*energies*

MDPI

*Article*

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

**Ola Björnsson * and Per Tunestål**

Department of Energy Sciences, Faculty of Engineering, Lund University, P.O. Box 118, 221 00 Lund, Sweden; per.tunestal@energy.lth.se

* Correspondence: ola.bjornsson@energy.lth.se

**Abstract:** Knock detection is critical for maintaining engine performance and preventing damage in spark-ignition engines. This study explores the use of ion current and knock indicators derived from a vibration sensor ($KI_v$) and ion current ($KI_i$) to improve knock detection accuracy. Traditional threshold-based methods rely on $KI_v$, but they are susceptible to mechanical noise and cylinder variations. In this work, we applied both logistic regression and neural networks, including fully connected (FCNN) and convolutional neural networks (CNN), to classify knock events based on these indicators. The CNN models used ion current as the primary input, with an extended version incorporating both $KI_v$ and $KI_i$ into the fully connected layers. The models were evaluated using area under the curve (AUC) as the primary performance metric. The results show that the CNN model with additional inputs outperformed the other models, achieving a better and more consistent performance across cylinders. The dual-input logistic regression and CNN models demonstrated reduced cylinder-to-cylinder variation in classification performance, providing a more consistent knock detection accuracy across all cylinders. These findings suggest that combining ion current and knock indicators enhances knock detection reliability, offering a robust solution for real-time applications in engine control systems.

**Keywords:** knock detection; machine learning; ion current

check for updates

**Citation:** Björnsson, O.; Tunestål, P. Knock Detection with Ion Current and Vibration Sensor: A Comparative Study of Logistic Regression and Neural Networks. *Energies* **2024**, *17*, 5693. https://doi.org/10.3390/en17225693

Academic Editor: Constantine D. Rakopoulos

Received: 9 October 2024
Revised: 8 November 2024
Accepted: 11 November 2024
Published: 14 November 2024

## 1. Introduction

Knock detection is crucial in ensuring optimal performance and preventing damage in spark-ignition (SI) engines. Accurate detection allows for timely adjustments, such as retarding spark timing, to avoid the engine damage caused by uncontrolled combustion events. Traditional knock detection methods primarily rely on vibration sensors attached to the engine block, where knock intensity is typically estimated as the sum of the absolute value of a band-pass filtered signal. However, due to the multiple moving parts within the engine, these sensors pick up unwanted mechanical noise, which interferes with the signal and makes vibration-based knock detection prone to cylinder-specific variability [1]. This inconsistency complicates the detection process and often results in inaccurate or unreliable knock classification across different cylinders.

Ion current is an in-cylinder measurement, typically sensed using the spark plug in SI engines. In other combustion concepts, ion current can be measured with a dedicated sensor. For SI engines, ion current is divided into three phases: the ignition phase, the chemical phase, and the thermal phase. The ignition phase is characterized by a strongly ringing signal, influenced by the ignition and ignition circuitry. The chemical phase begins almost immediately after ignition and captures the dynamics within the flame kernel's reaction zone near the spark plug's electrode gap. As the flame kernel moves away from the spark plug, the ion current decreases until the final phase, the thermal phase, when the in-cylinder temperature and pressure rise enough to ionize nitrogen in the air, increasing the number of charged particles and the ion current. Due to this, there is a strong correlation between the ion current, especially during the thermal phase, with in-cylinder pressure and temperature, making it a valuable tool for combustion diagnostics and the estimation of combustion parameters [2,3].

Ion current measurements offer a promising alternative to vibration sensors. By capturing combustion information directly within each cylinder using the spark plug as the sensor, ion current data are inherently cylinder-agnostic, providing a more consistent signal that is unaffected by external mechanical noise. Previous studies have shown the potential of ion-current-based detection for detecting knock [4–6]. Given the complementary nature of vibration sensors and ion current data, merging the two could provide an even more robust solution. This was demonstrated in a previous study [7], which showed improvements both in knock detection accuracy and reduced cylinder variability by combining knock indicators from both sources.

However, even though ion current measurements are cylinder-agnostic, there may be issues related to the cycle-to-cycle correlation between the knock intensity of the ion current and the reference knock intensity calculated using the in-cylinder pressure. For example, in [8], a correlation analysis between pressure and ion knock intensities was performed, with calculations conducted for different knock modes. These modes referred to specific pressure wave frequencies generated within a combustion chamber during knock events. The results showed that the correlation between pressure and ion signals varied across modes and knock intensity levels (calculated based on in-cylinder pressure). Mode 1, corresponding to the fundamental knock frequency, consistently exhibited the strongest correlation across all levels of knock intensities. Thus, care needs to be taken to ensure robust calculations of the knock intensity for the ion current.

Research on the application of ion current in engines has been ongoing for a long time, experiencing a renaissance period in the 1990s and early 2000s. However, recent literature on using ion current for knock detection remains limited. In [9], similar bandpass filter techniques, as previously mentioned, were applied to a boosted gasoline engine to distinguish between normal combustion, light knock, and heavy knock, though the accuracy of the model was not discussed. In [10], while not directly detecting knock, they used the ion current before the start of combustion to detect pre-ignition in a gasoline direct-injected boosted engine. The pre-ignition detection was then used in a control loop to prevent super-knock through fuel re-injection. An example of an artificial neural network (ANN) for knock detection based on ion current can be found in [11], where it was applied to a gasoline engine equipped with a passive pre-chamber. From the ion current measured inside the pre-chamber, they identified a secondary peak during the thermal phase, which only appeared during knocking combustion. They leveraged this by defining a parameter they denoted as LIP-CIP, the difference in crank angle degrees (CADs) between the last thermal peak and the chemical peak, which showed a strong correlation with engine knock. By using this parameter, along with related parameters derived from LIP-CIP as inputs to an ANN, they achieved a remarkable knock detection accuracy of 98.4%.

Beyond knock detection, due to its correlation with in-cylinder pressure and temperature, ion current has proven useful in various engine control applications, including estimating air–fuel ratio [12,13], detecting misfires [14,15], estimating peak pressure location [16], and predicting torque [17], among others. Its multi-functional nature makes it an attractive sensor for enhancing combustion control strategies.

In this article, we expand on previous work [7] by comparing knock detection performance using vibration sensor data, ion current data, and a combination of both. These comparisons were made across linear models, specifically logistic regression and machine learning models, including fully connected neural networks (FCNN) and convolutional neural networks (CNN). A key difference from the earlier study is that Mahalanobis distance was not used in our analysis, as it yielded worse results than the two-variable logistic regression model in our tests.

The remainder of this article is structured as follows: Section 2 describes the engine specifications and experimental setup, and Section 3 outlines the data collection process. Section 4 provides the definition and calculation of knock intensity, followed by Section 5, which discusses the data pre-processing steps. Section 6 introduces the machine learning models used in the study, and Section 7 presents typical data characteristics of the sensors. Section 8 presents the results and analysis, while Section 9 discusses the findings. Finally, Section 10 concludes the paper.

## 2. Engine Specifications & Experimental Setup

Data for this study were collected from a 13-L compressed natural gas (CNG) heavy-duty spark-ignited engine manufactured by Volvo. The engine specifications are outlined in Table 1. Each cylinder was equipped with an in-cylinder pressure sensor (AVL GU24D model) to measure real-time combustion data. An ignition control module (ICM), supplied by SEM AB, measured the ion current. The engine's operation was controlled by an engine control unit (ECU) provided by Metatron. An overview of the experimental setup, including the sensor placements, can be found in Figure 1.

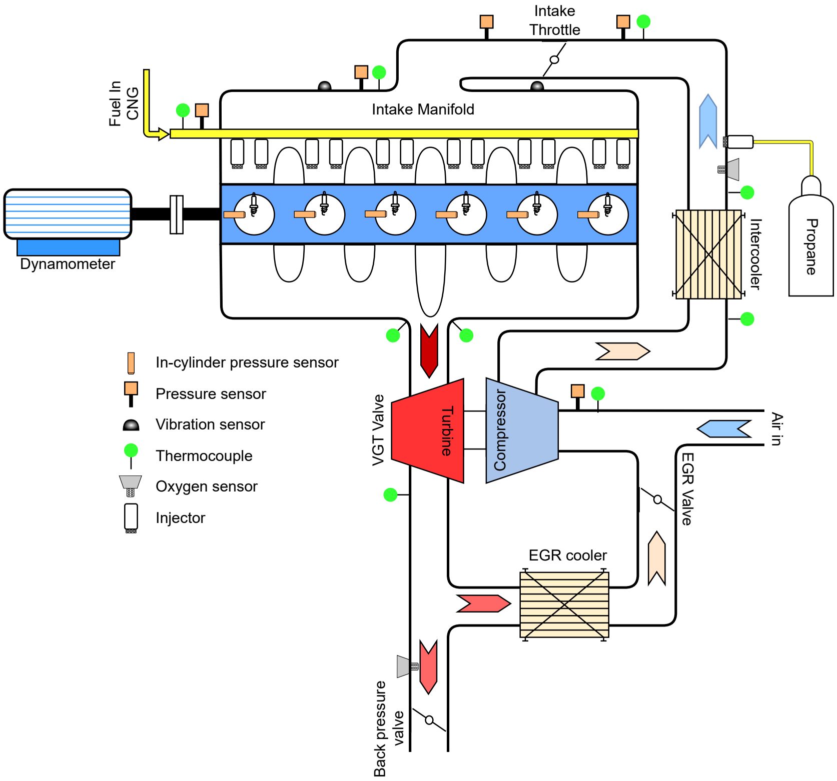


**Figure 1.** Schematic of experimental setup.

**Table 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 |
| Maximum Speed | 2000 RPM |
| Maximum Torque | 2200 Nm |

## 3. Data Collection

To protect the engine from potential damage, knock data were collected at slightly lower power levels than the engine's natural knock limit, reducing the maximum in-cylinder pressure. Data were gathered at engine speeds of 1000 and 1400 RPM, with a load of 1400 Nm. To induce knock in these conditions, the intake air temperature was increased. At the 1000 RPM operating point, a small amount of propane was also injected, due to its higher knock tendency compared to natural gas.

The engine operated under stoichiometric conditions throughout the tests. When propane was injected, the lambda feedback system adjusted the natural gas injection to maintain stoichiometry.

Data from the ion current, vibration sensors, and in-cylinder pressure were collected at a sample rate of 0.1 CAD. The ICM measured the ion current, and the signal was replicated to a real-time computer running LabVIEW for data logging. The ion current signal, multiplexed across cylinders, provided around 90 CAD worth of data per combustion event. For efficiency, the ion current data were truncated to 700 samples, covering the range from −30 CAD before top dead center (TDC) to 39.9 CAD after TDC, ensuring the necessary information was captured across all operating conditions. A total of 5988 cycles from each cylinder were recorded for each operating point.

## 4. Knock Intensity Definition and Calculation

In this study, the reference (true) knock intensity was determined by calculating the maximum amplitude pressure oscillations (MAPO). Two different estimators of knock intensity were derived from the vibration sensor and ion current measurements. These estimators are also referred to as knock indicators.

The MAPO represents the peak pressure fluctuations within the combustion chamber during knock events. These pressure oscillations are caused by shock waves generated from the uncontrolled ignition of the air–fuel mixture, leading to rapid pressure rises. Cycles are categorized into two classes based on MAPO values:

- **No-Knock** : $\text{MAPO} \leq 0.4$
- **Knock**: $0.4 < \text{MAPO}$

As shown in Equation (1), the MAPO is calculated by applying a band-pass filter to the in-cylinder pressure signal, isolating the frequency range associated with knock events. The maximum value of the absolute band-passed filtered signal is taken as the MAPO value. The band-pass filter was tuned to match the knock frequency characteristics of the engine, with the passband set between 4500 and 7000 Hz, based on the fundamental frequency of knock identified in the engine.

$$\text{MAPO} = \max(|p_{bp}|) \tag{1}$$

where $p_{bp}$ is the band-pass-filtered in-cylinder pressure.

In addition to MAPO, two knock indicators were introduced: $KI_v$, derived from the vibration sensor, and $KI_i$, based on ion current measurements. These knock indicators, unlike MAPO, are calculated by summing the absolute values of the band-pass-filtered signal, rather than focusing on the peak value. $KI_v$ and $KI_i$ served as inputs to the logistic regression and convolutional neural network models.

To ensure that the band-pass filter frequencies used to calculate $KI_i$ and $KI_v$ were set properly, the power spectral density was calculated for both the ion current and vibration sensor data, and compared between no-knock and knock cycles. For $KI_i$, the same frequency used to calculate MAPO was determined to be suitable. For $KI_v$, the same start frequency, 4500 Hz, was appropriate; however, the optimal stop frequency was as close to the Nyquist frequency as possible. To allow for future comparisons at different operating points, the stop-band frequency was set to 24,000 Hz to coincide with the Nyquist frequency at 800 RPM in the current measurement setup.

## 5. Data Pre-Processing

The dataset consisted of two classes: knock and no-knock cycles. Since the dataset had an inherent imbalance, with fewer knock cycles compared to no-knock cycles, the data were balanced before training the models. To achieve this, a random subset of no-knock cycles was selected to match the number of knock cycles. Balancing the dataset helps prevent the model from becoming biased toward the more frequent class, ensuring more accurate and reliable predictions.

After balancing, the dataset was split into training, validation, and test sets. A stratified split was performed to preserve the proportion of knock and no-knock cycles in each subset. Specifically, 70% of the data were allocated for training, 20% for validation, and 10% for testing. This approach ensured that the model was trained, validated, and tested on distinct subsets of data, allowing for an unbiased evaluation. The training set was used to fit the model, the validation set was employed to tune hyperparameters and prevent overfitting, and the test set provided an independent assessment of the final model's performance.

In addition, the ion current measurements were normalized by dividing each cycle's ion current values by the maximum ion current value in the training set. This normalization step ensured that the input data were on a consistent scale, improving the stability of the model during training.

The knock indicators $KI_v$ and $KI_i$ were normalized separately to scale the values between 0 and 1. This step ensured that both knock indicators were on the same scale, which is particularly useful for neural networks. However, this scaling did not affect the results of the logistic regression models, as they remain unaffected by the input data's range, focusing instead on the relationships between variables.

## 6. Description of Machine Learning Models

### 6.1. Logistic Regression

Logistic regression, or its equivalents, is already widely used in most engines, where a simple threshold is often applied based on the knock indicator $KI_v$ from the vibration sensor to detect knock events. While this approach does not require inputting the data into a formal model, it operates similarly to logistic regression by using a predefined threshold for classification. In this study, we employed logistic regression to formalize this process, offering a more interpretable analysis of knock detection by evaluating model performance using the area under the curve (AUC) of the receiver operating characteristic (ROC) curve. A formal introduction to AUC and ROC can be found in Section 6.5.

Logistic regression is a binary classification model that outputs the probability of an event, such as a knock, occurring. By applying a decision threshold, the model classifies an example as either a knock or non-knock event. In this work, we applied logistic regression to the knock indicators $KI_v$ and $KI_i$, derived from the vibration sensor and ion current, respectively.

Two logistic regression models were used. First, single-variable models were trained using $KI_v$ or $KI_i$ independently, following the same principle as the threshold-based methods currently used in engines. This allowed a structured evaluation of each knock indicator within the context of a statistical model. Second, a dual-variable logistic regression model was introduced, combining both $KI_v$ and $KI_i$ to leverage the complementary information they provide, aiming to improve the knock detection accuracy. Note that the dual-variable logistic regression model will also be interchangeably referred to as the dual-input logistic regression model, to maintain consistency with the naming convention used for dual-input convolutional neural networks.

By applying logistic regression in this manner, we not only replicated existing knock detection techniques but also enhanced the interpretability of the results through metrics like AUC. This provided a clearer, more standardized assessment of model performance and allowed a more detailed analysis of knock detection capabilities using different indicators.

### 6.2. Fully Connected Neural Network

A fully connected neural network, also known as a feedforward neural network, is a fundamental deep learning model used to learn complex relationships between input features and target outputs. In this study, a FCNN was employed to classify knock events based on ion current measurements. The term “fully connected” is used here to align with the naming convention of CNNs, which will be discussed in the next section.

As the name suggest, in an FCNN, each node in one layer is connected to every node in the subsequent layer. The input data, in this case the ion current, are passed through multiple fully connected layers, where the network learns various weighted combinations of the input features. Each connection between nodes is associated with a weight, and each node has a unique bias term, allowing it to adjust its activation independently of the input. These weights and biases are adjusted during the training process to minimize classification errors. A schematic representation of the FCNN is shown in Figure 2, where the connections (arrows) represent these weights. The final layer outputs a probability indicating the likelihood of a knock event.

Although the FCNN model has a more straightforward structure compared to the CNN architecture, it is a powerful tool for capturing non-linear relationships between input data and the target output, making it a versatile choice for classification tasks. However, a drawback of an FCNN is that its structure does not inherently leverage the sequential nature of the input, unlike CNNs, which excel at capturing spatial and temporal patterns.

In this study, only an FCNN trained on the ion current was evaluated, as its performance was suboptimal compared to the other models, as will be discussed later.

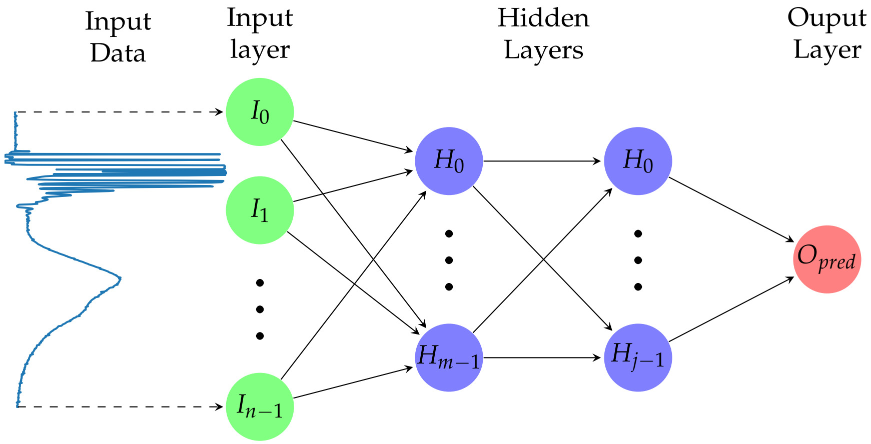


**Figure 2.** Schematic representation of an FCNN architecture, showing an input layer with $n$ inputs, two hidden layers (with $m$ nodes in the first layer and $j$ nodes in the second), and an output layer producing a single output, $O_{pred}$, representing the prediction. The diagram illustrates the ion current as the input data.

### 6.3. Convolutional Neural Network

Convolutional neural networks are a class of deep learning models that have achieved significant success in fields such as image recognition [18], text recognition [19], and time-series analysis [20]. CNNs are particularly adept at capturing spatial and temporal patterns. In contrast to fully connected neural networks, CNNs are more effective at detecting localized patterns, trends, or periodicities in data, which is crucial for knock detection. The architecture of CNNs is generally divided into two main components: feature extraction and classification.

The feature extraction part of the CNN consists of one or more convolutional layers, each followed by pooling layers. Convolutional layers apply filters that slide across the input data, detecting local patterns like trends, peaks, and other features indicative of knock events. These filters generate feature maps that highlight the presence of specific patterns at various positions in the data. Pooling layers, typically using max-pooling, then downsample the feature maps, reducing their dimensionality, while preserving important information. This process improves the computational efficiency and enhances the model's robustness by making it invariant to small shifts in the data.

In the classification part of the CNN, fully connected layers are used to interpret the features extracted during the convolutional process. These fully connected layers are similar to those in an FCNN, where each neuron is connected to every neuron in the subsequent layer. In this study, the pre-processed ion current data served as the input to the CNN, allowing the network to learn patterns associated with no-knock and knock events.

Two CNN models were trained in this study. The first model was trained solely on ion current data, while the second model, referred to as the dual-input CNN model, incorporated both $KI_v$ and $KI_i$ as additional inputs. In this dual-input model, the ion current data were processed through the convolutional layers to extract features. $KI_v$ and $KI_i$ were then added as separate input features, directly to the first fully connected layer. To facilitate this, the tensor containing $KI_v$ and $KI_i$ was concatenated with the tensor containing the features extracted from the ion current data after processing through the convolutional layers. This dual-input architecture aimed to enhance the classification performance by leveraging complementary information from both the ion current and the knock indicators. A flowchart of the dual-input CNN architecture is provided in Figure 3. The CNN model trained solely on ion current data followed a similar structure, without the "concatenate dual-input" step.

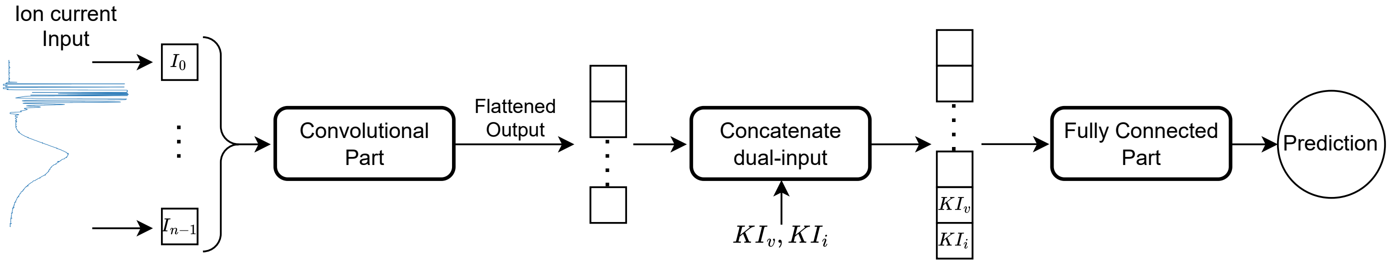


**Figure 3.** Flowchart of the dual-input CNN architecture. The model receives an ion current signal as input, which passes through the convolutional part of the network to extract features. The resulting flattened output is concatenated with knock indicators ($KI_v$ and $KI_i$), before passing through the fully connected part. The final output from the network is a prediction of the probability of knock occurrence.

### *6.4. Hyperparameter Tuning*

The hyperparameters for both the CNN and FCNN models were optimized using Optuna, a powerful hyperparameter optimization framework. For further details on Optuna, we refer to [21]. In this study, Optuna was employed to systematically explore the hyperparameter search space and identify the optimal configuration for each model.

The search space for the CNN and FCNN hyperparameters is outlined in Table 2. Some parameters are specific to the convolutional layers and were not applicable to the FCNN; these are denoted with a "-" in the table. The objective function in Optuna was set to maximize the AUC on the validation set. This allowed Optuna to search for the hyperparameters that yielded the best classification performance.

Both models were trained using the binary cross-entropy loss function, and early stopping was implemented with a patience of 10 epochs to prevent overfitting. Additionally, batch normalization was applied after each convolutional layer in the CNN model.

**Table 2.** Hyperparameter search space, the "-" in the FCNN column indicates that they are not applicable.

| Hyperparameter | FCNN Range/Choices | CNN Range/Choices |
|---|---|---|
| Number of conv layers | - | 1, 2, 3, 4 |
| Number of filters per layer | - | 2, 4, 8, . . . , 256 |
| Kernel sizes (Conv) | - | 3, 5, 7, 9 |
| Kernel sizes (Pool) | - | 2, 3 |
| Number of FC Layers | 1, 2, 3, 4 | 1, 2, 3, 4 |
| Number of nodes per FC Layer | 2, 4, 8, . . . , 256 | 2, 4, 8, . . . , 256 |
| Dropout rate | 0.1 to 0.5 | 0.1 to 0.5 |
| Learning rate | $10^{-5}$ to $10^{-3}$ (log-uniform) | $10^{-5}$ to $10^{-3}$ (log-uniform) |
| Activation functions | ReLU, LeakyReLU, Sigmoid, Tanh | ReLU, LeakyReLU, Sigmoid, Tanh |

### *6.5. Receiver Operating Characteristic*

The receiver operating characteristic (ROC) curve and its area under the curve (AUC) are useful and widely adopted metrics for evaluating the performance of binary classification models [22]. In the context of knock detection, ROC curves help visualize how well a model distinguishes between knock and no-knock cycles, providing a comprehensive picture of the trade-offs between true positive rates (TPR, or sensitivity) and false positive rates (FPR, or 1-specificity).

The ROC curve is a plot of the TPR against the FPR at various decision thresholds. To calculate the TPR and FPR, the decision threshold was adjusted across a range, and at each setting, we obtained counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). TPR was calculated as the ratio of true positives to the total actual positives:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

while FPR is the ratio of false positives to the total actual negatives:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{3}$$

Examples of ROC curves can be found in Section 8, Results and Analysis. A perfect classifier would be represented by a point at the top-left corner of the plot, indicating 100% sensitivity (TPR) and 0% false positive rate (FPR).

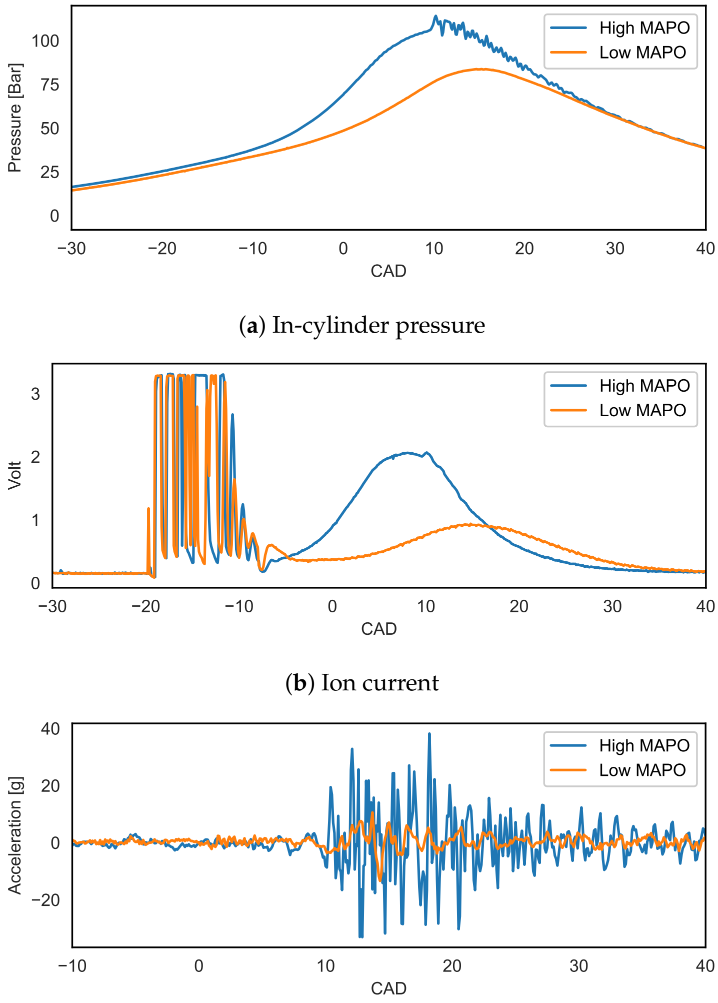
As the decision threshold varies, the ROC curve traces a model's performance across the full range of possible thresholds, allowing us to see the trade-offs between sensitivity and specificity. The AUC represents the probability that the classifier ranks a randomly chosen knock cycle higher than a randomly chosen no-knock cycle. The AUC is calculated as the integral of the ROC curve over all possible thresholds, with values ranging from 0 to 1. An AUC of 1 indicates a perfect classification, while an AUC of 0.5 suggests the model is no better than random guessing.

In this study, AUC was the primary metric used to evaluate the model performance, as it provides a more comprehensive assessment of model performance than simple accuracy.

## 7. Data Characteristics

Before presenting the results, we will first compare the signals for knocking and no-knock cases. This comparison is shown in Figure 4 below, where the knocking cycle had a high MAPO value of 3.4, and the no-knock cycle had a MAPO of 0.05. In Figure 4a, the pressure trace reveals the characteristic ringing of a knocking cycle. Due to the intensity of this knock, the combustion was also significantly advanced compared to the non-knocking cycle. Similarly, the ion current in Figure 4b shows both a higher amplitude and earlier onset for the knocking case. Although the characteristic ringing evident in the pressure trace is less pronounced in the ion current signal, analyzing the power spectral density (PSD) reveals a subtle difference in the frequency content, sharing the same fundamental knock frequency observed in the pressure trace. However, the magnitude of the PSD differed substantially, with the ion current showing much lower values (−76 dB/Hz) compared to the pressure trace (−55 dB/Hz). Lastly, Figure 4c compares the output from the vibration sensor, where, as expected, the amplitude difference between knocking and no-knock cases was substantial.

As discussed in the CNN section, convolutional neural networks are well-suited for identifying temporal and spatial relationships within data. Although CNNs are often considered “black-box” models, which makes it challenging to interpret which features are most important or how they are extracted, it is reasonable to assume that the model could capture the temporal differences between knock and no-knock cycles in the ion current signal. Given the relatively low magnitude of the frequency content in the ion current, dual-input models offer an effective approach to ensure that the CNN leverages both temporal and frequency-related information.



(**a**) In-cylinder pressure

(**b**) Ion current

(**c**) Vibration sensor

**Figure 4.** Characteristics of sensor signals for knock and no-knock cases. (**a**) In-cylinder pressure, (**b**) ion current, and (**c**) vibration sensor signals, illustrating differences in signal behavior between knock and no-knock events.

## 8. Results and Analysis

The results and analysis are structured to first present the performance of a logistic regression model based on individual knock indicators: one derived from the vibration sensor and the other from the ion current. This is followed by the results of a logistic regression model utilizing both knock indicators. Next, the analysis shifts to neural network models, beginning with a fully connected model that used only the ion current as input. We then present results from a CNN model also using only ion current data. Finally, we introduce a CNN model where the ion current served as the input to the convolutional layers, with both vibration sensor and ion current indicators integrated into the fully connected layers. No fully connected model was trained using both inputs, as the fully connected model based solely on ion current data was outperformed by the CNN, as will be demonstrated in the subsequent sections. Note that all the results presented in this section are based on the results from the test set.

### *8.1. Single-Variable Logistic Regression Models*

The performance of the single-variable logistic regression model based on $KI_v$ varied across cylinders, as expected, due to its susceptibility to mechanical noise from the engine. The AUC values ranged from 0.87 in cylinder 4, where the model was most affected by noise, to a high of 0.98 in cylinder 2, which exhibited the least noise interference. Cylinder 1 showed moderate sensitivity to noise, with an AUC of 0.93. In contrast, the remaining cylinders delivered more consistently, with AUC values ranging from 0.96 to 0.97. While variable, the $KI_v$ model achieved a strong AUC of 0.96 or higher in four of the six cylinders, demonstrating effective knock detection in most cases.

The logistic regression model based on $KI_i$ demonstrated a more stable performance across all cylinders. The AUC values for this model ranged from 0.91 to 0.94, highlighting its consistent ability to detect knock events across cylinders. While the top-performing cylinder in the $KI_i$ model (AUC = 0.94) did not exceed the best result from the $KI_v$ model, the reduced variation in performance underscores the advantage of ion current measurements for reliable knock detection.

A comparison of the ROC curves for the best and worst performing cylinders in both models is presented in Figure 5, illustrating the trade-off between the peak performance of $KI_v$ and the consistency of $KI_i$-based detection.

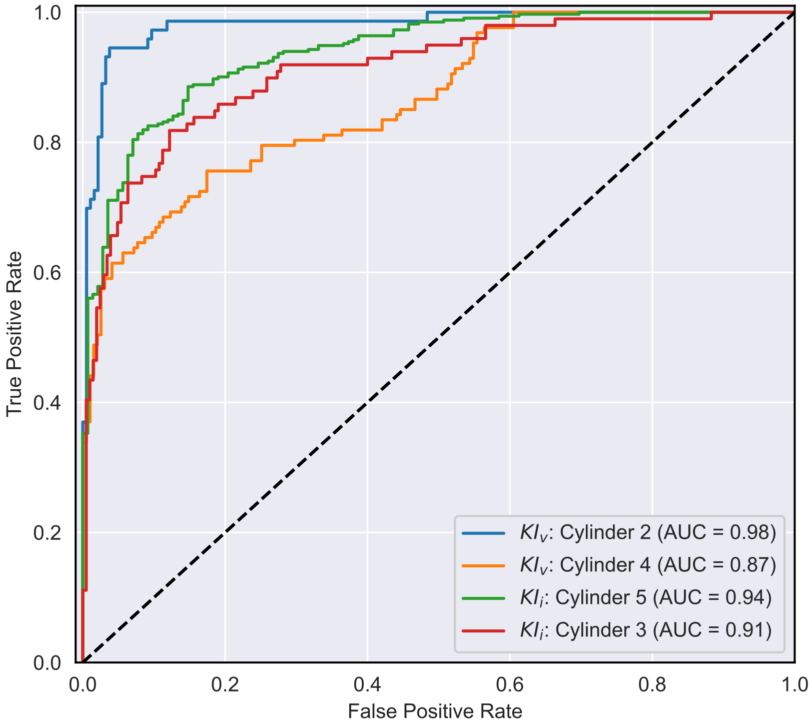


**Figure 5.** Depicts the ROC curves and the corresponding AUCs for the best and worst performing cylinders for the single-variable logistic regression classifiers based on either $KI_v$ or $KI_i$.

### 8.2. Dual-Variable Logistic Regression Model

The single-variable models, while effective in many cases, highlighted the limitations of using $KI_v$ and $KI_i$ independently for knock detection. $KI_v$ showed a strong performance in specific cylinders but suffered from mechanical noise, resulting in variability between the cylinders. On the other hand, $KI_i$ offered a more consistent performance across cylinders but did not reach the peak AUC values observed with $KI_v$ in the best-performing cylinders. Given these trade-offs, combining the strengths of both indicators in a dual-variable logistic regression model was explored, to provide a more balanced and robust solution.

The dual-variable model, which incorporated both $KI_v$ and $KI_i$, produced AUC values ranging from 0.94 in cylinder 4 to 0.98 in cylinder 2, with the corresponding ROC curves depicted in Figure 6. These results are consistent with those from the single-variable logistic regression model based on $KI_v$, where $KI_v$ contributed the least information in cylinder 4 and the most in cylinder 2. Overall, the dual-variable model outperformed both single-variable models, offering reduced variation in performance across cylinders and matching or exceeding their AUC performance, demonstrating an improvement in knock detection capability.

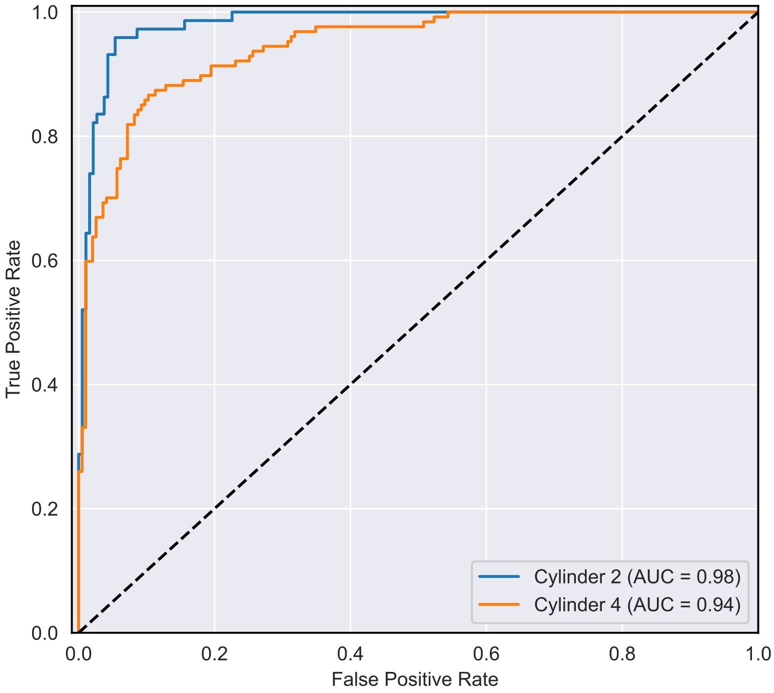


**Figure 6.** Depicts the ROC curves and the corresponding AUCs for the best and worst performing cylinders for the logistic regression classifiers based on both $KI_v$ and $KI_i$.

### 8.3. Fully Connected Model-Ion Current Input

The fully connected model based on ion current measurements failed to deliver any particularly noteworthy results. While the model benefited from the consistency provided by in-cylinder measurements, showing stable AUC values across all cylinders, its performance was limited compared to the logistic regression models. Specifically, the AUC values ranged from 0.89 in cylinder 5 to 0.92 in cylinder 3, showing a low spread in the AUC between the cylinders but also a generally moderate predictive capability. When compared to the logistic regression models based on either $KI_v$ or $KI_i$, the FC model did not perform as well overall. In fact, its predictive performance fell short of both single-variable models, making it a less effective solution for knock detection. Due to this underperformance, it was decided not to train an FC model using a combination of ion current and the two additional knock indicators, as there was little indication that such a model would significantly improve upon these results, especially in comparison to the CNN.

### *8.4. CNN Model-Ion Current Input*

In this section, we present the results of the CNN model trained on ion current measurements. The optimal hyperparameters, identified through the optimization process, are detailed in Table 3.

The CNN model demonstrated strong and consistent performance across all cylinders, with minimal variation in AUC values. The highest AUC was 0.97 in cylinder 6, while the lowest was 0.95 in cylinder 5. This narrow range of variation indicates the model's robustness in detecting knock events across different cylinders. The corresponding ROC curves are presented in Figure 7. These are excellent results, on par with the logistic regression model trained on both $KI_i$ and $KI_v$. This underscores the capability of the CNN model, using only ion current data, to deliver competitive performance in knock detection.

**Table 3.** Best CNN model hyperparameters when trained on the ion current.

| Hyperparameter | Value |
|---|---|
| Number of conv layers | 4 |
| Number of filters per layer | 128, 16, 256, 256 |
| Kernel size (conv) | 9 |
| Kernel size (Pool) | 2 |
| Number of FC Layers | 1 |
| Number of nodes per FC Layer | 64 |
| Dropout rate | 0.153 |
| Learning rate | $7.20 \times 10^{-4}$ |
| Activation function (conv) | Tanh |
| Activation function (FC) | LeakyReLU |

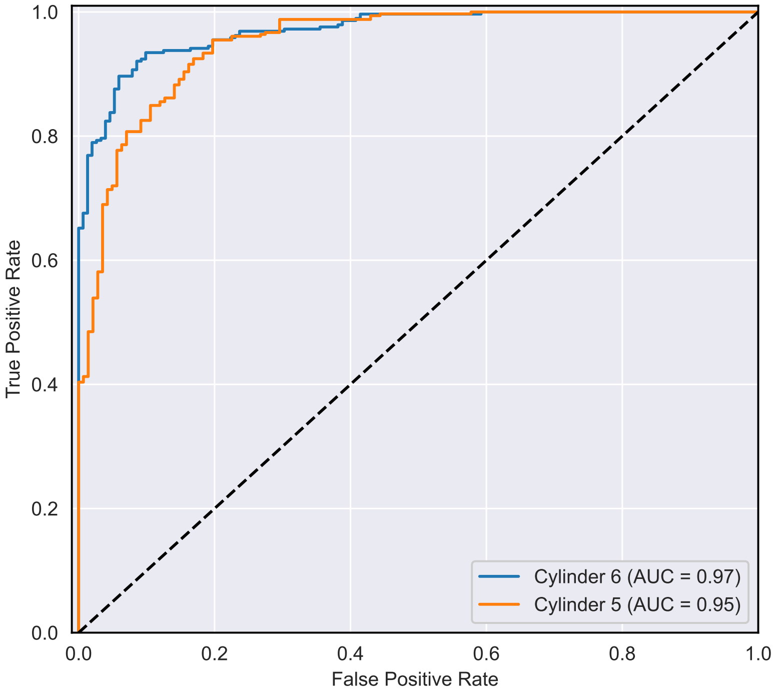


**Figure 7.** Depicts the ROC curves and the corresponding AUCs for the best and worst performing cylinders for the CNN model trained on the ion current measurements.

### *8.5. CNN Model-Ion Current Input and Knock Indicators*

The last model presented combined both the measured ion current and the two knock indicators, $KI_v$ and $KI_i$. This model delivered the best overall performance among all models tested. The optimal hyperparameter configuration is detailed in Table 4, resulting in minimal cylinder-to-cylinder variation in AUC. The best result was an AUC of 0.98 for cylinder 2, matching the top performance of both the dual-input logistic regression model and the single-variable logistic regression model based on $KI_v$. The lowest AUC, observed in cylinder 4, was 0.96, demonstrating consistently high performance across all cylinders. The corresponding ROC curves for these cylinders are presented in Figure 8.

This model exhibited greater stability compared to the other configurations, particularly in reducing the performance variability seen in previous models. The addition of $KI_v$ and $KI_i$ as extra inputs to the fully connected layers enhanced the model's ability to capture knock events and maintain consistent predictive performance.

**Table 4.** Best CNN model hyperparameters when trained on ion current and knock indicators ($KI_v$ and $KI_i$).

| Hyperparameter | Value |
|---|---|
| Number of conv layers | 4 |
| Number of filters per layer | 64, 256, 32, 8 |
| Kernel size (conv) | 9 |
| Kernel size (pool) | 2 |
| Number of FC layers | 4 |
| Number of nodes per FC layer | 256, 256, 64, 32 |
| Dropout rate | 0.469 |
| Learning rate | $9.86 \times 10^{-4}$ |
| Activation function (Conv) | Tanh |
| Activation function (FC) | LeakyReLU |

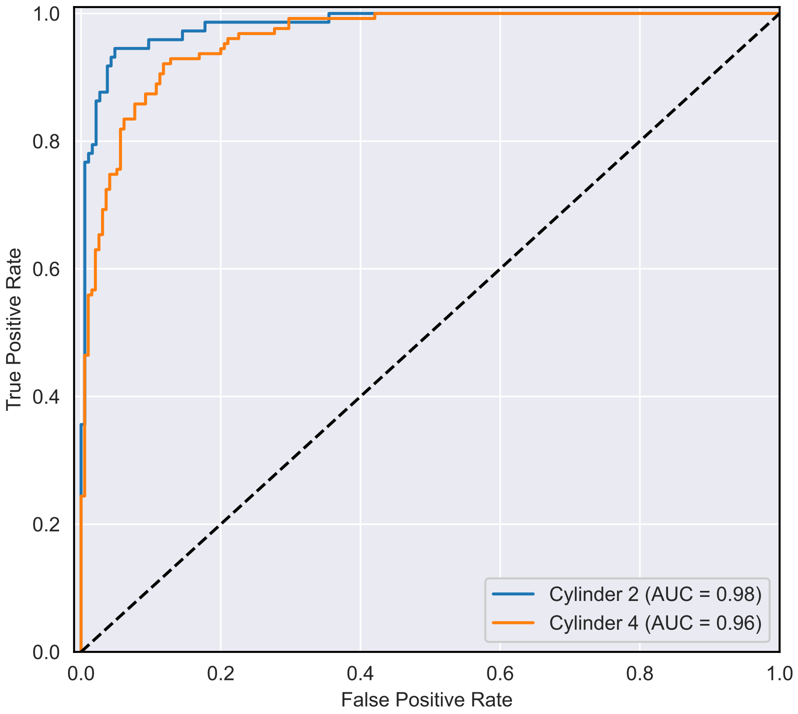


**Figure 8.** Depicts the ROC curves and the corresponding AUCs for the best and worst performing cylinders for the CNN model trained on the ion current measurements and knock indicators.

### *8.6. Performance Comparison of Dual-Input Models*

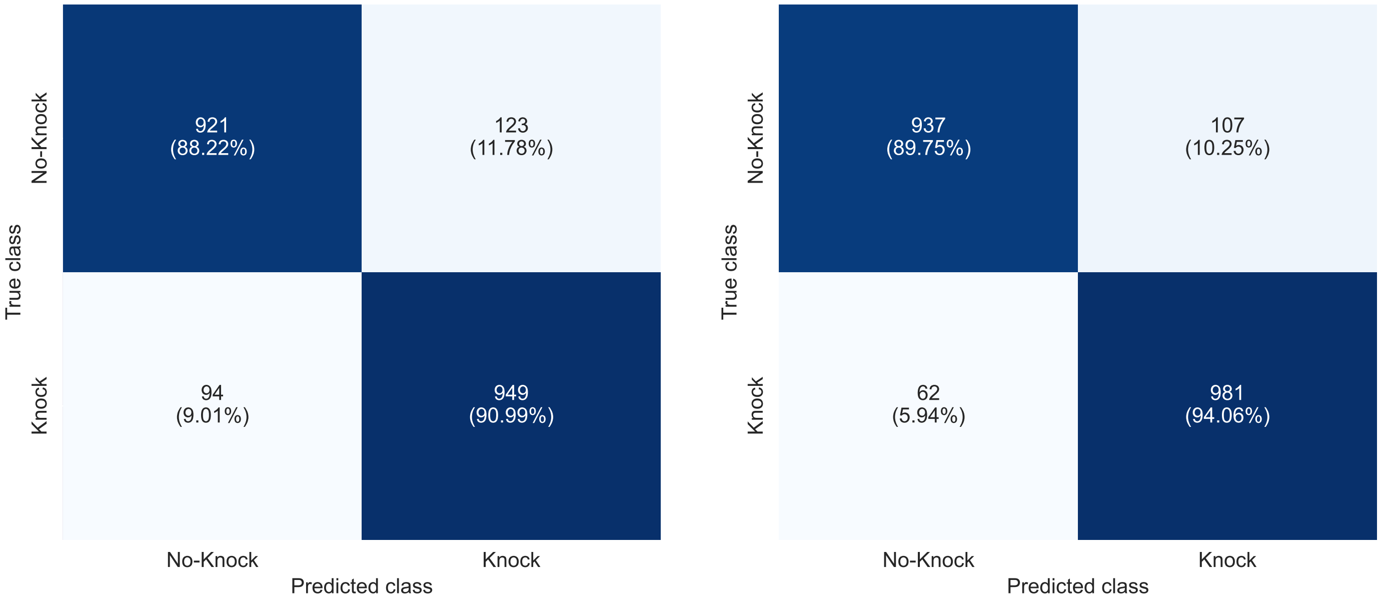
Using both $KI_v$ and $KI_i$ produced the best-performing models. Both the logistic regression and CNN models achieved excellent AUC results, with the CNN model slightly outperforming the logistic regression model. While AUC was the primary metric used to evaluate overall model performance, it is equally important to understand the types of misclassifications the models are prone to. In the following analysis, we shift focus from AUC to examining the actual classifications, which requires setting a decision threshold.

The model predicts the probability of a cycle being a knock event. Based on this probability, the examples were classified as knock or non-knock depending on whether the probability exceeded the threshold. While the default threshold is 0.5, this may not be optimal for every model. In the following sections, the threshold was optimized to maximize the accuracy based on the validation set. For the dual-input logistic regression model, the optimal threshold was found to be 0.4, while for the dual-input CNN model, it was 0.58.

In the following analysis, we focus on overall predictions rather than cylinder-specific results.

#### 8.6.1. Confusion Matrix Analysis

Figure 9 presents the confusion matrices for the logistic regression model and the CNN model, both incorporating the knock indicators $KI_v$ and ion current $KI_i$. A confusion matrix provides a visual representation of a model's classification performance by comparing actual and predicted classifications. Each cell shows the number of instances along with the percentage relative to the total examples for each class (no-knock, knock).



(**a**) Confusion matrix for the logistic regression model.

(**b**) Confusion matrix for the CNN model.

**Figure 9.** Confusion matrices for the logistic regression model (**a**) and the CNN model (**b**). The logistic regression model was based on both knock indicators ($KI_v$ and $KI_i$), while the CNN model used the ion current signal in the convolutional layers and incorporated both knock indicators in the fully connected layers. The confusion matrix compares actual and predicted classifications, with each cell indicating the number of instances and their percentage of the total instances for each class (no-knock, knock). The diagonal elements represent the instances that were correctly classified, while the off-diagonal elements show the instances that were misclassified.

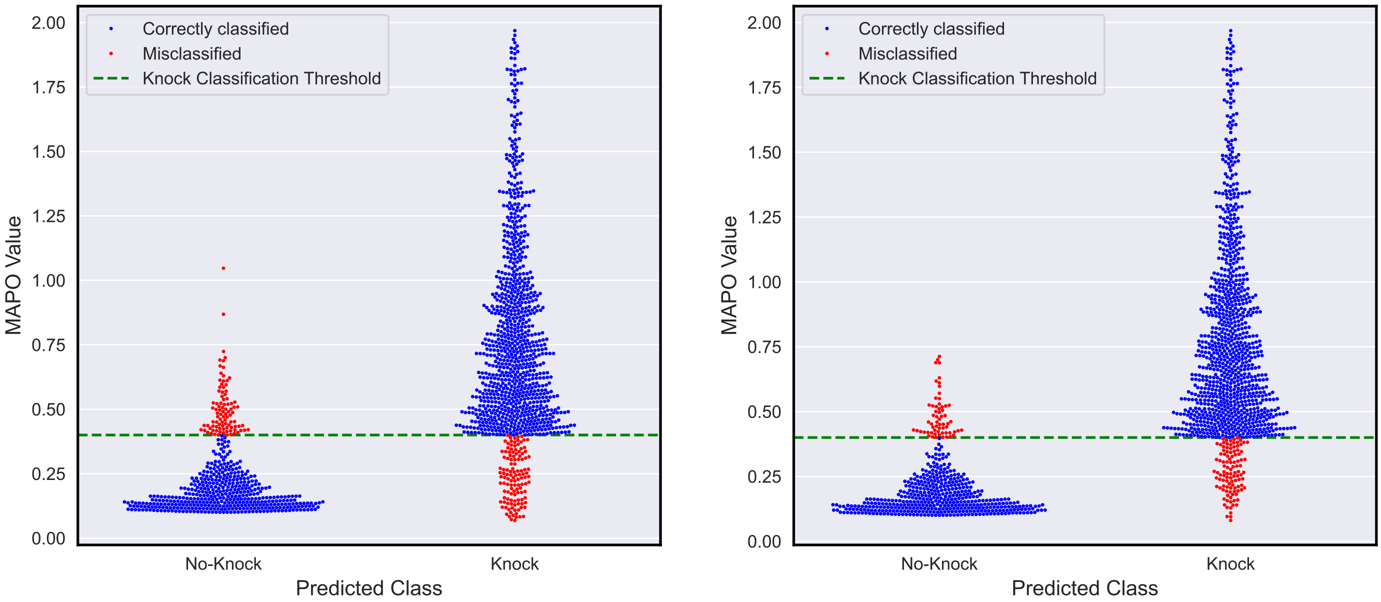
Figure 9a presents the confusion matrices for the dual-input logistic regression after applying the optimized threshold. For the logistic regression model, with a threshold of 0.4, the model correctly classified 90.99% of the knocking cycles and 88.22% of the non-knock cycles. While it misclassified 9.01% of the knocking cycles (false negative) and 11.78% of the no-knock cycles (false positives).

Figure 9b displays the confusion matrix for the dual-input CNN model, using a threshold of 0.58. The CNN model correctly identified 94.06% of knock events, misclassifying 5.94% as no-knock. Its performance in identifying no-knock events was slightly lower at 89.75%, and it misclassified 10.25% of the no-knock cycles.

Both models were more susceptible to erroneously classifying a cycle as knock. However, the CNN model demonstrated a better knock detection accuracy over the logistic regression model. This improvement suggests that the CNN's architecture was more effective in capturing patterns associated with knock events. Overall, the CNN model outperformed the logistic regression model, particularly regarding detecting knock.

#### 8.6.2. Visualization of Predictions from Dual-Input Models

This section examines whether there are differences between the dual-input models regarding the types of examples they misclassified in relation to their MAPO values. The swarm plots in Figure 10 illustrate these differences, with Figure 10a depicting the results from the dual-input logistic regression model and Figure 10b showing those from the dual-input CNN model. In these plots, correctly classified points are represented as blue dots, while misclassified examples are shown in red.



(**a**) Swarm plot for the logistic regression model.

(**b**) Swarm plot for the CNN model.

**Figure 10.** Swarm plot for the logistic regression model (**a**) and the CNN model (**b**). The logistic regression model was based on both knock indicators ($KI_v$ and $KI_i$), while the CNN model used the ion current signal in the convolutional layers and incorporated both knock indicators in the fully connected layers. The swarm plot depicts the distribution of MAPO values for each predicted class. Blue points signify correctly classified instances, and red points indicate misclassified ones. The horizontal dashed line represents the MAPO threshold that separates the true classes of no-knock and knock. Note that some points have been omitted to fit the plot's scale. The plot includes all misclassified examples but excludes examples with MAPO values greater than 2, as they were always correctly classified. Very low MAPO values of correctly classified instances have also been omitted.

To focus on the most relevant data, examples with a MAPO value greater than 2 were omitted because they were always correctly classified. Similarly, examples with

MAPO values less than 0.1 were excluded, unless they were misclassified; all misclassified examples were included, regardless of their MAPO value.

The most important result from Figure 10 is that there are no examples where cycles with exceedingly high MAPO values were misclassified. The main difference between the models' predictions can be seen in the examples that were classified as no-knock. While both models have most of the misclassified examples closer to the MAPO threshold that divides the examples into their true class, the logistic regression model has more examples with higher MAPO values that were misclassified, with the highest having a MAPO of 1.05, while the highest misclassified MAPO for the CNN was only 0.71.

## 9. Discussion

### *9.1. Benefits of Dual-Input Models*

Dual-input models that incorporate both ion current ($KI_i$) and knock sensor indicators ($KI_v$) offer significant advantages over single-input models by leveraging complementary information from two distinct sources. A key benefit of combining these inputs is the reduced cylinder-specific variability often observed when relying solely on a knock sensor. Since $KI_v$ is susceptible to mechanical noise, integrating cylinder-agnostic ion current data provides a more stable and consistent basis for knock detection across all cylinders. Notably, the dual-input CNN model showed the highest performance among the models tested, providing both accuracy and consistency across all cylinders.

This observation raises the question of whether other input parameters could further enhance the model performance. For example, in [23], an algorithm was used to transform tabular datasets into images, allowing a CNN to classify knock using data from conventional sensors. The study found that exhaust gas temperature had a significant impact on detection accuracy. Other promising input parameters include engine speed and load, as the correlation between actual knock intensity and $KI_v$ can vary depending on engine speed [24]. By incorporating engine speed into the input data, the model could adjust the relative importance of $KI_v$ based on speed, potentially improving the detection accuracy under varying operating conditions.

### *9.2. Impact of Vibration Sensor Noise on Model Performance*

The knock indicators based on the standard vibration sensor ($KI_v$) showed strong performance throughout the study. The lowest AUC for the logistic regression model based solely on $KI_v$ was 0.87, which is a respectable result on its own. However, the performance differences between the cylinders for $KI_v$ also had a noticeable impact on the performance of the dual-input models. In every case, cylinder 4, where $KI_v$ was most affected by mechanical noise, had the lowest AUC. This raises the question of what might happen if the performance of $KI_v$ was significantly worse in certain cylinders.

As demonstrated in [7], mechanical noise can severely degrade $KI_v$ performance in some cylinders, potentially reducing the accuracy to the point where it is only marginally better than random guessing. If this occurred, it would have a substantial impact on the dual-input models, as $KI_v$ would contribute more noise than useful information. For the logistic regression models, a practical solution would be to avoid using dual-variable models for these problematic cylinders, instead relying solely on ion current $KI_i$-based models.

For the CNN model, an alternative approach to mitigate the effects of poor $KI_v$ performance would be to include an additional input that encodes which cylinder the sample originates from. This could be achieved by adding a one-hot-encoded array to the fully connected layers, where each element of the array corresponds to a specific cylinder. By providing the model with cylinder-specific information, it may learn to better account for noise in certain cylinders, improving the overall performance.

### *9.3. CNN Dual-Input Discussion: Is Adding $KI_i$ Redundant?*

The dual-input CNN model incorporated knock indicators from both the ion current ($KI_i$) and vibration sensor data ($KI_v$) by feeding these indicators into the fully connected layers. However, given that the convolutional layers were trained directly on the ion current signal, one could question whether incorporating $KI_i$, derived from the same ion current data, adds meaningful value or introduces redundancy.

CNNs can capture localized patterns and temporal dependencies within input data, which suggests that the convolutional layers may already capture key aspects of the ion current signal, including the frequency content. Since $KI_i$ is derived from the same source, there is a potential overlap between what the convolutional layers learn and what $KI_i$ contributes when added as a separate input to the fully connected layers.

That said, even if $KI_i$ introduces redundancy, its inclusion should not detract from the model performance, provided that the network is trained on a sufficient amount of data. $KI_i$ represents an explicit measure of the knock intensity in the ion current data, and including it as an additional input might enhance the model's ability to generalize across cylinders by focusing on global knock trends.

### *9.4. Real-Time Implementation Challenges*

Neural networks inherently face challenges in real-time engine control applications compared to threshold-based decision methods, due to their computational complexity. The CNN models proposed in this study require millions of floating-point operations during inference to process inputs and generate predictions, which poses a challenge for ECUs with limited processing capabilities. Upgrading to more powerful hardware would increase costs. However, there are two reasons why neural networks remain a feasible and beneficial solution for real-time use.

First, the number of floating-point operations can be reduced without significantly impacting performance by using smaller, more efficient models. For example, a dual-input CNN model with three convolutional layers (with filter sizes of 4, 2, and 4) and two fully connected layers (with 256 and 4 nodes, respectively) performed well on the validation set, requiring only around 55,000 floating-point operations. While the focus of this study was on maximizing performance, resulting in more complex models, future real-time implementations could prioritize simpler models by restricting the hyperparameter ranges, thereby producing models that are more manageable for ECU integration.

Secondly, improved knock detection can enable more efficient engine operation near the knock limit by reducing unnecessary control actions. This, in turn, would reduce fuel consumption, helping to offset the potential extra cost of more powerful ECUs and making the investment more manageable for end consumers.

### *9.5. Future Work*

There are several interesting directions to extend this research. One potential direction is integrating data from additional sensors to provide a richer set of features for knock detection, which may further enhance the model accuracy. Another area of interest is exploring how the models perform when vibration sensor data from certain cylinders are highly degraded and investigating strategies to overcome this.

Additionally, the role of $KI_i$ in the dual-input CNN model warrants further study. While $KI_i$ provides useful knock intensity information, its redundancy when the CNN already directly processes ion current data should be assessed. Reducing the computational complexity of the CNN model is also a priority. This could involve downsampling the ion current input or using a more compact model with fewer layers. The challenge lies in determining whether these adjustments can be implemented without significantly compromising accuracy.

Finally, evaluating the performance of these models on larger datasets and across a broader range of operating conditions is needed to understand their generalizability and effectiveness in real-world applications.

## 10. Conclusions

This study aimed to improve knock detection in spark-ignition engines by leveraging knock indicators derived from vibration sensor ($KI_v$) and ion current measurements ($KI_i$). Logistic regression models were trained on $KI_v$ and $KI_i$ individually and combined, while the fully connected neural network was trained solely on the ion current signal. Two convolutional neural network models were also developed: one trained only on the ion current signal, and the other using the ion current signal alongside $KI_v$ and $KI_i$.

The single-variable logistic regression model based on $KI_v$ showed strong performance in most cylinders but exhibited variability in detection accuracy due to mechanical noise. In contrast, the single-variable model based on $KI_i$ demonstrated consistent performance across all cylinders, although it did not match the highest AUC values achieved by the best-performing $KI_v$-based model. Dual-input models, particularly the CNN model incorporating both $KI_v$ and $KI_i$, demonstrated superior overall detection performance, while effectively reducing cylinder variability in knock detection compared to single-variable models. Among all the models tested, the dual-input CNN achieved the best overall results, combining high accuracy and consistent performance across cylinders.

These findings suggest that combining vibration sensor and ion current data can lead to more reliable knock detection, improving engine efficiency, especially near the knock limit. Although the computational complexity of CNNs poses a challenge for real-time applications, more efficient model architectures could make them feasible for onboard engine control systems.

**Author Contributions:** Conceptualization, O.B.; Data curation, O.B.; Formal analysis, O.B.; Funding acquisition, P.T.; Investigation, O.B.; Methodology, O.B.; Project administration, O.B. and P.T.; Resources, O.B. and P.T.; Software, O.B.; Supervision, O.B.; Validation, O.B.; Visualization, O.B.; Writing—original draft, O.B.; Writing—review and editing, O.B. and P.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the KCFP Engine Research Center, Swedish Energy Agency (project number 22485-4)

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are part of an ongoing study and were collected using proprietary hardware but are available upon reasonable request.

**Acknowledgments:** We would like to extend our sincere gratitude to Metatron S.p.A. for supplying the ECU and their assistance, SEM AB for providing the ICM along with their support, and Volvo Penta for supplying the engine and support.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| ANN | Artificial Neural Network |
| AUC | Area under the ROC curve |
| CAD | Crank Angle Degree |
| CNG | Compressed Natural Gas |
| CNN | Convolutional Neural Network |
| ECU | Engine Control Module |
| FCNN | Fully Connected Neural Network |
| FN | False Negative |
| FP | False Positive |
| FPR | False Positive Rate |
| ICM | Ignition Control Module |
| $KI_i$ | Knock Indicator based on ion current measurements |
| $KI_v$ | Knock Indicator based on the knock sensor (vibration sensor) |
| MAPO | Maximum Amplitude Pressure Oscillation |
| PSD | Power Spectral Density |

| | |
|---|---|
| ROC | Receiver Operating Characteristic |
| TDC | Top Dead Center |
| TN | True Negative |
| TP | True Positive |
| TPR | True Positive Rate |

## References

1. Naber, J.; Blough, J.R.; Frankowski, D.; Goble, M.; Szpytman, J.E. Analysis of Combustion Knock Metrics in Spark-Ignition Engines. In Proceedings of the SAE 2006 World Congress & Exhibition, Detroit, MI, USA, 3–4 April 2006; SAE Technical Paper 2006-01-0400. [CrossRef]
2. Eriksson, L.; Nielsen, L. Ionization current interpretation for ignition control in internal combustion engines. *Control. Eng. Pract.* **1997**, *5*, 1107–1113. [CrossRef]
3. Yoshiyama, S.; Tomita, E.; Hamamoto, Y. Fundamental Study on Combustion Diagnostics Using a Spark Plug as Ion Probe. In Proceedings of the International Fuels & Lubricants Meeting & Exposition, Baltimore, MD, USA, 16–19 October 2000; SAE Technical Paper 2000-01-2828. [CrossRef]
4. Giglio, V.; Police, G.; Rispoli, N.; Di Gaeta, A.; Cecere, M.; Ragione, L.D. Experimental Investigation on the Use of Ion Current on SI Engines for Knock Detection. In Proceedings of the SAE 2009 Powertrains Fuels and Lubricants Meeting, San Antonio, TX, USA, 2–4 November 2009; SAE Technical Paper 2009-01-2745. [CrossRef]
5. Cavina, N.; Po, G.; Poggio, L. Ion Current Based Spark Advance Management for Maximum Torque Production and Knock Control. In Proceedings of the Volume 4: Fatigue and Fracture, Heat Transfer, Internal Combustion Engines, Manufacturing, and Technology and Society, Torino, Italy, 4–7 July 2006; pp. 537–545. [CrossRef]
6. Auzins, J.; Johansson, H.; Nytomt, J. Ion-Gap Sense in Misfire Detection, Knock and Engine Control. In Proceedings of the International Congress & Exposition, Detroit, MI, USA, 27 February–2 March 1995; SAE Technical Paper 950004. [CrossRef]
7. Ängeby, J.; Johnsson, A.; Hellström, K. Knock detection using multiple indicators and a classification approach. *IFAC-PapersOnLine* **2018**, *51*, 297–302. [CrossRef]
8. Abhijit.; Naber, J. Ionization Signal Response during Combustion Knock and Comparison to Cylinder Pressure for SI Engines. *SAE Int. J. Passeng. Cars Electron. Electr. Syst.* **2008**, *1*, 349–364. [CrossRef]
9. Tong, S.; Yang, Z.; He, X.; Deng, J.; Wu, Z.; Li, L. Knock and Pre-Ignition Detection Using Ion Current Signal on a Boosted Gasoline Engine. In Proceedings of the WCX™ 17: SAE World Congress Experience, Detroit, MI, USA, 3–5 April 2017; SAE Technical Paper 2017-01-0792. [CrossRef]
10. Wang, J.; Hu, Z.; Zhu, D.; Ding, W.; Li, L.; Yan, W.; Jian, T.; Chen, L. In Cycle Pre-Ignition Diagnosis and Super-Knock Suppression by Employing Ion Current in a GDI Boosted Engine. In Proceedings of the WCX SAE World Congress Experience, Virtually, 21–23 April 2020; SAE Technical Paper 2020-01-1148. [CrossRef]
11. Wang, J.; Shi, J.; Deng, J.; Miao, X.; Liu, Y.; Pan, S.; Li, L. Misfire and Knock Detection Based on the Ion Current inside a Passive Pre-Chamber of Gasoline Engine. *Fuel* **2022**, *311*, 122528. [CrossRef]
12. Wickstrom, N.; Taveniku, M.; Linde, A.; Larsson, M.; Svensson, B. Estimating Pressure Peak Position and Air-Fuel Ratio Using the Ionization Current and Artificial Neural Networks. In Proceedings of the Conference on Intelligent Transportation Systems, Boston, MA, USA, 12 November 1997; pp. 972–977. [CrossRef]
13. Schneider, D.; Lai, M.C. Real-Time Air/Fuel-Ratio Control in a Small SI Engine Using the Ionic Current Signal. In Proceedings of the Small Engine Technology Conference & Exposition, Madison, WI, USA, 28–30 September 1999; SAE Technical Paper 1999-01-3323. [CrossRef]
14. Liu, Y.; Deng, J.; Hu, Z.; Li, L. In-Cycle Combustion Feedback Control for Abnormal Combustion Based on Digital Ion Current Signal. *Int. J. Engine Res.* **2018**, *19*, 241–249. [CrossRef]
15. Ayad, S.; Sharma, S.; Verma, R.; Henein, N. Combustion Ionization for Detection of Misfire, Knock, and Sporadic Preignition in a Gasoline Direct Injection Engine. *J. Energy Resour. Technol.* **2019**, *141*, 112209. [CrossRef]
16. Aengeby, J. Closed Loop Control of the Combustion Phase in SI Engines Using Alternative Fuels. *SAE Int. J. Adv. Curr. Pract. Mobil.* **2020**, *3*, 312–319. [CrossRef]
17. Rao, R.; Honnery, D. The Prediction of Torque in a Diesel Engine Using Ion Currents and Artificial Neural Networks. *Int. J. Engine Res.* **2014**, *15*, 370–380. [CrossRef]
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
19. Goodfellow, I.J.; Bulatov, Y.; Ibarz, J.; Arnoud, S.; Shet, V. Multi-Digit Number Recognition from Street View Imagery Using Deep Convolutional Neural Networks. *arXiv* **2014**, arXiv:1312.6082.
20. Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; Zhao, J.L. Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. In *Web-Age Information Management*; Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Kobsa, A., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8485, pp. 298–310. [CrossRef]

21. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019.
22. Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
23. Hosseini, M.; Chitsaz, I. Knock Probability Determination Employing Convolutional Neural Network and IGTD Algorithm. *Energy* **2023**, *284*, 129282. [CrossRef]
24. Cavina, N.; Po, G.; Poggio, L.; Zecchetti, D. Individual Cylinder Knock Detection Based on Ion Current Sensing: Correlation Analysis. In Proceedings of the ASME 2006 Internal Combustion Engine Division Spring Technical Conference (ICES2006), Aachen, Germany, 8–10 May 2006; pp. 171–178. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.