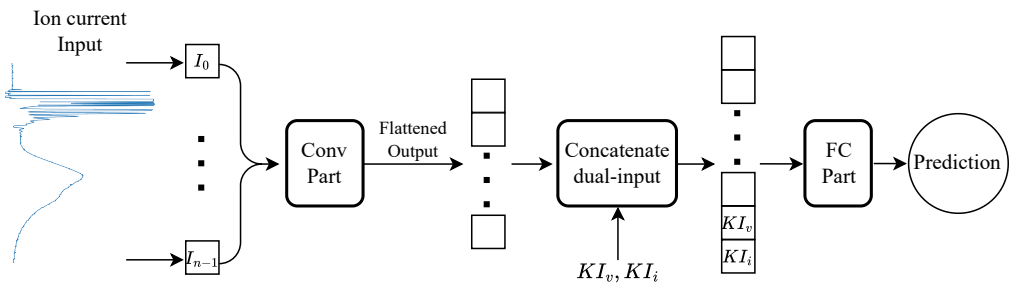



Figure 4.5: Flowchart of the dual-input CNN architecture. Knock indicators ($KI_v$ and $KI_i$) are incorporated into the fully connected layers alongside features extracted from the ion current signal.

## 4.3 Generalization and Regularization

The primary goal of any machine learning model is to make accurate predictions or perform reliable regression on unseen data. This ability is known as generalization. To assess a model's generalization capability, it is common to divide the data into separate subsets for training, validation, and testing. During training, the model learns from the training data, minimizing the training error, which represents the error the model makes on this subset. The validation set is used to tune hyperparameters, evaluate different models, and prevent overfitting by monitoring performance during training. Finally, the generalization error, also referred to as the test error, is evaluated on the test set. This error quantifies how well the model performs on unseen data and is typically the final step after identifying the model that performs best during training and validation.

### 4.3.1 Underfitting and Overfitting

Goodfellow et al. [55] highlight two primary factors that decides on a network performance:

1. Minimizing the training error.
2. Minimizing the gap between training error and test error.

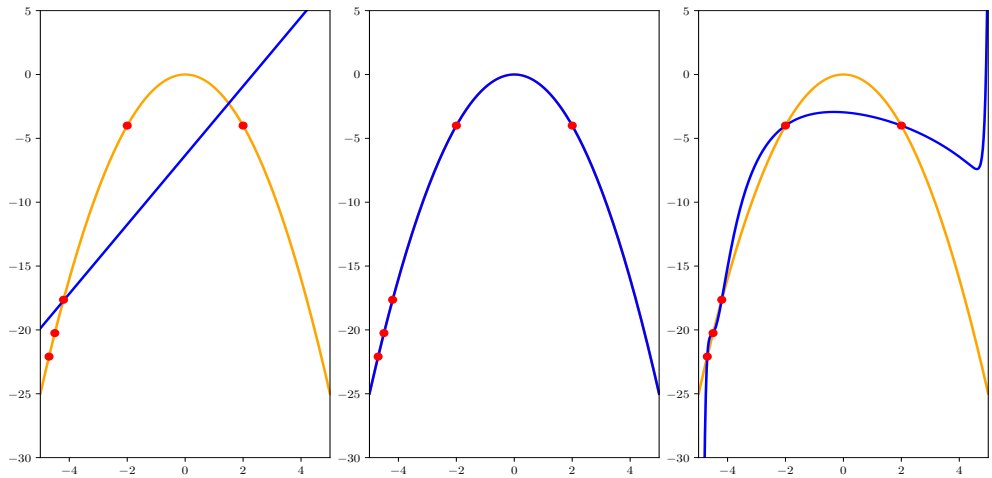
Low training error can be achieved by increasing the network's capacity, allowing it to effectively create a one-to-one mapping between inputs and outputs. However, this will result in overfitting, where the model performs exceptionally well on the training data but poorly on the test data due to its inability to generalize.

Conversely, reducing the network's capacity can prevent overfitting, as the model becomes less capable of memorizing the training data. However, this might result in underfitting, where training and test errors remain high because the model fails to capture the underlying patterns in the data.

Figure 4.6 illustrates the concepts of underfitting and overfitting, along with their impact on model performance. In the leftmost plot, underfitting occurs when the model is too simple to capture the underlying pattern of the data. Here, a linear polynomial is fitted to the samples, resulting in a poor representation of the true distribution (yellow line) and a high training error. In the center plot, the model has an appropriate capacity, fitting a quadratic polynomial that closely aligns with the true distribution while generalizing effectively to unseen data. This scenario represents the ideal balance between bias and variance. Naturally, in practice, a fit this good is not achievable due to the stochastic nature of real-world data, where noise and variability make it challenging to capture the underlying

distribution perfectly. In the rightmost plot, overfitting is demonstrated by fitting a high-degree polynomial to the samples. While the model perfectly captures the training data, it fails to generalize to new data, as evidenced by its deviation from the true distribution.

To address these challenges, various techniques collectively referred to as regularization are employed.



(a) Underfitting (b) Good fit (c) Overfitting

Figure 4.6: Illustration of underfitting and overfitting. The red dots represent samples drawn from the true distribution (yellow line), while the blue line shows the resulting polynomial fit after training on the samples. Adapted from [55].

### 4.3.2 Regularization

Regularization techniques aim to improve generalization by mitigating overfitting while ensuring that the model has enough capacity to avoid underfitting. Below, several regularization methods are described, which were employed while training the models presented in the papers included in this thesis.

#### Norm Penalties

Norm penalties, such as $L_1$ and $L_2$ regularization, are widely used to constrain the model's parameters during training. In short, they add penalty terms $\Omega(\mathbf{w})$ to the loss function $E$, resulting in the modified loss function $\tilde{E}$, defined below:

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda\Omega(\mathbf{w}). \tag{4.18}$$

The primary objective is to restrict the magnitude of the weights, and thus $\Omega(\mathbf{w})$ is introduced to prevent them from becoming excessively large. The term $\lambda$ is a predefined hyperparameter that controls the trade-off, encouraging the model to favor smaller weights.

**$\mathbf{L}_1$ Regularization:** This adds the sum of the absolute values of the weights to the loss function,

$$L_1 \text{ penalty: } \Omega(\mathbf{w}) = \sum_i |w_i|.$$

$L_1$ regularization tends to produce sparse models, setting many weights to zero.

**$\mathbf{L}_2$ Regularization:** This adds the sum of the squared weights to the loss function,

$$L_2 \text{ penalty: } \Omega(\mathbf{w}) = \sum_i w_i^2.$$

$L_2$ regularization encourages smaller weights and is often used to prevent large weight values that could lead to overfitting.

### Early Stopping

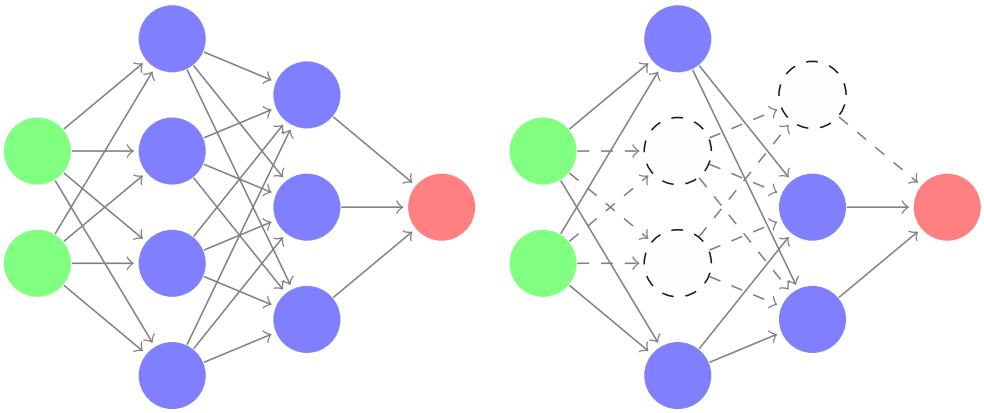
Early stopping is a simple yet effective technique to prevent overfitting [63]. During training, the model's performance is monitored on the validation set, and training is halted when the validation error ceases to improve. This prevents the model from overfitting the training data.

However, optimizing a neural network is not a convex problem, and the process is highly susceptible to getting stuck in local minima. For this reason, it is not always ideal to stop training immediately when the validation error plateaus. Instead, a predefined number of epochs, referred to as *patience*, can be used to allow the model additional time to recover and continue improving. If the validation error does not improve within this patience window, training is stopped, and the model state corresponding to the lowest validation error is restored. This ensures that the final model represents the best-performing state observed during training.

During the model development for the papers in this thesis, all models were trained using early stopping with a patience of 10 epochs.

### Dropout

Dropout is conceptually straightforward but was not proposed until 2014 by Srivastava et al. [64]. After its introduction, it quickly became a default regularization method for training neural networks. It is a stochastic regularization technique where, during training, a fraction of the nodes in a network is randomly deactivated. This means their outputs are set to zero, resulting in a thinned network, as illustrated in Figure 4.7. The random deactivation prevents the model from relying too heavily on specific nodes. It also reduces the risk of co-adaptation between nodes. Co-adaptation occurs when multiple nodes learn to depend on each other in a way that limits their ability to generalize independently.



(a) Complete network (b) Thinned network

**Figure 4.7:** Illustration of dropout in neural networks. (a) The complete network, where all nodes and connections are active. (b) The thinned network, where dropout is applied, randomly deactivating certain nodes and connections. Active nodes are shown in solid colors, while dropped-out nodes and connections are shown as dashed circles and lines.

Dropout has a similar effect to training an ensemble of models. As the dropout is re-sampled for every mini-batch it effectively evaluate numerous different architectures during training. However, unlike explicit ensembles, dropout achieves this in a computationally efficient manner. It avoids the need to train separate models individually.

During training, the gradient for a dropped node is not updated. When the model is used for inference, dropout is turned off, and all nodes in the network remain active. If a model is trained with a dropout probability $p$, only $\sim (1-p)$ of the nodes remain active at a time. Naturally, when all nodes are active during inference, the expected output from each node differs from its output during training. To ensure consistency in the expected output, the weights $\mathbf{w}$ for each node are scaled by the probability of the node being retained during training, i.e., $1-p$, as described by the equation below:

$$\mathbf{w}_{inferens} = (1 - p)\mathbf{w}_{train}. \tag{4.19}$$

**Batch Normalization**

Batch normalization normalizes the inputs to each layer of a network by adjusting and scaling them based on the mean and variance computed within mini-batches during training [62]. For each mini-batch, the input activations are normalized to have zero mean and unit variance, followed by a learned linear transformation using scale and shift parameters. Due to its simplicity and elegance, the algorithm for performing batch normalization per mini-batch is included in Algorithm 1. $\gamma$ is the scale parameter, controlling the amplitude of the normalized activations, while $\beta$ serves as the shift parameter, adjusting their mean.

The primary benefit of applying batch normalization is the reduction of a phenomenon known as internal covariate shift. Internal covariate shift refers to the change in the distribution of inputs to a layer during training, which can slow down convergence and make learning less efficient. By stabilizing these input distributions, batch normalization helps the model learn more effectively.

Additionally, batch normalization serves as a regularizer by introducing noise through the computation of batch statistics for each mini-batch. This noise reduces the model's reliance on specific nodes, similar to the effect of dropout, and improves generalization. During inference, the mean and variance are replaced with fixed values computed over the entire training set to ensure consistency.

In this thesis, batch normalization was applied specifically to the convolutional layers in the CNN models.

**Algorithm 1:** Batch normalizing transform, applied to activation $a$ over a mini-batch. Adapted from [62].

**Input:** Values of $a$ over a mini-batch: $\mathcal{B} = \{a_1, \ldots, a_m\}$.
Parameters to be learned: $\gamma, \beta$.
**Output:** $\{y_i = \mathrm{BN}_{\gamma,\beta}(a_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} a_i \qquad \textbf{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(a_i - \mu_{\mathcal{B}})^2 \qquad \textbf{// mini-batch variance}$$

$$\hat{a}_i \leftarrow \frac{a_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \textbf{// normalize}$$

$$y_i \leftarrow \gamma \hat{a}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(a_i) \qquad \textbf{// scale and shift}$$

## 4.4 Data Partitioning

As previously mentioned, the end goal of any machine learning model is to generalize, meaning that it can perform well on data it has not been trained on. To avoid overfitting, the dataset is divided or partitioned into separate subsets, commonly referred to as the training set, validation set, and test set. This method of partitioning is typically referred to as in-sample evaluation, where all subsets are drawn from the same dataset. An illustration of this process is shown in Figure 4.8.

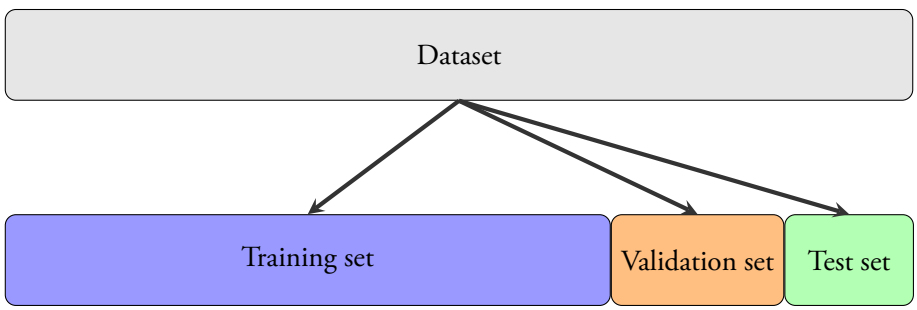



**Figure 4.8:** Illustration of dataset in-sample partitioning into training, validation, and test sets. The dataset is divided into three subsets: the training set, used to optimize model parameters, the validation set, used for model selection and hyperparameter tuning, and the test set, used to evaluate the final model performance.

The training set is used to optimize the model parameters by minimizing the loss function during training. Throughout training, the model is continuously evaluated on the validation set, typically after every epoch, to monitor for overfitting. The validation set acts as a guide for model selection and hyperparameter tuning, allowing us to assess the model's ability to generalize to unseen data. The test set, on the other hand, is a completely independent subset used solely to evaluate the final performance of the model after training is completed.

The best-performing model is often chosen based on its performance on the validation set. Importantly, the test set remains untouched during the training and validation phases. It is only used once, after the final model has been selected, to provide an unbiased estimate of the model's generalization error.

In an ideal scenario, the test set should be completely out-of-sample, meaning that it must not only contain data the model has not seen in any form during training or validation but also come from a different source or context. In engine research, achieving true out-of-sample testing would preferably involve data collected from a different engine and under operating conditions that differ from those used during training. By testing on entirely new conditions, the evaluation can better reflect how well the model generalizes to other engines or operating scenarios.

**Data-Balancing and Stratified Split**

Since two of the papers written in this thesis involved the classification of knock events, the datasets were inherently imbalanced. Even when operating the engine close to the knock limit, most recorded cycles did not exhibit knock. To address this imbalance, we applied two techniques: data-balancing and stratified splitting.

First, we performed data-balancing by downsampling the more frequent classes. In paper III, this involved downsampling the no-knock and medium-knock examples to match the number of occurrences of the heavy-knock examples. For paper IV, the no-knock class was downsampled to include the same number of examples as the knock class. Data-balancing ensures that each class contributes equally during training, preventing the model from being biased toward the majority class.

An imbalanced dataset can also negatively affect how subsets are created when splitting the data. Random splitting, in such cases, can lead to subsets that do not reflect the overall class distribution, resulting in unfair evaluation or poorly trained models.

After balancing the data, the risk of reintroducing imbalance during the train-validation-test split is reduced. Nevertheless, a stratified split was employed to further ensure that class proportions were preserved across the subsets. A stratified split guarantees that each subset

(training, validation, and test) maintains the same class distribution as the original dataset.

In paper II, a stratified split was also performed to ensure that the subsets contained an equal number of examples from each of the different operating points that were evaluated.

## 4.5 Hyperparameter Optimization

Hyperparameter optimization plays a critical role in training machine learning models, as it directly influences model performance. The process can be formulated as an optimization problem:

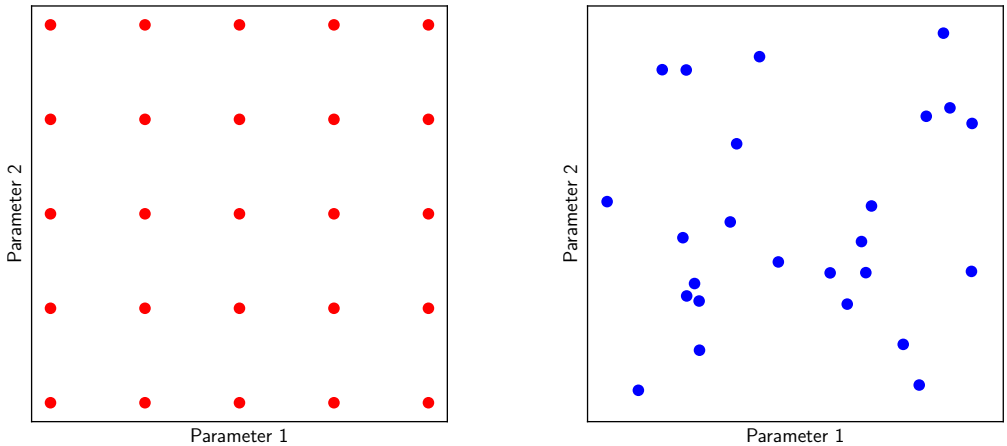
$$\mathbf{h}^* = \arg\min_{\mathbf{h}\in\mathcal{H}} f(\mathbf{h}), \tag{4.20}$$

where $\mathbf{h}$ represents the hyperparameters used to define the network, and $\mathcal{H}$ is the configuration space of the hyperparameters. The objective $f(\mathbf{h})$ quantifies the model's performance, typically evaluated on a validation dataset.

Hyperparameter optimization methods can be categorized into grid search, random search, and more advanced approaches. Grid search explores the hyperparameter space systematically by evaluating all possible combinations of the predefined values. Random search instead samples hyperparameter values randomly within a specified range. Figure 4.9 illustrates the differences between grid search and random search. In grid search, the parameter space is explored in a structured and uniform grid. Random search, on the other hand, samples points sparsely, making it more efficient for high-dimensional spaces.

While both approaches are widely used, their computational cost becomes prohibitive as the number of hyperparameters increases. To address this, Bayesian optimization offers a more efficient alternative by sequentially exploring the hyperparameter space based on prior evaluations.

For the papers constituting this thesis, hyperparameter tuning is based on Bayesian optimization. The Optuna framework is used to tune the hyperparameters of the machine learning models [65]. Optuna is a powerful and flexible Python library that automates hyperparameter search through Bayesian methods.




(a) Uniform grid (b) Random grid

**Figure 4.9:** Comparison of uniform grid and random grid sampling. The left figure shows a uniform grid, where sampling points are evenly distributed across the parameter space. The right figure illustrates a random grid, where sampling points are distributed randomly. The uniform grid provides structured coverage but may miss critical areas, while the random grid increases the likelihood of sampling diverse regions in the parameter space.

### 4.5.1 Bayesian Optimization with Optuna

Bayesian optimization is an iterative, derivative-free approach that automatically identifies the optimal hyperparameter configuration by minimizing an objective function $f(\mathbf{h})$. It achieves this by constructing a probabilistic model of the objective function, which guides the search process. The model is used to select the next set of hyperparameters to evaluate. The core idea is to balance exploration, which involves sampling areas of uncertainty, and exploitation, which focuses on sampling areas with promising results. Optuna achieves this by constructing an acquisition function which directs the search toward optimal hyperparameters.

In Optuna, the Tree-structured Parzen Estimator (TPE) algorithm [66] is employed to perform Bayesian optimization. The TPE algorithm models the objective function using two probability density functions: one for hyperparameter values that yield the best results, $l(\mathbf{h})$, and another for all other hyperparameter values, $g(\mathbf{h})$. The acquisition function is constructed as a ratio of these two densities to identify promising regions in the search space:

$$\mathrm{EI}(\mathbf{h}) \propto \frac{l(\mathbf{h})}{g(\mathbf{h})}.$$

Here, $\mathrm{EI}(\mathbf{h})$ represents the expected improvement, which quantifies the benefit of evaluating a particular set of hyperparameters $\mathbf{h}$.

The TPE algorithm iteratively samples new hyperparameter values based on this ratio. Regions with higher $l(\mathbf{h})/g(\mathbf{h})$ are prioritized, as they are more likely to improve the model's performance.

The TPE algorithm is significantly more efficient than random search, particularly in large and complex search spaces [67]. By adaptively refining the search based on previous evaluations, TPE consistently identifies better-performing configurations in fewer iterations. In earlier work, Bergstra and Bengio [68] demonstrated that random search itself outperforms grid search for hyperparameter optimization, especially when only a subset of the hyperparameters significantly influences model performance. Unlike grid search, which allocates resources evenly across all dimensions, random search provide better coverage of relevant regions in high-dimensional spaces due to its stochastic sampling nature. Building on this, the TPE algorithm further improves efficiency by prioritizing exploration of promising regions, making it an attractive choice for optimizing machine learning models with high-dimensional hyperparameter spaces.

# Engine knock

Engine knock is an abnormal combustion phenomenon that occurs when the end-gas region of the air-fuel mixture, situated ahead of the propagating flame, autoignites prematurely [69]. The bottom sequence in Figure 5.1 illustrates the presence of auto-ignition in the end-gas region resulting in knocking combustion, and the top sequence a normal combustion. The uncontrolled ignition leads to fast pressure rise followed by intense pressure waves within the cylinder. The rapid and repetitive nature of these pressure oscillations places significant stress on engine components, diminishes efficiency, and may cause severe mechanical damage.

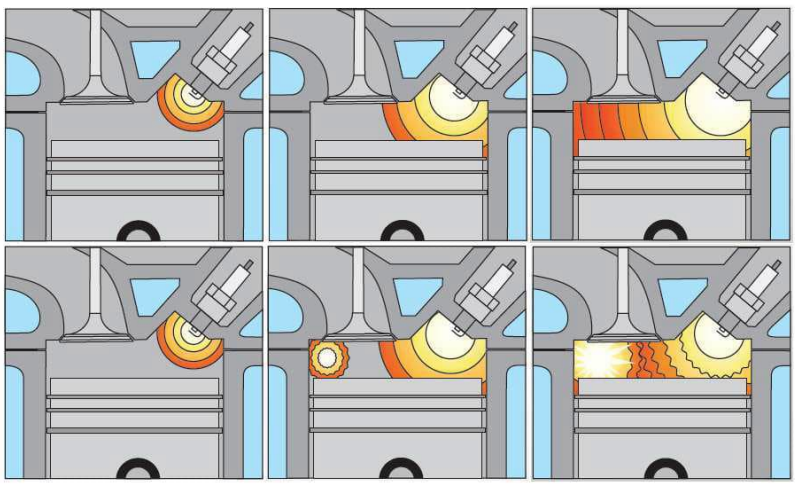


**Figure 5.1:** Comparison of normal SI combustion (top) and knocking combustion (bottom) [70].

Knock is, in essence, an acoustic event, and the characteristics of the pressure waves are influenced by the cylinder's geometry, combustion gas properties, and its associated acoustic resonant modes [71]. These modes represent distinct wave patterns within the combustion chamber, most commonly described as the circumferential, radial, and axial modes,

depicted in Figure 5.2. The axial mode frequencies depend on crank angle, while the circumferential and radial modes do not [72]. Furthermore, as depicted in Figure 5.2, the frequencies of the pressure waves increase with each successive mode, with the lowest frequency being around 6 kHz in the first circumferential mode [73]. The first circumferential mode typically dominates the energy distribution and is responsible for the hallmark "knocking" sound that indicates knocking combustion. The interested reader can find a detailed visualization of the combustion chamber showing different modes in the work by He et al. [74].

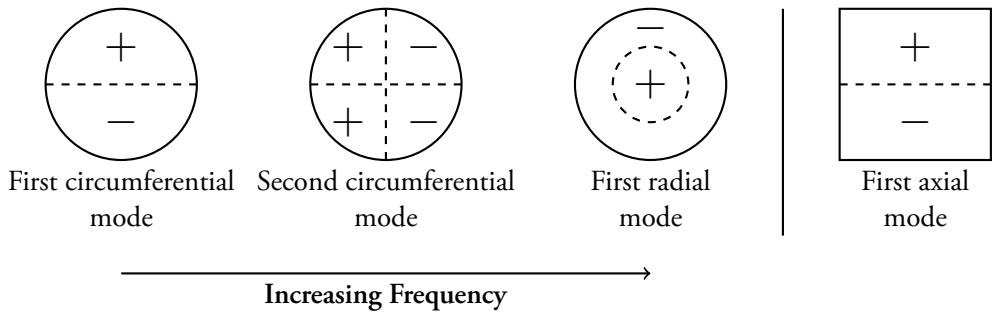



Figure 5.2: Visualization of the acoustic modes. Adapted from [71] and [73].

## 5.1 Knock Measurements

During knock, pressure waves propagate and interact within the engine structure, generating vibrations that can be measured. In vehicle applications, one or more piezoelectric accelerometers, often referred to as vibration sensors or knock sensors, are mounted on the engine body to measure these vibrations. The signals are transmitted to the ECU, where they are analyzed to evaluate their intensity during the timeframe when knock is expected.

To study knock or develop engine control strategies, it is essential to have a reference signal that accurately quantifies knock intensity. This is typically achieved using in-cylinder pressure measurements obtained from a pressure sensor mounted in the cylinder head. A common approach for estimating knock intensity involves calculating the amplitude of the pressure oscillations. These oscillations are isolated by applying a band-pass filter to the in-cylinder pressure signal, centered around the frequency of the first circumferential mode. In the papers included in this thesis, the Maximum Amplitude of Pressure Oscillations (MAPO) was used as the reference for knock intensity. MAPO is calculated by taking the maximum value of the absolute band-pass filtered in-cylinder pressure, as shown below:

$$\text{MAPO} = \max(|p_{bandpass}|). \tag{5.1}$$

A similar approach can be applied to estimate knock intensity from the vibration sensor. However, instead of using a single point estimate, an integration, or more precisely a summation since the signal is discrete, is typically used. In paper IV, the knock intensity calculated in this manner based on the vibration sensor signal is denoted $KI_v$ and is defined as:

$$KI_v = \sum |v_{bandpass}|. \tag{5.2}$$

The rapid increase in pressure and the resulting pressure oscillations during knock lead to fluctuations in the temperature of the burned gas. Since the ionization process is highly dependent on temperature, these fluctuations cause variations in the electrical conductivity of the gas, which, in turn, affect the ion current measurements. Thus, a knock intensity based on the ion current is also derived and denoted $KI_i$. This knock intensity is defined in the same fashion as $KI_v$ and is defined as:

$$KI_i = \sum |i_{bandpass}|. \tag{5.3}$$

## 5.2 Signal Characteristics

Figure 5.3 illustrates the in-cylinder pressure, ion current, and vibration sensor signals and their respective band-pass filtered versions during a normal combustion cycle and a knocking cycle for each case. It is important to note that the scaling of the signals has been adjusted to ensure the figure is informative and does not represent actual magnitudes. Additionally, the phasing of all signals is consistent across the figure.

For the in-cylinder pressure, the characteristic oscillations caused by knocking are clearly visible in the raw signal. These oscillations are even more distinct in the band-pass filtered version, making it straightforward to identify the knocking cycle.

In contrast, the ringing in the ion current signal during knock is not as apparent in the raw signal. However, when the band-pass filter is applied, the oscillatory behavior becomes visible, albeit not as pronounced as in the pressure or vibration sensor signals. Furthermore, the knocking cycle shows a higher overall amplitude and a more rapid increase in the ion current signal compared to the normal cycle. This suggests that machine learning could be a valuable tool for knock detection using ion current signals, as it might be able to leverage features not directly visible to human observation but embedded in the signal.

Lastly, the vibration sensor signal shows a significantly higher amplitude during the knocking cycle. This difference is clearly visible in both the raw signal and the band-pass filtered

version, making it an effective indicator for identifying knock events.

It should be noted that the knocking cycle had a MAPO of 3.4, which could be considered a heavy knock, while the normal combustion cycle had a MAPO of 0.05, which is a very low value. Despite this, there is still considerable signal amplitude in the band-pass filtered version of the vibration sensor signal. This highlights one of the drawbacks of these types of sensors: the susceptibility to mechanical noise. The engine's many moving parts generate vibrations, which can interfere with the sensor signals. Depending on the sensor placement, this mechanical noise can at times degrade the knock detection performance for certain cylinders, to the point where it is barely better than random guessing, as discussed in [22].

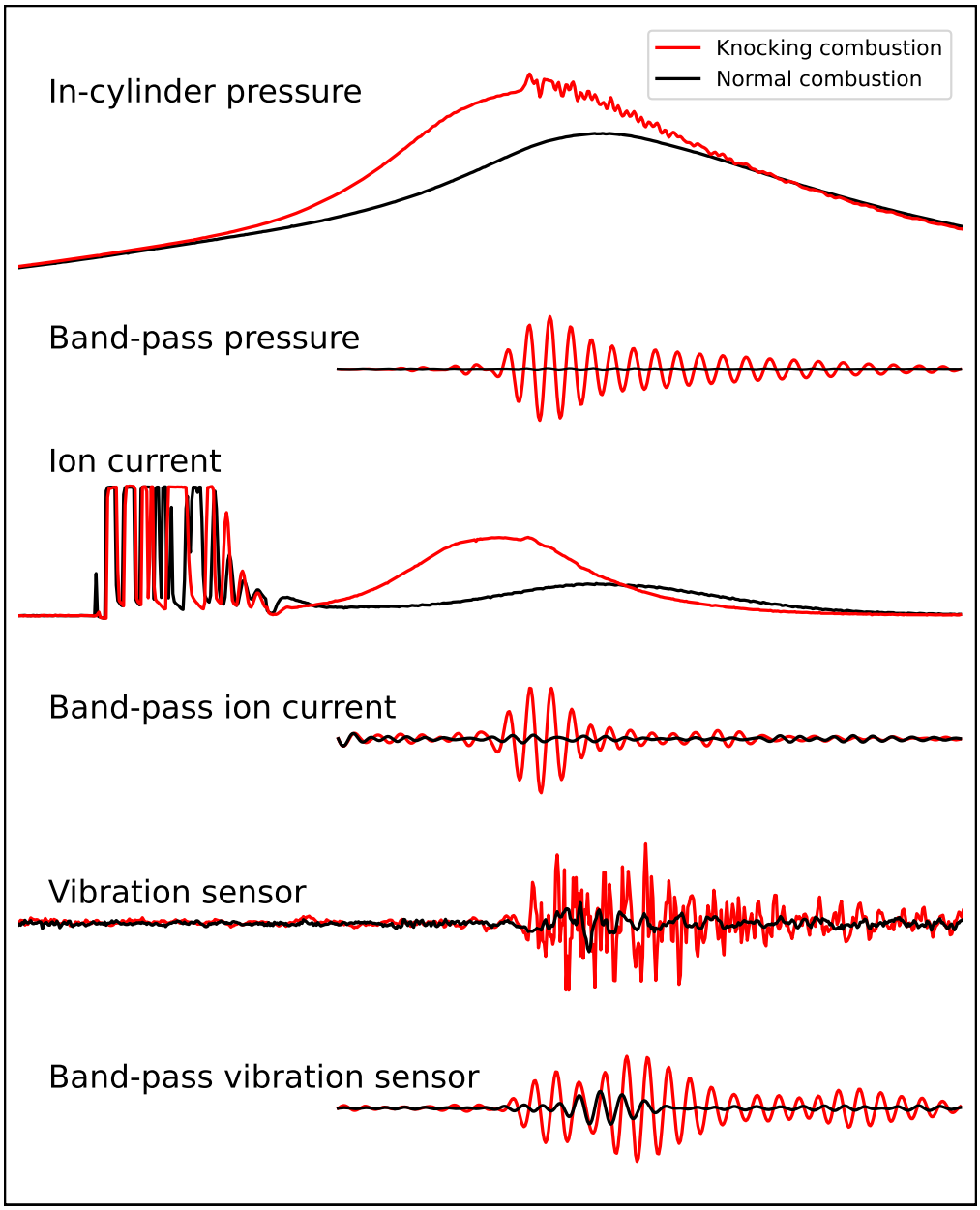



**Figure 5.3:** Comparison of signals during normal combustion (black) and knocking combustion (red). The figure shows in-cylinder pressure, band-pass filtered in-cylinder pressure, ion current, band-pass filtered ion current, vibration sensor signals, and band-pass filtered vibration sensor signals. The knocking cycle had a MAPO of 3.4, while the normal combustion cycle had a MAPO of 0.05. Note: The scaling of the signals has been adjusted for visualization purposes and does not represent actual magnitudes.

## 5.3 Knock Classification and Control

The previous sections presented one method to calculate the "true" knock intensity, MAPO, from the in-cylinder pressure, and two methods to estimate the knock intensity from the vibration sensor ($KI_v$) and the ion current ($KI_i$). However, the question remains: how do we go from knock intensities to classifying knock? The process is quite simple: knock is classified by defining a threshold. Values above the threshold are classified as knocking, while values below are classified as normal combustion. This approach is used to label data as either normal or knocking combustion based on the selected metric from the in-cylinder pressure. It is also employed for online knock classification, typically relying on the vibration sensor signal and a knock intensity estimation like $KI_v$.

In the experiments conducted for this thesis, the MAPO threshold for labeling the experimental data was set at 0.4. This threshold was determined after inspecting the pressure traces and identifying an appropriate value where the characteristic pressure oscillations disappeared. One significant challenge with knock lies in the fact that it is not strictly a binary phenomenon but rather exists along a spectrum of intensity. For instance, is there truly a meaningful difference between a MAPO of 0.39 and 0.41?

The most common approach to knock control is to retard the spark timing when knock is detected. This adjustment shifts the combustion phase later in the cycle, leading to lower in-cylinder pressure and temperatures, thereby avoiding knock. The spark timing is then progressively advanced towards the knock limit again, and when knock inevitably occurs, the process repeats. As a result, every time knock is falsely detected, engine efficiency is unnecessarily sacrificed because the spark timing adjustment was not required. Conversely, if a knocking cycle goes undetected, the engine may suffer severe damage.

Thus, choosing the knock detection threshold is a critical but challenging task. Moreover, to address the previously stated question, there are no practical differences between a MAPO of 0.39 and 0.41, making the chosen threshold inherently arbitrary. As a result, and because no knock detection algorithm is perfect, the threshold is typically set conservatively to ensure safety. This emphasizes the necessity of developing more accurate and reliable knock detection methods. Such improvements are essential for reducing unnecessary control actions and improving engine efficiency, particularly at high loads where the engine is more prone to knock.

# Discussion

This chapter builds upon the discussions presented in the included papers. The aim is to address topics that may not have been fully explored in the papers.

## 6.1 The Use of EGR in Paper I

The proverbial elephant in the room is the use of EGR in paper I to modify the gas composition, rather than altering the natural gas composition. This was, in the end, an unfortunate but necessary decision. The primary reason was that the lab was not equipped to fuel a fully operational six-cylinder heavy-duty engine of this size at higher loads for extended periods using gas bottles. For the same reason, it was not possible to obtain accurate fuel measurements, as the composition of the natural gas supplied by the municipality's pipeline varied.

The thermal gas flow meter used to measure the flow of natural gas had to be modified with an ever-changing conversion factor to account for the varying composition. However, the results were never reliable enough. This led to forgoing efficiency measurements and instead presenting the results in terms of indicated mean effective pressure.

While adding EGR was not part of the original plan, it caused a delayed combustion phase that reduced the engine's power, ultimately making it effective in demonstrating the concept we set out to prove. Through closed-loop control of the combustion phase, achieved by modifying the spark timing based on the peak pressure location estimated from the ion current, the combustion phase was effectively restored to the nominal phase, recovering a significant portion of the power lost. Furthermore, enabling the closed loop control also led to an increased combustion stability.

## 6.2 Data Partitioning: Choosing Train/Validation/Test Splits Over K-Fold Cross-Validation

K-fold cross-validation is a commonly used technique in machine learning to evaluate model performance. It begins with an initial split of the dataset into a train and test set, similar to the train/validation/test split. The train set is then further divided into $k$ subsets, or folds, and the model is iteratively trained on $k-1$ folds while validating on the remaining fold. This process is repeated $k$ times, ensuring each fold is used for validation once. By averaging the results across all folds, this method provides a more robust assessment of a model's generalizability.

While K-fold cross-validation offers significant benefits, its importance diminishes as the size of the dataset increases. With a sufficiently large dataset, the train/validation/test split is often adequate to capture the variability within the data and provides reliable performance metrics. The larger the dataset, the less likely it is that splitting the data once will introduce biases, as the subsets are more likely to represent the overall data distribution. Additionally, in the work presented, the dataset was balanced, and the train/validation/test split was performed in a stratified fashion. This further reduced the risk of introducing biases by preserving the distribution of classes across all subsets.

A train/validation/test split was used instead of K-fold cross-validation, primarily due to practical considerations. For Papers III and IV, training the machine learning models was computationally intensive, requiring significant time and resources. The iterative process of training and validating models across multiple folds would have been prohibitively time-consuming, given the complexity of the architectures and the size of the datasets. As the dataset was sufficiently large to ensure representative splits, the added computational cost of K-fold cross-validation was deemed unnecessary.

By using a train/validation/test split, the approach balanced computational efficiency with reliable performance evaluation, aligning with the practical constraints of the included work.

## 6.3 Generalizability of Models to Other Operating Points and Engines

The generalizability of the models to other operating points on the same engine remains an open question that requires further verification. The virtual pressure sensor presented in Paper II was trained on a very coarse grid of operating points, while the knock detection models were only trained on data collected from two operating points. Machine learning

models perform best when trained on as much diverse data as possible to capture the variability inherent in the system. Fortunately, this is not necessarily a limiting factor, as a vast amount of data can be recorded from an engine test cell, ensuring sufficient coverage across different operating conditions. However, the challenge lies in the time and computational resources required to train models on such large datasets.

Extending the generalizability to different engine models introduces additional complexities. It is likely that the models would need to be retrained using data specific to the target engine model, as differences in design, operating characteristics, and sensor configurations could significantly impact performance. That said, it is not necessary to reinvent the wheel. Once a suitable model architecture has been developed, its structure can serve as a conceptual foundation for further training.

An effective approach to expedite this process is transfer learning, which allows a model trained on one dataset to be fine-tuned on another. This avoids the need to cold-start training from scratch and can significantly reduce the computational effort required while leveraging existing knowledge embedded in the original model. By employing transfer learning, the adaptability of these models to new engines or operating points can be enhanced, ensuring efficient deployment.

## 6.4 Estimating Knock Intensity Using Machine Learning

This thesis has primarily focused on using machine learning to classify knock events, i.e., constructing classification models. While the probability outputs from such classifiers can be interpreted as a measure of likelihood, they do not inherently capture the physical magnitude of knock intensity. Consequently, extending ML methodologies to estimate knock intensity as a continuous variable presents an opportunity for providing a richer characterization of knocking events.

Estimating knock intensity could offer significant advantages over probability-based classification by providing a more detailed representation of knocking events. Knock intensity, often derived from metrics such as MAPO, quantifies the severity of knock rather than its likelihood. Therefore, estimations of knock intensity would align more closely with the underlying physics of combustion. Precise intensity estimates would allow for finer control strategies, enabling parameters such as spark timing to be adjusted proportionally to the severity of knock. This approach could optimize engine performance by minimizing unnecessary interventions during light knock events while applying stronger corrective actions in cases of severe knock. Real-time intensity predictions could further enhance engine control by enabling dynamic thresholding and adaptive strategies designed to account for operating conditions.

## 6.5 Correlation between Knock Indicators and MAPO

This section discusses the correlation between the two knock indicators, $KI_v$ and $KI_i$, and MAPO, with an emphasis on $KI_i$. Previous research has highlighted that the placement of pressure or ion current sensors significantly affects the robustness of knock detection. Due to the characteristics of the acoustic modes of knock, as discussed in chapter 5, placing the sensor at the center of the cylinder head is particularly disadvantageous. Such a position is insensitive to all circumferential modes [75] and generally exhibits the lowest sensitivity to the various acoustic modes [71]. Consequently, a central position typically results in a lower correlation between $KI_i$ and MAPO.

Nonetheless, the study by Daniels et al. [76] reported a high correlation between $KI_i$ and their pressure-based knock indicator, even with a centrally positioned spark plug measuring the ion current. It should be noted, however, that their spark plug was not truly at the cylinder head's center but slightly off-center due to the size differences between the intake and exhaust valves. Furthermore, their chosen pressure-based knock reference was not MAPO but rather a method similar to how we calculated $KI_v$ and $KI_i$, specifically the sum of the absolute value of the band-pass filtered in-cylinder pressure. For clarity, we will refer to this alternative knock reference as $KI_p$. Their results showed that the correlation between $KI_i$ and $KI_p$ remained consistently high, around 0.83, regardless of engine speed. In contrast, the correlation between $KI_v$ and $KI_p$ decreased significantly at higher engine speeds due to increased noise levels.

In our case, the correlation between $KI_i$ and MAPO is consistently lower than the correlation between $KI_v$ and MAPO. Despite this, as shown in Paper IV, a Logistic Regression (LR) model based solely on $KI_i$ did not always underperform compared to an LR model based solely on $KI_v$. One plausible explanation is the central positioning of the spark plug in our setup, which probably contributed to the observed results. Another significant factor, which we realized after completing Papers III and IV, is that the secondary coil, through which the ion current passes, acts as a low-pass filter at approximately 6 kHz. This filtering substantially impacts the band-pass filtered signal used to calculate $KI_i$, effectively cutting out half of the frequency band.

# Conclusion and Future Work

This thesis has demonstrated the potential of ion current measurements and machine learning to enhance combustion diagnostics and control in spark-ignition engines. By integrating experimental investigations with machine learning approaches, this work addresses key challenges in improving engine performance, knock detection, and real-time control, while reducing reliance on costly in-cylinder pressure sensors.

Ion current measurements have proven to be a reliable and cylinder-agnostic diagnostic tool, capable of capturing critical combustion parameters. The use of ion current, particularly in combination with machine learning models, offers a cost-effective alternative to traditional in-cylinder pressure sensors. This was exemplified by the successful use of a feed-forward neural network to predict in-cylinder pressure traces from ion current signals, demonstrating the feasibility of virtual pressure sensing. Such advancements highlight the potential of ion current-based diagnostics as a compelling choice for real-time engine monitoring and control.

The application of convolutional neural networks to knock detection illustrates the power of machine learning in uncovering spatial and temporal patterns within ion current signals. CNNs trained on ion current data achieved performance levels comparable to traditional knock detection methods, and the integration of knock indicators derived from both ion current ($KI_i$) and vibration sensors ($KI_v$) further enhanced detection robustness. Logistic regression models, while less sophisticated, provided interpretable insights and demonstrated that simpler models can still play a valuable role in engine diagnostics, particularly in applications with constrained computational resources.

Additionally, this thesis has demonstrated the feasibility of closed-loop control systems based on ion current-derived PPL estimates. Such systems successfully mitigated performance losses caused by variations in fuel composition, recovering up to 2% of engine power in scenarios involving increased EGR. This demonstrates how ion current-based systems to maintain optimal combustion phasing, leading to improved efficiency and reducing emissions in real-world applications.

Despite these advances, several challenges remain. The sensitivity of vibration-based knock indicators to mechanical noise limits their reliability in certain cylinders, emphasizing the need for robust sensor fusion strategies. However, since such strategies also incorporate data from the vibration sensor, they too would be affected by the noise. A robust sensor fusion strategy would adapt by lowering the importance of vibration sensor information for cylinders with significant noise interference. In the context of the dual-input CNN, this could be achieved by adding information related to which cylinder is being analyzed, allowing the model to dynamically reduce the weight assigned to the vibration sensor for cylinders with high noise levels. Furthermore, the computational complexity of CNNs presents a barrier to their real-time application, though the demonstrated benefits of these models underscore the importance of developing more efficient architectures.

While this work primarily focuses on spark-ignition engines fueled by natural gas, the findings have broader implications. The methodologies and models presented here are adaptable to alternative fuels and other engine types, paving the way for more sustainable and efficient internal combustion engines. In summary, this thesis establishes a strong foundation for the integration of ion current diagnostics and machine learning in advanced engine control systems.

Building on the findings of this thesis, there are many interesting avenues for future work. Some examples include:

- **Integration of Additional Sensors:** Incorporate data from sensors such as exhaust gas temperature to enhance model robustness and accuracy.
- **Exploration of Alternative Fuels:** Extend the study to engines using other fuels to assess the adaptability of the proposed methods, especially in non-stoichiometric combustion regimes.
- **Optimized Neural Network Architectures:** Investigate lighter and more computationally efficient models tailored for real-time applications. Potential strategies include downsampling input signals, reducing the number of convolutional layers, or exploring recurrent neural networks for capturing sequential patterns.
- **Validation Across Broader Operating Conditions:** Test the models on larger datasets encompassing a wider range of engine speeds, loads, and cylinder conditions. Such validation would provide deeper insights into the generalizability and reliability of the proposed methods.
- **Estimating Knock Intensity:** Develop ML models to estimate knock intensity as a continuous variable instead of probabilities.

- **Development of Real-Time Control Strategies:** Implement the knock detection models in closed-loop engine control systems. This would enable the models to actively adjust engine parameters in real-time, optimizing performance and mitigating knock under varying operating conditions.

- **Real-Time Model Optimization:** Develop a framework that enables continuous improvement of the model in real-time. While implementing such a system in a production vehicle presents significant challenges due to limited computational resources and the lack of in-cylinder pressure sensors, it is feasible in a laboratory setting. In this approach, a cloud-based model could receive data from the lab in real-time, enabling continuous training without requiring intermediate storage for the vast amount of data generated during experiments. This eliminates the need for local data handling, as the data is directly ingested by the model, significantly streamlining the overall process. On-road vehicles could then periodically update their models by downloading the improved version from the cloud. This method ensures that deployed models remain up-to-date while reducing the logistical burden associated with storing and transferring large datasets.

# References


[1] HAAGEN-SMIT, A. J. "Chemistry and Physiology of Los Angeles Smog". *Industrial & Engineering Chemistry*, June 1952, vol. 44, no. 6, p. 1342–1346, ISSN 0019-7866, doi:10.1021/ie50510a045.

[2] IPCC, . *Global Warming of 1.5° C: IPCC Special Report on Impacts of Global Warming of 1.5° C above Pre-industrial Levels in Context of Strengthening Response to Climate Change, Sustainable Development, and Efforts to Eradicate Poverty*. 1 ed. Cambridge University Press, 2018. doi:10.1017/9781009157940, ISBN 978-1-009-15794-0 978-1-009-15795-7.

[3] EUROPEAN COMMISSION, . "Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions: The European Green Deal". Communication COM(2019) 640 final, European Commission, Brussel, Dec. 2019. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2019%3A640%3AFIN.

[4] EUROPEAN COMMISSION, . "Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee, and the Committee of the Regions. 'Fit for 55': Delivering the EU's 2030 Climate Target on the Way to Climate Neutrality". Communication COM(2021) 550 final, European Commission, Brussels, June 2022. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021DC0550.

[5] EUROPEAN COMMISSION, . "Regulation (EU) 2023/851 of the European Parliament and Council of 19 April 2023 Amending Regulation (EU) 2019/631 Regarding CO2 Emission Performance Standards for New Passenger Cars and Light Commercial Vehicles". Technical report, European Parliament, Apr. 2023, text with EEA relevance.

[6] EUROPEAN ENVIRONMENT AGENCY, . "Greenhouse Gases Viewer - Data Viewers". Aug. 2024. Available from: https://www.eea.europa.eu/en/analysis/maps-and-charts/greenhouse-gases-viewer-data-viewers [Accessed 2024-12-23].

[7] ALGAYYIM, S. J. M., SALEH, K., WANDEL, A. P., *et al.* "Influence of Natural Gas and Hydrogen Properties on Internal Combustion Engine Performance, Combustion, and Emissions: A Review". *Fuel*, Apr. 2024, vol. 362, p. 130844, ISSN 0016-2361, doi:10.1016/j.fuel.2023.130844.

[8] KAIADI, M., TUNESTÅL, P., and JOHANSSON, B. "Using Hythane as a Fuel in a 6-Cylinder Stoichiometric Natural-gas Engine". *SAE International Journal of Fuels and Lubricants*, June 2009, vol. 2, no. 1, p. 932–939, doi:10.4271/2009-01-1950.

[9] KAIADI, M., TUNESTÅL, P., and JOHANSSON, B. "How Hythane with 25% Hydrogen Can Affect the Combustion in a 6-Cylinder Natural-gas Engine". *SAE International Journal of Fuels and Lubricants*, May 2010, vol. 3, no. 2, p. 47–59, doi:10.4271/2010-01-1466.

[10] ERIKSSON, L., NIELSEN, L., and GLAVENIUS, M. "Closed Loop Ignition Control by Ionization Current Interpretation" SAE Technical Paper 970854 (1997). doi:10.4271/970854.

[11] ZHANG, Z., ZHAO, F., LI, L., *et al.* "Closed-Loop Control of Low Temperature Combustion Employing Ion Current Detecting Technology" SAE Technical Paper 2014-01-1362 (2014). doi:10.4271/2014-01-1362.

[12] CAVINA, N., ROJO, N., BUSINARO, A., *et al.* "Comparison between Pressure- and Ion-Current-Based Closed-Loop Combustion Control Performance". *SAE International Journal of Engines*, Apr. 2019, vol. 12, no. 2, p. 03–12–02–0016, ISSN 1946-3944, doi:10.4271/03-12-02-0016.

[13] SAITZKOFF, A., REINMANN, R., MAUSS, F., *et al.* "In-Cylinder Pressure Measurements Using the Spark Plug as an Ionization Sensor" SAE Technical Paper 970857 (1997). doi:10.4271/970857.

[14] SAITZKOFF, A., REINMANN, R., BERGLIND, T., *et al.* "An Ionization Equilibrium Analysis of the Spark Plug as an Ionization Sensor" SAE Technical Paper 960337 (1996). doi:10.4271/960337.

[15] GAZIS, A., PANOUSAKIS, D., CHEN, R., *et al.* "Computationally Inexpensive Methods of Ion Current Signal Manipulation for Predicting the Characteristics of Engine In-Cylinder Pressure". *International Journal of Engine Research*, June 2006, vol. 7, no. 3, p. 271–282, ISSN 1468-0874, 2041-3149, doi:10.1243/14680874JER04005.

[16] PANOUSAKIS, D., GAZIS, A., PATTERSON, J., *et al.* "Analysis of SI Combustion Diagnostics Methods Using Ion-Current Sensing Techniques". In *SAE 2006 World Congress & Exhibition*, Apr. 2006, p. 2006–01–1345, doi:10.4271/2006-01-1345.

[17] GAO, Z., DENG, Y., WEN, Y., *et al.* "Deep-Learning Based in-Cylinder Pressure Modeling and Resolution of Ion Current Signals". *Fuel*, Dec. 2020, vol. 282, p. 118722, ISSN 00162361, doi:10.1016/j.fuel.2020.118722.

[18] TONG, S., YANG, Z., HE, X., *et al.* "Knock and Pre-Ignition Detection Using Ion Current Signal on a Boosted Gasoline Engine". In *WCX™ 17: SAE World Congress Experience*, Mar. 2017, p. 2017–01–0792, doi:10.4271/2017-01-0792.

[19] GIGLIO, V., POLICE, G., RISPOLI, N., *et al.* "Experimental Investigation on the Use of Ion Current on SI Engines for Knock Detection". In *SAE 2009 Powertrains Fuels and Lubricants Meeting*, Nov. 2009, p. 2009–01–2745, doi:10.4271/2009-01-2745.

[20] AUZINS, J., JOHANSSON, H., and NYTOMT, J. "Ion-Gap Sense in Misfire Detection, Knock and Engine Control" SAE Technical Paper 950004 (1995). doi:10.4271/950004.

[21] ZHANG, Z., SAIKI, N., TODA, H., *et al.* "Detection of Knocking by Wavelet Transform Using Ion Current". In *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*, Kaohsiung, Taiwan: IEEE. Dec. 2009, ISBN 978-1-4244-5543-0, p. 1566–1569, doi:10.1109/ICICIC.2009.173.

[22] ÄNGEBY, J., JOHNSSON, A., and HELLSTRÖM, K. "Knock Detection Using Multiple Indicators and a Classification Approach". *IFAC-PapersOnLine*, 2018, vol. 51, no. 31, p. 297–302, ISSN 24058963, doi:10.1016/j.ifacol.2018.10.063.

[23] ABADI, M., BARHAM, P., CHEN, J., *et al.* "TensorFlow: A System for Large-Scale Machine Learning". In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, USA: USENIX Association. Nov. 2016, ISBN 978-1-931971-33-1, p. 265–283.

[24] ABADI, M., AGARWAL, A., BARHAM, P., *et al.* "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". *arXiv*: 1603.04467, Mar. 2016. Available from: http://arxiv.org/abs/1603.04467.

[25] ANSEL, J., YANG, E., HE, H., *et al.* "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, La Jolla CA USA: ACM. Apr. 2024, ISBN 979-8-4007-0385-0, p. 929–947, doi:10.1145/3620665.3640366.

[26] TUNESTÅL, P. "Self-Tuning Gross Heat Release Computation for Internal Combustion Engines". *Control Engineering Practice*, Apr. 2009, vol. 17, no. 4, p. 518–524, ISSN 09670661, doi:10.1016/j.conengprac.2008.09.012.

[27] GATOWSKI, J. A., BALLES, E. N., CHUN, K. M., *et al.* "Heat Release Analysis of Engine Pressure Data" SAE Technical Paper 841359 (1984). doi:10.4271/841359.

[28] ERIKSSON, L. and NIELSEN, L. "Ionization Current Interpretation for Ignition Control in Internal Combustion Engines". *Control Engineering Practice*, Aug. 1997, vol. 5, no. 8, p. 1107–1113, ISSN 09670661, doi:10.1016/S0967-0661(97)00103-2.

[29] YOSHIYAMA, S., TOMITA, E., and HAMAMOTO, Y. "Fundamental Study on Combustion Diagnostics Using a Spark Plug as Ion Probe" SAE Technical Paper 2000-01-2828 (2000). doi:10.4271/2000-01-2828.

[30] BJÖRNSSON, O. and TUNESTÅL, P. "Virtual Pressure Sensor Based on Ion Current Measurements Using Artificial Neural Networks". In *ASME 2024 ICE Forward Conference*, San Antonio, Texas, USA: American Society of Mechanical Engineers. Oct. 2024, ISBN 978-0-7918-8852-0, p. V001T04A002, doi:10.1115/ICEF2024-140748.

[31] ERMAN, G. "Über Die Fähigkeit Der Flamme, Der Knochen Und Des Luftleeren Raumes, Die Wirkungen Der Voltaischen Säule Zu Leiten". *Annalen der Physik*, 1802, vol. 11, no. 6, p. 147–168.

[32] TUFTS, F. L. "The Phenomena of Ionization in Flame Gases and Vapors". *Physical Review (Series I)*, Apr. 1906, vol. 22, no. 4, p. 193–220, ISSN 1536-6065, doi:10.1103/PhysRevSeriesI.22.193.

[33] CALCOTE, H. "Mechanisms for the Formation of Ions in Flames". *Combustion and Flame*, Dec. 1957, vol. 1, no. 4, p. 385–403, ISSN 00102180, doi:10.1016/0010-2180(57)90001-9.

[34] GREEN, J. and SUGDEN, T. "Some Observations on the Mechanism of Ionization in Flames Containing Hydrocarbons". *Symposium (International) on Combustion*, Jan. 1963, vol. 9, no. 1, p. 607–621, ISSN 00820784, doi:10.1016/S0082-0784(63)80070-3.

[35] CALCOTE, H., KURZIUS, S., and MILLER, W. "Negative and Secondary Ion Formation in Low-Pressure Flames". *Symposium (International) on Combustion*, Jan. 1965, vol. 10, no. 1, p. 605–619, ISSN 00820784, doi:10.1016/S0082-0784(65)80206-5.

[36] CALCOTE, H. "Ion and Electron Profiles in Flames". *Symposium (International) on Combustion*, Jan. 1963, vol. 9, no. 1, p. 622–637, ISSN 00820784, doi:10.1016/S0082-0784(63)80071-5.

[37] GOODINGS, J. M., BOHME, D. K., and SUGDEN, T. M. "Positive Ion Probe of Methane-Oxygen Combustion". *Symposium (International) on Combustion*, Jan. 1977, vol. 16, no. 1, p. 891–902, ISSN 0082-0784, doi:10.1016/S0082-0784(77)80382-2.

[38] PEETERS, J. and VINCKIER, C. "Production of Chemi-Ions and Formation of CH and CH2 Radicals in Methane-Oxygen and Ethylene-Oxygen Flames". *Symposium (International) on Combustion*, Jan. 1975, vol. 15, no. 1, p. 969–977, ISSN 0082-0784, doi:10.1016/S0082-0784(75)80362-6.

[39] HAYHURST, A. N. and JONES, H. R. N. "Chemi-Ionization in Oxyacetylene Flames". *Nature*, Mar. 1982, vol. 296, no. 5852, p. 61–63, ISSN 0028-0836, 1476-4687, doi:10.1038/296061a0.

[40] VINCKIER, C., GARDNER, M. P., and BAYES, K. D. "A Study of Some Primary and Secondary Chemi-Ionization Reactions in Hydrocarbon Oxidations". *Symposium (International) on Combustion*, Jan. 1977, vol. 16, no. 1, p. 881–889, ISSN 0082-0784, doi:10.1016/S0082-0784(77)80381-0.

[41] SCHOFIELD, K. "The Enigmatic Mechanism of the Flame Ionization Detector: Its Overlooked Implications for Fossil Fuel Combustion Modeling". *Progress in Energy and Combustion Science*, June 2008, vol. 34, no. 3, p. 330–350, ISSN 0360-1285, doi:10.1016/j.pecs.2007.08.001.

[42] HOWATSON, A. M. *An Introduction to Gas Discharges*. 2d ed ed. Oxford New York: Pergamon Press, 1976. Applied Electricity and Electronics. ISBN 978-0-08-020574-8.

[43] FRIDMAN, A. *Plasma Chemistry*. 1 ed. Cambridge University Press, May 2008. doi:10.1017/CBO9780511546075, ISBN 978-0-521-84735-3 978-0-511-54607-5 978-1-107-68493-5. Available from: https://www.cambridge.org/core/product/identifier/9780511546075/type/book.

[44] GLASSMAN, I. and YETTER, R. A. *Combustion*. 4th ed ed. Amsterdam Boston: Elsevier, 2008. ISBN 978-0-12-088573-2.

[45] LAVOIE, G. A., HEYWOOD, J. B., and KECK, J. C. "Experimental and Theoretical Study of Nitric Oxide Formation in Internal Combustion Engines". *Combustion Science and Technology*, Feb. 1970, vol. 1, no. 4, p. 313–326, ISSN 0010-2202, 1563-521X, doi:10.1080/00102206908952211.

[46] CAVINA, N., ROJO, N., POGGIO, L., *et al.* "Investigation on Pre-Ignition Combustion Events and Development of Diagnostic Solutions Based on Ion Current Signals". *SAE International Journal of Engines*, 2017, vol. 10, no. 4, p. 1518–1523, ISSN 1946-3936, 26422542. Available from: https://www.jstor.org/stable/26422542.

[47] WANG, J., SHI, J., DENG, J., *et al.* "Misfire and Knock Detection Based on the Ion Current inside a Passive Pre-Chamber of Gasoline Engine". *Fuel*, Mar. 2022, vol. 311, p. 122528, ISSN 00162361, doi:10.1016/j.fuel.2021.122528.

[48] COLLINGS, N. and BRAY, K. "Ionization Sensors for Internal Combustion Engine Diagnostics". *Endeavour*, Jan. 1991, vol. 15, no. 1, p. 10–12, ISSN 0160-9327, doi:10.1016/0160-9327(91)90080-U.

[49] LEE, A. and PYKO, J. S. "Engine Misfire Detection by Ionization Current Monitoring" SAE Technical Paper 950003 (1995). doi:10.4271/950003.

[50] ANDERSON, R. L. "In-Cylinder Measurement of Combustion Characteristics Using Ionization Sensors" SAE Technical Paper 860485 (1986). doi:10.4271/860485.

[51] REINMANN, R., SAITZKOFF, A., and MAUSS, F. "Local Air-Fuel Ratio Measurements Using the Spark Plug as an Ionization Sensor" SAE Technical Paper 970856 (1997). doi:10.4271/970856.

[52] WICKSTROM, N., TAVENIKU, M., LINDE, A., *et al.* "Estimating Pressure Peak Position and Air-Fuel Ratio Using the Ionization Current and Artificial Neural Networks". In *Proceedings of Conference on Intelligent Transportation Systems*, Boston, MA, USA: IEEE. 1997, ISBN 978-0-7803-4269-9, p. 972–977, doi:10.1109/ITSC.1997.660605.

[53] ERIKSSON, L., NIELSEN, L., and NYTOMT, J. "Ignition Control by Ionization Current Interpretation" SAE Technical Paper 960045 (1996). doi:10.4271/960045.

[54] AENGEBY, J. "Closed Loop Control of the Combustion Phase in SI Engines Using Alternative Fuels". *SAE International Journal of Advances and Current Practices in Mobility*, Sept. 2020, vol. 3, no. 1, p. 312–319, ISSN 2641-9645, doi:10.4271/2020-01-2088.

[55] GOODFELLOW, I., BENGIO, Y., and COURVILLE, A. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[56] BISHOP, C. M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006. Information Science and Statistics. ISBN 978-0-387-31073-2.

[57] KINGMA, D. P. and BA, J. "Adam: A Method for Stochastic Optimization". *arXiv*: 1412.6980, Jan. 2017. Available from: http://arxiv.org/abs/1412.6980.

[58] FUKUSHIMA, K. "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position". *Biological Cybernetics*, Apr. 1980, vol. 36, no. 4, p. 193–202, ISSN 1432-0770, doi:10.1007/BF00344251.

[59] LECUN, Y. "Generalization and Network Design Strategies". Techincal Report CRG-TR-89-4, University of Toronto, 1989.

[60] LECUN, Y., BOTTOU, L., BENGIO, Y., *et al.* "Gradient-Based Learning Applied to Document Recognition". *Proceedings of the IEEE*, Nov. 1998, vol. 86, no. 11, p. 2278–2324, ISSN 1558-2256, doi:10.1109/5.726791.

[61] KRIZHEVSKY, A., SUTSKEVER, I., and HINTON, G. E. "ImageNet Classification with Deep Convolutional Neural Networks". *Communications of the ACM*, May 2017, vol. 60, no. 6, p. 84–90, ISSN 0001-0782, 1557-7317, doi:10.1145/3065386.

[62] IOFFE, S. and SZEGEDY, C. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". *arXiv*: 1502.03167, Mar. 2015. Available from: http://arxiv.org/abs/1502.03167.

[63] PRECHELT, L. "Early Stopping — But When?". In MONTAVON, GRÉGOIRE, ORR, GENEVIÈVE B., and MÜLLER, KLAUS-ROBERT (eds.). *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer, 2012. p. 53–67. ISBN 978-3-642-35289-8. doi:10.1007/978-3-642-35289-8_5.

[64] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., *et al.* "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". *Journal of Machine Learning Research*, 2014, vol. 15, no. 56, p. 1929–1958. Available from: http://jmlr.org/papers/v15/srivastava14a.html.

[65] AKIBA, T., SANO, S., YANASE, T., *et al.* "Optuna: A Next-generation Hyperparameter Optimization Framework". In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, New York, NY, USA: Association for Computing Machinery. July 2019, ISBN 978-1-4503-6201-6, p. 2623–2631, doi:10.1145/3292500.3330701.

[66] BERGSTRA, J., BARDENET, R., BENGIO, Y., *et al.* "Algorithms for Hyper-Parameter Optimization". In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, Red Hook, NY, USA: Curran Associates Inc. Dec. 2011, ISBN 978-1-61839-599-3, p. 2546–2554.

[67] BERGSTRA, J., YAMINS, D., and COX, D. "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In *Proceedings of the 30th International Conference on Machine Learning*, PMLR. Feb. 2013, ISSN 1938-7228, p. 115–123. Available from: https://proceedings.mlr.press/v28/bergstra13.html.

[68] BERGSTRA, J. and BENGIO, Y. "Random Search for Hyper-Parameter Optimization". *Journal of Machine Learning Research*, 2012, vol. 13, no. 10, p. 281–305. Available from: http://jmlr.org/papers/v13/bergstra12a.html.

[69] CORRIGAN, D. J., FONTANESI, S., CORRIGAN, D. J., *et al.* "Knock: A Century of Research". *SAE International Journal of Engines*, July 2021, vol. 15, no. 1, p. 57–127, doi:10.4271/03-15-01-0004.

[70] ROBEYN, T. *An Alternative Pressure-Oscillation-Based Octane Rating Method, for Fuels below and beyond 100 RON, to Aid the Development of a Second-Generation Spark-Ignition Biofuel*. Dissertation thesis, Ghent University, Ghent, Belgium, 2024.

[71] SCHOLL, D., DAVIS, C., RUSS, S., *et al.* "The Volume Acoustic Modes of Spark-Ignited Internal Combustion Chambers" SAE Technical Paper 980893 (1998). doi:10.4271/980893.

[72] HEYWOOD, J. *Internal Combustion Engine Fundamentals 2E*. 2 edition ed. New York: McGraw-Hill Education, May 2018. ISBN 978-1-260-11610-6.

[73] HICKLING, R., FELDMAIER, D. A., CHEN, F. H. K., *et al.* "Cavity Resonances in Engine Combustion Chambers and Some Applications". *The Journal of the Acoustical Society of America*, Apr. 1983, vol. 73, no. 4, p. 1170–1178, ISSN 0001-4966, 1520-8524, doi:10.1121/1.389261.

[74] HE, X., QI, Y., WANG, Z., *et al.* "Visualization of the Mode Shapes of Pressure Oscillation in a Cylindrical Cavity". *Combustion Science and Technology*, Oct. 2015, vol. 187, no. 10, p. 1610–1619, ISSN 0010-2202, doi:10.1080/00102202.2015.1038385.

[75] FÖRSTER, J., GÜNTHER, A., KETTERER, M., *et al.* "Ion Current Sensing for Spark Ignition Engines" SAE Technical Paper 1999-01-0204 (1999). doi:10.4271/1999-01-0204.

[76] DANIELS, C. F., ZHU, G. G., and WINKELMAN, J. "Inaudible Knock and Partial-Burn Detection Using In-Cylinder Ionization Signal" SAE Technical Paper 2003-01-3149 (2003). doi:10.4271/2003-01-3149.

# Scientific Publications

Printed by Media-Tryck, Lund 2025 NORDIC SWAN ECOLABEL 3041 0903

Faculty of Engineering (LTH)
Department of Energy Sciences
Division of Combustion Engines

ISBN 978-91-8104-372-3
ISRN LUTMDN/TMHP-25/1179-SE
ISSN 0282-1990