CHALMERS
UNIVERSITY OF TECHNOLOGY

A Virtual CO_2 Sensor for Vehicle Cabins Using Deep Learning and Physics-Based Modelling

Master's thesis in Complex Adaptive Systems

ABDULRAZAK MOHAMED

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

A Virtual CO_2 Sensor for Vehicle Cabins Using Deep Learning and Physics-Based Modelling

Master Thesis in the Master's Program Complex Adaptive Systems

ABDULRAZAK MOHAMED

Department of Physics
Division of Science and Technology
Chalmers University of Technology
Gothenburg, Sweden 2026

A Virtual CO_2 Sensor for Vehicle Cabins Using Deep Learning and Physics-Based Modelling
ABDULRAZAK MOHAMED

Supervisor: Krishna Hariharan, Volvo Group
Examiner: Giovanni Volpe, Department of Physics

Master's Thesis 2026
Department of Physics
Division of Science and Technology
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 79 076 7012

A Virtual CO_2 Sensor for Vehicle Cabins Using Deep Learning and Physics-Based Modeling

ABDULRAZAK MOHAMED
DEPARTMENT OF PHYSICS
Chalmers University of Technology

Abstract

The air quality inside vehicle cabins is an important factor affecting driver alertness, occupant comfort, and occupational health. Among the constituents of cabin air, carbon dioxide (CO_2) is of particular interest because it is produced continuously by occupants and accumulates rapidly in the enclosed space of a modern vehicle. Sustained exposure above 1,000 ppm measurably degrades driver alertness and decision-making, creating a clear need to monitor cabin CO_2 during operation. Physical CO_2 sensors are the conventional approach and are already used in some vehicles to trigger ventilation switching, but they rely on infrared or photo-acoustic measurement principles that are susceptible to drift from temperature and humidity, require periodic recalibration as optics age, and degrade under the vibration and thermal cycling of vehicle operation. These limitations render them unreliable for continuous, fleet scale monitoring. This thesis addresses this gap by developing a virtual CO_2 sensor that predicts cabin concentration from signals already available on the vehicle controller area network (CAN) bus, eliminating the dependence on dedicated hardware. Three modelling approaches are developed and compared: a purely data-driven deep learning model, a physics-based mass-balance model, and a hybrid approach in which the physics model output is incorporated as an input feature to the data-driven model. Seven architectures are evaluated, trained, and tested on 445,017 samples corresponding to 123 hours of real-world driving data collected across cold, mild, and warm ambient conditions. The best performing architecture is Hybrid TCN-LSTM, with a mean absolute percentage error of 8.35% using only CAN bus signals. Adding the physics-based equilibrium signal as an input feature reduces this to 7.93%, with consistent improvement seen across all seven architectures and all temperature conditions. The physics signal captures ventilation history in a way that raw CAN bus signals do not directly express, and its contribution shows that the two approaches carry different information. The results establish that CAN bus signals alone are sufficient for accurate cabin CO_2 prediction, and that incorporating physical knowledge into the model pushes that accuracy further still.

Keywords: virtual sensing, cabin air quality, deep learning, hybrid physics-informed model, temporal convolutional network, CAN bus, CO_2 prediction.

Acknowledgements

All Praise belongs to the Most High.

This thesis was made possible through the support of several people. Krishna Hariharan's technical knowledge and clarity of thought shaped how this project developed, from the initial problem formulation to the modeling decisions made throughout. During the whole project period, Maria Krantz gave constant coordination and support, both essential to complete the project. To both of them, I am sincerely grateful.

I would also like to thank my examiner, Giovanni Volpe, for his trust and for the freedom to explore and define the direction of this work.I am also grateful to my partner, Dingyuan Wu, for his support and patience throughout this period.

Finally, I would like to thank my colleagues at the Thermal Management department at Volvo Group Truck Technology for sharing their expertise and for being open for the technical aspects of this work.

Abdulrazak Mohamed, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

CAN	Controller Area Network
CO_2	Carbon Dioxide
Conv1D	One-Dimensional Convolutional Neural Network
EWMA	Exponentially Weighted Moving Average
GBDT	Gradient Boosting Decision Tree
GRU	Gated Recurrent Unit
HVAC	Heating, Ventilation, and Air Conditioning
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
NDIR	Non-Dispersive Infrared
OEM	Original Equipment Manufacturer
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
TCN	Temporal Convolutional Network

Nomenclature

Below is the nomenclature of parameters and variables that have been used throughout this thesis.

Parameters

C_{amb}	Ambient CO_2 concentration (ppm)
C_{eq}	Equilibrium CO_2 concentration (ppm)
C_{ex}	CO_2 concentration in exhaled air (ppm)
Q_{ex}	Exhalation flow rate per occupant (L/min)
Q_l	Cabin body leakage flow rate (L/s)
S	Occupant CO_2 generation rate (ppm·L/s)
T_{amb}	Ambient temperature (°C)
V	Cabin volume (m^3)
τ	Cabin time constant (s)

Variables

$C(t)$	Cabin CO_2 concentration at time t (ppm)
$C_{\mathrm{eq}}(t)$	Moving equilibrium concentration at time t (ppm)
h_t	Hidden state of a recurrent network at time t
$Q_l(t)$	Time-varying cabin leakage flow rate (L/s)
x_t	Input feature vector at time t
y_t	Target CO_2 concentration at time t (ppm)
$\hat{y}_t$	Predicted CO_2 concentration at time t (ppm)
$\tilde{y}$	Log-standardized target variable
α	EWMA smoothing factor

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The air quality inside vehicle cabins is an important factor affecting driver alertness, occupant comfort, and occupational health. Among the constituents of cabin air, carbon dioxide (CO_2) is of particular interest because it is produced continuously by occupants and accumulates rapidly in the enclosed space of a modern vehicle. Field measurements show that CO_2 concentrations in passenger cars routinely exceed 2,500 parts per million (ppm) during ordinary driving and can surpass 4,000 ppm under recirculation [1]. These levels are well above the 1,000 ppm threshold commonly linked to cognitive decrements, such as reduced decision-making performance [3]. Professional drivers are at particular risk, as long shift durations in heavy-duty vehicles can lead to recurring exposure to high concentrations that impair safety-critical performance.

These CO_2 concentrations depend on several factors, including ventilation mode, cabin volume, and the number of occupants. Controlled experiments have demonstrated that recirculated airflow consistently produces the highest concentrations, with levels rising proportionally to the number of passengers [6]. In vehicles with windows closed and recirculation active, concentrations can reach 3,000 ppm in under 20 minutes [4]. While fresh air intake can maintain CO_2 at safe levels, drivers often prefer recirculation to block outdoor pollutants or to reduce the energy load on the climate control system [5]. This creates a trade-off between air purity and metabolic CO_2 build-up that modern ventilation systems must manage.

1.1.1 Limitations of Physical CO_2 Sensing

Measuring CO_2 inside vehicle cabins is technically difficult. The most common method is non-dispersive infrared (NDIR) spectroscopy, which uses an infrared light source and a detector. NDIR sensors are selective and stable, but their readings drift with temperature and humidity, requiring compensation algorithms [2]. Some designs use a second detector channel to correct this drift, but this adds cost.

Photo-acoustic spectroscopy is an alternative. Sensors based on this principle use a pulsed light source and a microphone to detect the sound made by CO_2 molecules as they heat up and cool down. They can be made smaller than NDIR sensors because they do not need a long optical path. Huber and Webber showed that a miniaturized photo-acoustic sensor can work in a vehicle over a wide temperature range [13]. However, the signal is weak and the sensor needs good shielding and compensation.

Both sensor types have the same practical problems. They need regular recalibration because the light source ages and the optics get contaminated. Vibration and temperature cycling can cause mechanical failure. For a manufacturer that wants to put CO_2 sensors in every vehicle, the total cost of the sensor, calibration, and replacement is often too high.

1.1.2 Data-Driven Approaches to Virtual Sensing

These limitations have motivated the use of virtual sensing, where the quantity of interest is estimated from other available measurements rather than measured directly with a dedicated sensor. In urban air quality monitoring, machine learning models trained alongside reference instruments can significantly improve the accuracy of low-cost gas sensors and, in some cases, predict pollutant concentrations that no sensor in the system directly measures [12]. The key point is that calibration and virtual estimation are the same learning problem, and that a well-trained model can generalize to unseen conditions.

In indoor environments, Long Short-Term Memory (LSTM) networks have been used as virtual CO_2 sensors in buildings, reaching the accuracy of physical instruments while using only signals already available in the building management system [7]. This demonstrates that when the underlying process has enough structure, a learned model can replace a hardware sensor. However, buildings operate under relatively stable conditions: the Heating, Ventilation and Air Conditioning (HVAC) system runs at near-constant set-points and the main source of variation is occupant density. The cabin of a moving vehicle is fundamentally different. Vehicle speed, blower speed, recirculation ratio, and ambient temperature all change continuously, and the ventilation state can shift from full fresh air to full recirculation within seconds. A virtual sensor for a truck cabin must therefore learn a non-stationary system whose governing parameters shift with every change in driving condition.

Most relevant to this work, a deep learning-based virtual sensor of the cabin temperature of an electric passenger vehicle was developed with CAN bus signals as inputs, comparing recurrent, convolutional and hybrid architectures [19]. The best model, a temporal convolutional network (TCN), achieved a mean absolute percentage error of about 3.5%. This shows that CAN bus signals carry enough information to estimate an internal cabin variable at an accuracy useful for control applications, and that the choice of architecture matters. Cabin temperature, however, is influenced primarily by a single dominant factor (solar load) and responds with relatively slow thermal dynamics. CO_2 concentration, by contrast, responds rapidly to changes in ventilation and is driven by the combined effect of multiple operating parameters simultaneously. The present work extends this line of research from temperature to CO_2, and from a passenger vehicle to a heavy-duty truck cabin, where the wider range of operating conditions makes the estimation problem more challenging.

1.1.3 Towards a virtual CO_2 sensor for truck cabins

The findings reviewed above suggest that a virtual CO_2 sensor for the heavy-duty vehicle cabin is both technically achievable and practically useful. The demonstra-

tion that CAN bus signals alone can support deep-learning-based virtual sensing of internal vehicle variables in a production fleet [9] provides an important methodological reference point. The present work follows a similar approach in terms of data sources and model comparison, but differs in three main ways. First, the target variable is CO_2 concentration rather than temperature. While cabin temperature is governed by heat transfer driven by external conditions and the HVAC system, CO_2 concentration is governed by a mass balance in which the source term comes from the occupants and the main removal mechanisms are ventilation and infiltration.
Second, the application involves sustained long-duration operation. A passenger vehicle trip typically lasts minutes to a few hours, during which the ventilation state may remain relatively stable. A heavy-duty truck, by contrast, operates over days and weeks, crossing multiple climate zones and alternating between driving, idling, and rest periods. Thus, the model must be accurate over much longer time windows, capturing CO_2 dynamics that include shift-long recirculation episodes, overnight dormancy, and cold-start transients following extended stops. This places a stronger demand on the model's ability to learn sustained temporal dependencies.
Third, the model comparison is carried out within a hybrid framework that combines a data-driven model with a physics-based component. A first-principles cabin mass-balance model is used to derive the equilibrium CO_2 concentration under current ventilation conditions, which is then provided as an additional input to the learned model. This type of hybrid approach has not been widely explored in the heavy-duty vehicle context. The signals available on the CAN bus of a modern truck, including outside temperature, vehicle speed, blower level, recirculation flap position, and related actuator states, encode the boundary conditions that drive the CO_2 mass balance inside the cabin. This study investigates whether a deep learning model can map CAN bus signals to cabin CO_2 concentration with sufficient accuracy to serve as a software-based alternative to a dedicated physical sensor. The model families most commonly applied to this type of time-series problem (LSTM, gated recurrent units(GRU), TCN, and hybrid TCN-recurrent architectures) differ in their structural assumptions and generalization behaviour, yet their relative performance on heavy-duty vehicle cabin data has not been systematically evaluated.

1.2 Aim

The aim of this thesis is to develop and evaluate a virtual CO_2 sensor for heavy-duty truck cabins that predicts cabin CO_2 concentration from signals available on the vehicle CAN bus, without requiring a dedicated physical sensor. To achieve this, three approaches are investigated: a purely data-driven deep learning model, a physics-based model derived from the cabin CO_2 mass balance, and a combined approach in which the physics model output is used as an additional input to the data-driven model. The following research questions guide the work:

1. To what extent can cabin CO_2 concentration be accurately predicted from CAN bus signals alone, and which deep learning architecture performs best for this task?
2. How well the physics-based mass balance model predicts cabin CO_2 concentration under real driving conditions and its limitations?

3. Does incorporating the physics model output as an input feature improve the predictive performance of the data-driven model compared to using CAN bus signals alone?

1.3 Thesis outline

The remainder of this report is organized as follows. Chapter 2 presents the theoretical background, including HVAC working principles, the physics of cabin CO_2 accumulation, as well as the deep-learning architectures used in this study. Chapter 3 describes the collected data, the CAN-bus signals used as model inputs, the pre-processing steps performed, and the methods used to develop, train and evaluate the virtual sensor models. Chapter 4 reports the results obtained and Chapter 5 discusses these results in relation to the research questions and the existing literature. Chapter 6 provides the conclusions of the study and suggests directions for future work.

2
Theory

This chapter presents the theoretical background needed for the work in this thesis. Section 2.1 covers the working principle of the HVAC system. Section 2.2 discusses the deep learning architectures for time-series regression, beginning with simple recurrent neural networks and progressing to gated architectures (GRU and LSTM) and convolutional architectures (TCN and hybrid TCN-recurrent models). Section 2.3 develops the first-principles mass-balance model describing the change in the CO_2 concentration in a vehicle cabin.

2.1 HVAC System Working Principle

The HVAC unit performs a number of processes on the air before delivering it to the cabin. Understanding this sequence is important, as it can be used to read the CAN bus signals that drive the virtual CO_2 sensor, since each of the stages in air conditioning adds a measurable signature to the vehicle data stream.

2.1.1 Air Intake and Mixing

The air enters the HVAC unit via one of two routes: from the external environment via the fresh air intake, or from the cabin interior via the recirculation duct, as shown in illustrated in Figure 2.1. The recirculation flap, a motorized damper controlled by the climate management system, determines the proportion of recirculated air that enters the HVAC [14]. In partial recirculation mode, both streams are drawn in together and mixed before entering the conditioning stages. In full recirculation mode, the fresh air intake is completely closed and only cabin air is processed. The mixed airstream then passes through a particulate filter that removes dust, pollen, soot and other contaminants. In vehicles equipped with activated carbon filtration, gaseous pollutants such as volatile organic compounds (VOCs) are additionally adsorbed, improving the chemical quality of the cabin air supply [14].

2.1.2 Cooling and Dehumidification

In cooling mode, the filtered airstream is directed over the evaporator, a heat exchanger through which refrigerant circulates at low pressure and temperature. The warm air from the cabin or outside passing over the cold evaporator surface exchanges sensible heat with the refrigerant, bringing down the air temperature. At the same time, if the air temperature drops below its dew point, condensation of

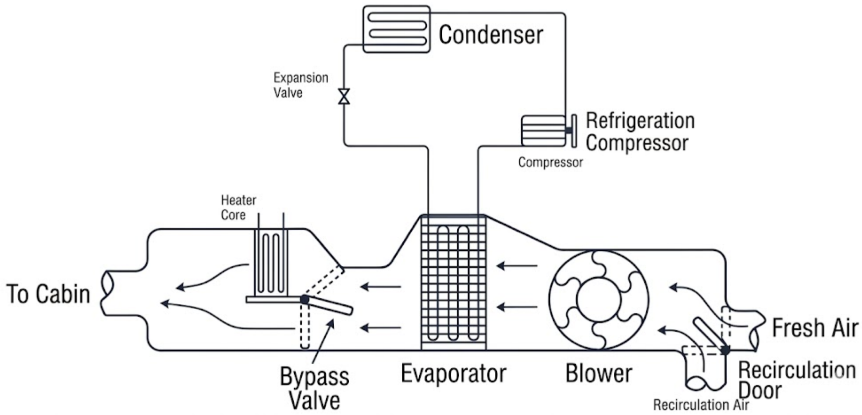

Figure 2.1: Schematic of a vehicle HVAC system, showing the fresh air and recirculation intake pathways, blower, evaporator, heater core, bypass valve, and refrigeration circuit. Adapted from [8]

water vapour occurs on the evaporator fins and is drained away, reducing the absolute humidity of the supply air [14]. This dehumidification is important not only for thermal comfort but also for preventing mist formation on glazing surfaces.

2.1.3 Heating

After passing through the evaporator, the air is directed toward the heater core. In conventional vehicles, the heater core is supplied with hot engine coolant. In battery electric trucks, engine waste heat is largely absent, so heating is instead provided by auxiliary PTC (Positive Temperature Coefficient) electric resistance heaters or a heat pump [14]. The amount of heating applied is governed by either the coolant flow rate (water flow type) or the position of a mixing flap that blends the heated airstream with a cooler bypass stream (air mix type) [15]. The air mix approach is more dynamically responsive, since the flap can be repositioned rapidly, whereas the water flow type relies on the slower thermal inertia of the coolant circuit.

2.1.4 Air Distribution

The conditioned air is distributed to the cabin through a network of ducts and vents, with the distribution pattern determined by the active operating mode [14]. In *ventilation mode*, air is directed through the upper dashboard vents for cooling comfort in mild to warm conditions. In *heating mode*, air is routed through the floor vents, where it rises by natural convection and creates a vertical temperature gradient with warm feet and a cooler head, a pattern that is well-suited to occupant comfort [14]. This is the dominant mode during cold weather driving, which in Scandinavian operating conditions represents a substantial portion of annual truck operation. In *demisting and defrosting mode*, hot dehumidified air is directed at the wind-shield and side windows to raise the glazing surface temperature above the local dew point [14]. In *bi-level mode*, air is supplied through both floor and upper vents simultaneously for intermediate comfort conditions.

2.1.5 Control Architecture

Modern HVAC systems are based on automatic climate control where the driver specifies a desired cabin temperature and the system autonomously adjusts all actuators including fan speed, recirculation flap, mixing flap, and compressor state to meet this target while balancing thermal comfort, energy efficiency, glazing visibility and air quality [15]. In electric trucks heating demand at low ambient temperatures can be a large fraction of total energy consumption, creating a strong incentive to maximize recirculation [14]. However, high recirculation fractions result in CO_2 accumulation and humidity build-up [21]. The control system therefore continuously navigates this trade-off, producing a time-varying recirculation strategy that depends on ambient temperature, cabin temperature error, occupancy, and glazing conditions. This dynamic behaviour is a key reason why cabin CO_2 concentration cannot be reliably predicted from a single static rule, motivating the data-driven approach developed in this thesis.

2.2 Deep Learning Models for Time-Series Regression

The control dynamics described above have an important consequence for the modelling approach: the cabin CO_2 does not change instantaneously with the ventilation state. When the recirculation flap is closed or the blower speed is changed, the concentration changes slowly over a time period that depends on the volume of the cabin and the current rate of air exchange. This means that two cabins operating at identical instantaneous recirculation and blower settings can have very different CO_2 concentrations depending on how long those settings have been in effect. A static regression model that maps only current CAN bus signals to a concentration estimate would not be able to make this distinction and would result in large errors during and just after ventilation transitions. A time-series model, by contrast, can use the sequence of past observations to infer the current dynamic state, effectively learning the temporal lag inherent in the system.
The task is therefore formulated as a multivariate time-series regression problem, which requires architectures capable of exploiting temporal context. Given an input sequence $\{x_1, x_2, \ldots, x_T\}$, with $x_t \in \mathbb{R}^d$, the goal is to learn a parametrized mapping $f_\theta : \mathbb{R}^{T\times d} \rightarrow \mathbb{R}^T$ that produces an estimate $\hat{C}_c(t)$ of the cabin concentration at each time step. This motivates the use of recurrent, convolutional, and hybrid architectures, introduced in the sections that follow.

2.2.1 Simple recurrent neural network

The key idea behind a recurrent neural network (RNN) is the introduction of a feedback loop: the output of the network at one step is fed back as an additional input at the next step, giving the network a form of memory. A useful analogy is a comb filter in signal processing, where the output at each step is a weighted average of the current input and the previous output. An RNN generalizes this by replacing fixed weights with learnable parameters and by applying a non-linear activation function. At each time step t, the simple (or "vanilla") RNN reads an input x_t and a previous hidden state h_{t-1}, and produces a new hidden state h_t according to [23]

$$h_t = \phi(W_h h_{t-1} + W_x x_t + b)\,, \tag{2.1}$$

where W_h, W_x, and b are learnable parameters and ϕ is a non-linear activation, typically the hyperbolic tangent. The hidden state h_t summarizes all the information seen in the input up to time t, and a per-step output $\hat{y}_t$ is obtained from h_t through an additional affine layer. The same parameters W_h, W_x, b are shared across all time steps. Figure 2.2 shows the network unrolled in time, which makes the recurrent structure explicit and motivates the term "back-propagation through time" for the training procedure.

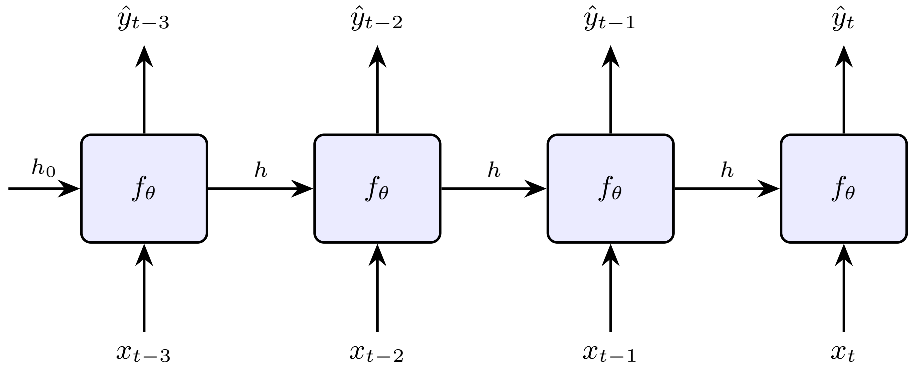

Figure 2.2: A recurrent neural network unrolled in time. At each step the cell f_θ takes the current input and the previous hidden state, produces a new hidden state, and emits an output. The same parameters θ are shared across all time steps.

In principle the simple RNN can represent arbitrary temporal relationships, but in practice it is difficult to train when the relevant dependencies span many time steps. As discussed above, cabin CO_2 concentration responds slowly to changes in ventilation state, and the model must therefore retain information about conditions that occurred minutes earlier to accurately predict the current concentration. This requirement of long-range memory reveals a fundamental limitation of the simple RNN: repeated multiplication by the recurrent weight matrix during back-propagation through time causes the gradient to either decay exponentially or grow without bound, a phenomenon referred to as vanishing or exploding gradient problem [24]. The vanishing-gradient case is the more common of the two and prevents the network from learning precisely the kind of long-range correlations that cabin CO_2 dynamics demand. The architectures introduced in Sections 2.2.3 and 2.2.4 were specifically designed to mitigate this problem.

2.2.2 Stacked recurrent neural networks

A single recurrent layer is able to model the temporal dependencies, but is limited in the complexity of the feature representations it can build at a time. Stacking multiple recurrent layers on top of one another increases the depth of the network and allows each layer to operate on progressively higher-level representations of the input [25]. In a two-layer stacked RNN, the hidden state of the lower layer at each time step serves as the input to the upper layer at the same time step,

$$h_t^{(1)} = \phi\Big(W_h^{(1)}\, h_{t-1}^{(1)} + W_x^{(1)}\, x_t + b^{(1)}\Big)\,, \tag{2.2}$$

$$h_t^{(2)} = \phi\Big(W_h^{(2)}\, h_{t-1}^{(2)} + W_x^{(2)}\, h_t^{(1)} + b^{(2)}\Big)\,, \tag{2.3}$$

and the final output is obtained from the top layer. An important property of stacked recurrent networks is that each intermediate layer passes its entire output sequence to the next layer, not just the final hidden state; this allows every layer to process the full time dependent sequence and discover temporal patterns at multiple levels of abstraction [25]. This structure naturally generalises to more than two layers, as shown in Figure 2.3. Stacking adds a depth dimension that complements the temporal dimension and has been observed to improve performance on a range of sequence-modelling tasks. However, depth comes at a cost: gradients now have to propagate not only backward through time but also through each additional layer, which amplifies the vanishing and exploding gradient problem already inherent in recurrent networks. Deeper stacked RNNs are therefore harder to train and more sensitive to initialization, which is why the stacking principle yields the greatest benefit when applied to gated architectures, described in the sections that follow, where more stable gradient flow in the temporal direction provides a stronger foundation for additional depth.

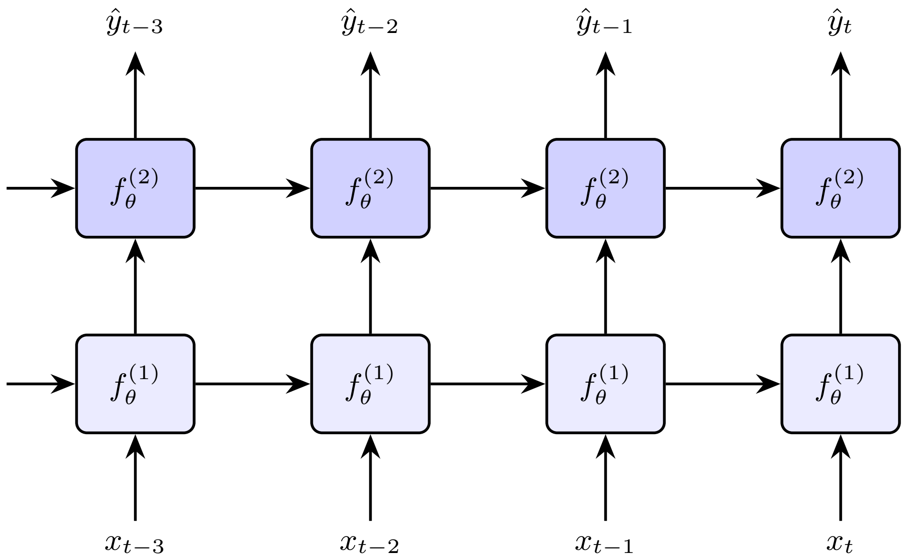

Figure 2.3: A two-layer stacked recurrent neural network. The hidden state of the lower layer at each time step is fed as the input to the upper layer at the same time step, while each layer maintains its own recurrent connection in time.

2.2.3 Gated recurrent unit

The gated recurrent unit (GRU) addresses the vanishing-gradient problem of the simple RNN by introducing two learnable gates that control how much of the previous hidden state is retained and how much new information is incorporated at each time step [26]. Given an input x_t and a previous hidden state h_{t-1}, the GRU computes an update gate z_t and a reset gate r_t,

$$\begin{aligned} z_t &= \sigma(W_z\, x_t + U_z\, h_{t-1} + b_z)\,, && (2.4)\\ r_t &= \sigma(W_r\, x_t + U_r\, h_{t-1} + b_r)\,, && (2.5) \end{aligned}$$

where σ denotes the logistic sigmoid function and the weight matrices and bias vectors are learnable parameters. The reset gate r_t decides which components of the previous hidden state should contribute to a candidate hidden state $\tilde{h}_t$,

$$\tilde{h}_t = \tanh(W\, x_t + U\,(r_t \odot h_{t-1}) + b)\,, \tag{2.6}$$

where $\odot$ denotes element-wise multiplication. The new hidden state is then formed as a convex combination of the previous hidden state and the candidate, with the update gate z_t controlling the mixing,

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \tag{2.7}$$

When z_t is close to zero, the previous hidden state is preserved almost unchanged, which provides a direct path for gradients to flow backwards in time and stabilizes training over long sequences. When z_t is close to one, the new candidate replaces the previous state. The GRU therefore has a learnable mechanism for balancing memory retention against the incorporation of new information, while keeping the parameter count and computational cost lower than those of the LSTM described in the next section. Because gating increases the number of trainable parameters relative to the simple RNN, dropout is commonly applied during training to reduce the risk of over-fitting. Dropout randomly deactivates a subset of neurons in each training step, forcing the remaining neurons to develop more robust representations; all neurons are fully active during inference [22].

2.2.4 Long short-term memory

The long short-term memory (LSTM) cell predates the GRU and is the classical solution to the vanishing-gradient problem in recurrent networks [27]. The LSTM augments the hidden state h_t with a separate cell state c_t, which acts as a protected memory that can be selectively written to or erased through four gates: an input gate i_t, a forget gate f_t, a cell gate $\tilde{c}_t$ (also called the candidate gate), and an output gate o_t [27]. Given an input x_t and the previous hidden and cell states h_{t-1} and c_{t-1}, the LSTM performs the following computations,

$$
\begin{aligned}
f_t &= \sigma(W_f\, x_t + U_f\, h_{t-1} + b_f)\,, && (2.8)\\
i_t &= \sigma(W_i\, x_t + U_i\, h_{t-1} + b_i)\,, && (2.9)\\
o_t &= \sigma(W_o\, x_t + U_o\, h_{t-1} + b_o)\,, && (2.10)\\
\tilde{c}_t &= \tanh(W_c\, x_t + U_c\, h_{t-1} + b_c)\,, && (2.11)
\end{aligned}
$$

after which the cell and hidden states are updated as

$$
\begin{aligned}
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, && (2.12)\\
h_t &= o_t \odot \tanh(c_t). && (2.13)
\end{aligned}
$$

The forget gate decides which components of the previous cell state should be preserved, the input gate decides which components of the candidate $\tilde{c}_t$ should be written, and the output gate controls how much of the resulting cell state is exposed in the hidden state h_t. The cell-state update in equation (2.12) consists only of element-wise products and an addition, with no non-linearity along the temporal path; this is what allows information to be carried across many time steps without the gradient signal degrading.
The LSTM and GRU achieve similar goals through different parameterizations. The LSTM maintains an explicit, separately gated cell state in addition to the hidden state, while the GRU collapses these into a single hidden state and uses fewer gates and fewer parameters. Empirically the two architectures perform comparably on a wide range of sequence-modeling tasks, with the relative ranking depending on the specific application, the amount of training data available, and the way hyperparameters are selected [28]. Both are evaluated in this thesis.

2.2.5 Training and loss function

All of the architectures described above are trained by minimizing a loss function $\mathcal{L}(\theta)$ over a labeled training set, where θ collects all learnable parameters of the network. For the regression problem considered here, the mean absolute error (MAE) between the predicted and reference cabin CO_2 concentrations is used as the training criterion,

$$
\mathcal{L}(\theta) = \frac{1}{N\,T} \sum_{n=1}^{N} \sum_{t=1}^{T} \left| \hat{C}_c^{(n)}(t) - C_c^{(n)}(t) \right|, \tag{2.14}
$$

where N is the number of training sequences and T is the sequence length. The MAE treats all prediction errors equally regardless of their magnitude, which makes it more robust to occasional large deviations than the mean squared error and gives the loss value a direct physical interpretation in the units of the target quantity. Gradients of $\mathcal{L}$ with respect to θ are computed by back-propagation, and for recurrent networks specifically by back-propagation through time, in which the unrolled computational graph is traversed from the final time step backwards to the first. The parameters are then updated using the Adam optimizer [29], which adapts the per-parameter learning rate based on running estimates of the first and second moments of the gradient. The specific values of the optimizer hyper-parameters and the regularization strategy used in this work are described in Chapter 4.

2.2.6 Temporal Convolutional Network

Recurrent networks are the most common approach to sequence modelling, but convolutional architectures can perform as well or better on certain tasks [32]. The Temporal Convolutional Network (TCN) is a convolutional architecture designed specifically for sequence modelling [32, 33]. It combines two key ideas: dilated causal convolution and residual connections.

2.2.6.1 Dilated Causal Convolution

Causal convolution ensures that the output at time step t depends only on current and past inputs. A dilated causal convolution can be written as

$$h_t = \sigma\left(\sum_{i=0}^{k-1} W_i x_{t-d\,i} + b\right),$$

where k is the kernel size, d is the dilation factor, W_i are the convolution weights, b is the bias term, and $\sigma(\cdot)$ is the activation function. When $d = 1$, this reduces to a standard causal convolution. When $d > 1$, the convolution skips time steps, allowing the network to look further into the past without increasing the kernel size. Stacking layers with exponentially increasing dilation factors gives the TCN a large effective receptive field with relatively few parameters.

2.2.6.2 Residual Block

Each TCN block contains two dilated causal convolutional layers, each followed by a ReLU activation and dropout. A residual connection adds the block input to the block output, which stabilizes training and allows deeper networks to be constructed. The output of block l is

$$\mathbf{h}^{(l)} = \mathrm{ReLU}\left(F(\mathbf{h}^{(l-1)}) + \mathbf{P}^{(l)}\mathbf{h}^{(l-1)}\right),$$

where $F(\cdot)$ represents the two convolutional layers and $\mathbf{P}^{(l)}$ is a 1×1 convolution used to match dimensions when the number of channels changes.

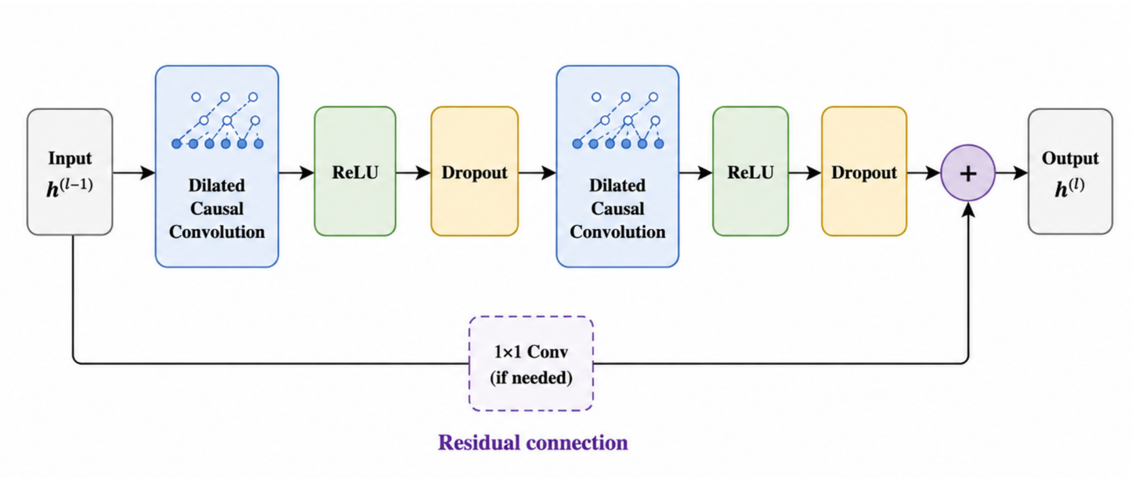

Figure 2.4: Architecture of a TCN residual block.

2.2.7 Hybrid TCN-Recurrent Models

The TCN captures local temporal patterns effectively, but recurrent models are better suited to modelling sequential state transitions. A hybrid TCN-recurrent model combines both approaches: the TCN acts as a temporal feature extractor, and its output is passed to an LSTM or GRU which models the sequential evolution of those features and produces the final prediction.

Several studies have adopted this approach. Dudukcu et al. [34] proposed a hybrid TCN-RNN architecture for chaotic time-series prediction, where the TCN extracts low-level features and recurrent layers capture temporal information. Zhou et al. [35] developed a TCN-GRU model for short-term bike-sharing demand prediction, and Xiang et al. [36] proposed a TCN-ECANet-GRU model for photovoltaic power forecasting. These studies show that combining TCN with recurrent layers is an effective way to model both local temporal patterns and longer sequential dependencies.

In this thesis, hybrid TCN-LSTM and TCN-GRU models were implemented and evaluated. The results in Chapter 4 show that these hybrid models improve prediction performance compared to using either a TCN or a recurrent model alone.

2.3 Cabin CO_2 Mass Balance

The vehicle cabin can be treated as a single, well-mixed control volume in which CO_2 is generated by the occupants and removed by ventilation and infiltration [21]. This assumption means that the CO_2 concentration is considered spatially uniform inside the cabin, so that a single cabin concentration can be used to represent the overall cabin air state.
A schematic of the cabin air system is shown in Figure 2.5. Outside air with ambient CO_2 concentration C_{amb} enters the cabin through ventilation and leakage paths, while cabin air leaves through body vents and other leakage openings. Under normal operation, the incoming and outgoing volumetric flow rates are assumed to be balanced. Their common effective value is denoted as the cabin body leakage flow rate Q_l. The occupants act as an internal CO_2 source. For n occupants, the source term is defined as

$$S = n\, C_{ex} Q_{ex}, \tag{2.15}$$

where C_{ex} is the CO_2 concentration in exhaled air and Q_{ex} is the exhalation flow rate per occupant.

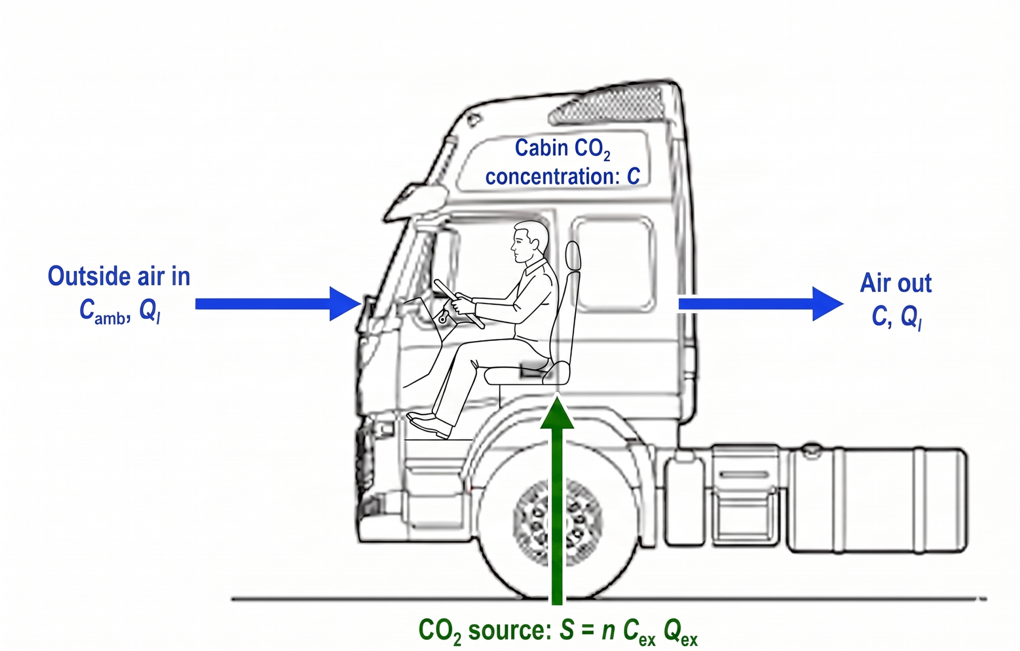

Figure 2.5: Schematic of the cabin air system used as the basis for the CO_2 mass-balance model. Outside air enters the cabin with ambient concentration C_{amb}, cabin air leaves through body vents and leakage paths, and occupants act as an internal CO_2 source. Cabin geometry adapted from [10].

Applying conservation of mass to CO_2 inside the cabin gives

$$\frac{\mathrm{d}m_c}{\mathrm{d}t} = S + C_{\text{amb}} Q_l - C Q_l, \tag{2.16}$$

where m_c is the amount of CO_2 in the cabin, C is the cabin CO_2 concentration, and Q_l is the effective cabin body leakage flow rate. The first term on the right-hand side represents CO_2 generated by occupant respiration, the second term represents ambient CO_2 entering the cabin, and the third term represents CO_2 removed from the cabin by outgoing air.

Using the relation $C = m_c/V$, where V is the cabin volume, Equation (2.16) can be rewritten as a concentration-based ordinary differential equation:

$$V\frac{\mathrm{d}C(t)}{\mathrm{d}t} = S + C_{\mathrm{amb}}Q_l(t) - C(t)Q_l(t). \tag{2.17}$$

Equivalently,

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = \frac{S}{V} - \frac{Q_l(t)}{V}C(t) + \frac{Q_l(t)}{V}C_{\mathrm{amb}}. \tag{2.18}$$

This equation describes the dynamic balance between occupant CO_2 generation and dilution by outside air. The source term S/V increases the cabin concentration, while the remaining two terms describe the exchange between cabin air and ambient air.

Two limiting cases are useful for interpreting the model. When the vehicle is stationary and the ventilation fan is switched off, the leakage flow Q_l is approximately zero. In this case, Equation (2.17) reduces to a linear accumulation model:

$$C(t) = C_0 + \frac{S}{V}t, \tag{2.19}$$

where C_0 is the initial cabin CO_2 concentration. This case shows that, without ventilation or leakage, the cabin concentration increases approximately linearly with time due to occupant respiration.

During normal driving, $Q_l > 0$. If Q_l is assumed to remain constant over a short time interval, Equation (2.17) has the first-order solution

$$C(t) = (C_0 - C_{\mathrm{eq}})\exp\Big(-\frac{Q_l}{V}t\Big) + C_{\mathrm{eq}}, \tag{2.20}$$

where the equilibrium concentration is

$$C_{\mathrm{eq}} = C_{\mathrm{amb}} + \frac{S}{Q_l}. \tag{2.21}$$

The equilibrium concentration represents the CO_2 level that the cabin would approach if the current ventilation condition remained unchanged for a sufficiently long time. If the current cabin concentration is below C_{eq}, the internal source dominates and the cabin CO_2 concentration tends to increase. If the current cabin concentration is above C_{eq}, ventilation dominates and the concentration tends to decrease.

The speed of this response is determined by the cabin time constant

$$\tau = \frac{V}{Q_l}. \tag{2.22}$$

A larger cabin volume or smaller leakage flow leads to a larger time constant, meaning that the cabin CO_2 concentration changes more slowly. Conversely, a larger leakage flow results in faster dilution and a shorter response time.

Under steady-state conditions, the cabin concentration no longer changes with time, so $\mathrm{d}C/\mathrm{d}t = 0$. In this case, the measured cabin concentration can be approximated by the equilibrium concentration. Rearranging Equation (2.21) gives

$$Q_l = \frac{S}{C_{\mathrm{eq}} - C_{\mathrm{amb}}}. \tag{2.23}$$

This expression is important because it allows the effective leakage flow rate Q_l to be inferred from cabin CO_2 measurements when the cabin is close to steady state. In real driving data, exact steady-state conditions are rarely observed because vehicle speed, blower operation, recirculation setting, and ambient conditions vary continuously. However, if a time segment has sufficiently small CO_2 variation and slope, its local mean cabin concentration can be treated as an approximation of C_{eq}. This provides the theoretical basis for identifying quasi-steady-state points and using them to estimate Q_l.
A practical complication is that the occupant source term S is not directly measured on the vehicle. It depends on the number of occupants, breathing rate, and activity level. In this study, the truck is assumed to be occupied by the driver only for most driving sessions, which is representative of typical heavy-duty truck operation. The per-occupant product $C_{ex}Q_{ex}$ is taken from the literature as a nominal value for adults at rest [21]. Another uncertainty is that Q_l is not constant in real driving. Previous work has shown that cabin body leakage depends on operating conditions such as vehicle speed, ventilation fan speed, and HVAC duct geometry [30]. In addition, the recirculation ratio affects the effective amount of fresh outside air introduced into the cabin, and therefore influences the apparent ventilation state [31]. Therefore, the mass balance model described in this section captures the correct functional form of the cabin CO_2 dynamics, but its parameters are subject to uncertainty under real driving conditions. This motivates the use of a learned model that takes CAN-bus signals as inputs and is supported by, rather than fully constrained by, the physics-based formulation.

3 Methods

3.1 Overview

This chapter explains the data processing procedures and the three methods used in this thesis for predicting the CO_2 concentration inside the vehicle using signals from the CAN bus. The methods for collecting raw driving data, classifying sessions according to temperature conditions, and defining the target variable are detailed in Section 3.2. Outlier removal, missing value processing, feature scaling, target transformation, and the generation of fixed-length input sequences are all covered in Section 3.3, which is inclusive of all pre-processing processes that are common to all approaches. The first approach, described in Section 3.4, is a data-driven model that uses deep learning to directly predict CO_2 from CAN bus signals. The second approach is detailed in Section 3.5. It is a model grounded in physics that uses numerical solutions of the cabin mass balance equation to mimic the concentration of CO_2 in the cabin. In the final approach, which is detailed in Section 3.6, a combined physics-informed prediction pipeline is created by using the output of the physics-based model as an extra input feature to the data-driven model. Lastly, the evaluation criteria used to compare all three approaches and the post-processing used to smooth model predictions are detailed in Sections 3.7 and 3.8.

3.2 Data Collection

The data were obtained from an instrumented Volvo heavy-duty truck that was operated in a range of real-world road environments. Considering the mixed-duty cycles typical of heavy-duty long-haul and distribution transport, sessions were conducted on public roads encompassing highways, rural areas, and major cities. A part of the vehicle's power-train CAN data, which is normally confidential in commercial trucks, had to be accessed in order to carry out the research. Since the Volvo gives structured CAN data that can be read with the right instruments, the diagnostic interface of the vehicle allowed access to the necessary signals. However, the signals included in this study are comprised of generic HVAC and ambient data, which can be found on platforms from numerous manufacturers for heavy-duty trucks. The suggested approach is thus theoretically transferable to different truck platforms, provided that the appropriate CAN signals can be read. The original signal sampling periods differed among channels, but the CAN signals were captured at a consistent 1 Hz rate. To avoid false averaging effects and maintain temporal consistency, the data was resampled to this common rate. The truck cabin was equipped

with two CO_2 sensors to supply the measurements for the ground-truth target. The final dataset contains 445,017 samples, corresponding to approximately 123 hours of driving, distributed across the three temperature conditions as summarized in Table 3.1.
To ensure that the models are exposed to the full range of ambient conditions encountered in commercial vehicle operation, data were collected across three distinct climatic settings. Cold condition sessions captured sub-zero temperatures down to approximately $-20\,°C$, where the engine and cabin require significant warm-up time and the heating system operates at full capacity for extended periods. Warm condition sessions covered high ambient temperatures with strong solar load, causing the HVAC system to operate predominantly in recirculation mode and leading to sustained CO_2 accumulation. Mild condition sessions represented moderate ambient temperatures between 0 and $20\,°C$ with intermediate ventilation requirements. The air conditioning system was operated in different modes across sessions, including both automatic and manual control, to capture the range of ventilation strategies a driver may apply. Most driving sessions were conducted with the driver as the sole occupant, which is representative of typical heavy-duty vehicle operation.
Sessions were grouped into three ambient temperature conditions based on the ambient temperature, as defined in Table 3.1. These conditions reflect distinct HVAC operating regimes: at cold temperatures the system operates primarily in heating mode with constrained fresh-air admission, while at warm temperatures elevated recirculation demand accelerates CO_2 build-up. Sessions that did not belong to a clear driving regime which include parking, lunch stops, and overnight sleep modes were excluded from model training and evaluation, as they represent fundamentally different thermal and ventilation dynamics. Consequently, the virtual sensor developed in this thesis is designed for operation under active driving conditions and may not be applicable to stationary periods when the cabin is unoccupied or when the HVAC system is running in a non-driving mode.

Condition Label	**Temperature Range (T_{amb})**	**Sample Count**
`driving_cold`	$< 0\,°C$	85 567
`driving_mild`	$[0\,°C, 20\,°C]$	169 636
`driving_warm`	$> 20\,°C$	189 814
Total		445 017

Table 3.1: Ambient temperature conditions and sample distribution.

3.2.1 Target Variable

The target variable for all models is the mean cabin CO_2 concentration, computed as the arithmetic mean of the two physical sensor readings introduced in Section 3.2. A single sensor placed at a fixed location may not be representative of the overall air quality experienced by the occupants, since CO_2 concentration varies spatially depending on occupant position, ventilation outlet placement, and recirculation patterns. The target variable is defined as follows:

$$y_t = \frac{s_t^{(1)} + s_t^{(2)}}{2} \tag{3.1}$$

where $s_t^{(1)}$ and $s_t^{(2)}$ are the readings of the two physical sensors at time t.

The distribution of the target variable differs substantially across the three temperature conditions, reflecting the different HVAC operating modes associated with each. Figures 3.1, 3.2, and 3.3 show the normalized target distributions for the cold, mild, and warm conditions respectively.

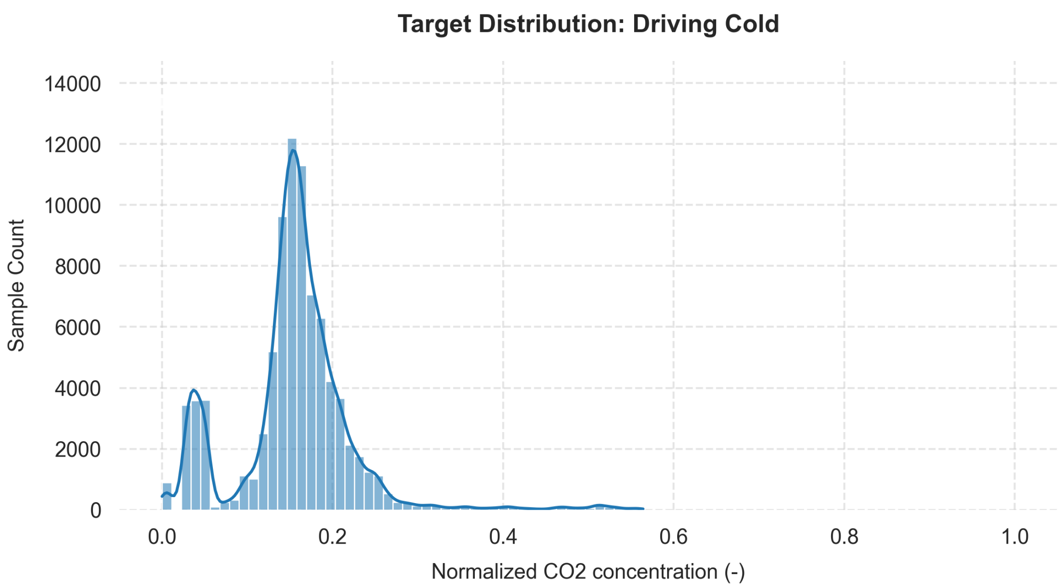

Figure 3.1: Target CO_2 distribution across the driving cold condition ($T_{\text{amb}} < 0\,°\text{C}$).

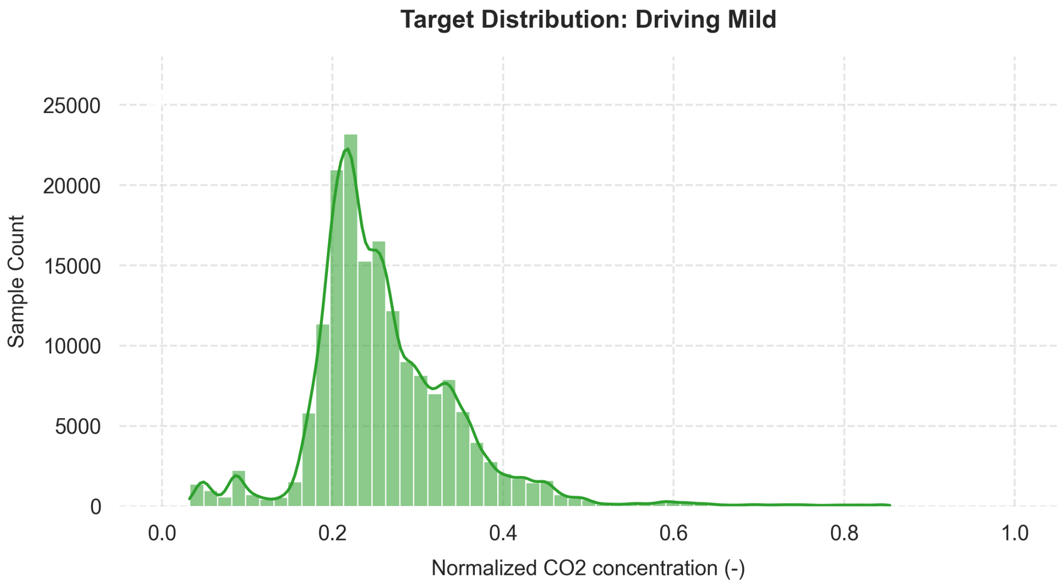

Figure 3.2: Target CO_2 distribution across the driving mild condition ($0\,°C \leq T_{amb} \leq 20\,°C$).

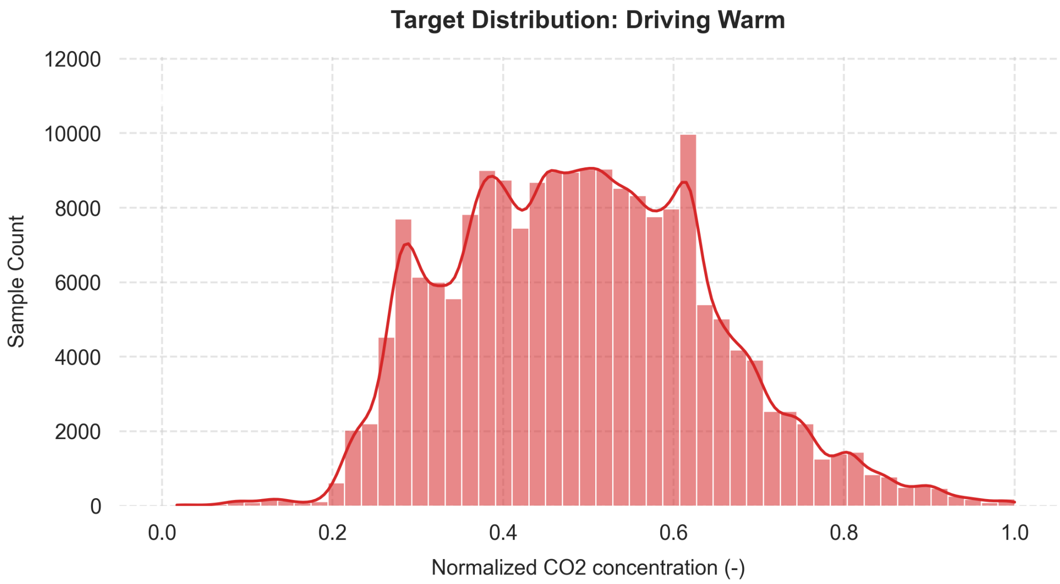

Figure 3.3: Target CO_2 distribution across the driving warm condition ($T_{amb} > 20\,°C$).

In the cold condition, the distribution is narrow and concentrated at low concentrations, reflecting the frequent use of fresh-air intake for windscreen demisting and cabin heating, which continuously dilutes cabin CO_2. The distribution shifts to higher concentrations and acquires a longer tail to the right for the mild condition, indicating that moderate concentrations dominate but episodes of elevated CO_2 are produced during periods of sustained recirculation. In the warm condition, the distribution is broad and shifted substantially toward higher concentrations, consistent with recirculation-dominated HVAC operation. At high ambient temperatures, the system prioritises cooling efficiency over fresh-air admission, which limits CO_2 dilution and drives sustained accumulation throughout the session [1, 21].

3.3 Data Pre-Processing

The quality and reliability of the data are essential for developing a data-driven prediction system. All three approaches underwent the same pre-processing steps to guarantee the data's reliability for model training.

Outlier removal was initially performed separately for each session. The CO_2 sensors did not show any systematic drift but some implausible readings were observed from time to time, due to CAN network transmission problems and sensor initialisation at the beginning of a session. Samples below 460 ppm were removed since an occupied cabin will always sustain CO_2 above the outdoor ambient level of approximately 420 ppm [21]. Our study only focuses on occupied cabin conditions during driving, where the cabin CO_2 concentration is above the ambient level. Therefore our model cannot be expected to perform in unoccupied cabins, near-ambient CO_2 conditions or stationary truck operation. Further, readings that that deviate unrealistically between consecutive time steps were treated as transmission faults and discarded. These outliers typically appeared as isolated single-sample spikes that were physically inconsistent with the surrounding measurements, given that cabin CO_2 concentration changes slowly under normal driving conditions. Small isolated gaps remaining after removal were imputed using the nearest valid observation during training only, and rows with unrecoverable missing values in the target were discarded.

All input features were then standardized to zero mean and unit variance using Standard Scaler fitted exclusively on the training set and applied without refitting to the validation and test sets, preventing any information leakage from held-out data [16].

The target variable was log-standardized before training. As seen in Figures 3.1, 3.2, and 3.3, the CO_2 distribution is right-skewed, with a long tail of high-concentration values. When a target distribution is skewed, the training loss is disproportionately influenced by the few large residuals in the tail, which can push the model to prioritize those rare extreme cases at the expense of typical operating conditions [18]. A logarithmic transformation is a standard approach to address this: it compresses large values while spreading out small ones, making the distribution more symmetric and stabilizing the variance across the range of the target [17]. The transformed values are then standardized to zero mean and unit variance so that the target is on the same numerical scale as the input features, which improves gradient flow during training [16].

The full transformation is defined as:

$$\tilde{y} = \frac{\log(y) - \mu_{\log}}{\sigma_{\log}} \tag{3.2}$$

where y is the original CO_2 concentration in ppm, and $\mu_{\log}$ and $\sigma_{\log}$ are the mean and standard deviation of $\log(y)$ computed from the training set only, ensuring no information from the validation or test sets influences the transformation. At inference time, predictions in the transformed space are inverted back to the original ppm scale via:

$$\hat{y} = \exp\Big(\hat{\hat{y}} \cdot \sigma_{\log} + \mu_{\log}\Big) \tag{3.3}$$

Finally, the data were structured into sequences to preserve the time-series characteristics of the CO_2 signal, in which past observations directly influence future concentration. A sliding-window technique was used, where a fixed-length sequence of past measurements is used to predict a future value. These sequences were then processed by a recurrent neural network. This approach allows the model to learn the temporal patterns in ventilation behaviour and CO_2 accumulation over time, rather than treating each time step in isolation.

After testing several window lengths, 120 s was selected as the input window to predict the CO_2 value 30 s ahead. It provided the best trade-off between capturing the relevant CO_2 dynamics, maintaining training stability, and remaining computationally manageable. Cabin CO_2 responds slowly to changes in ventilation; the concentration builds up and dissipates over timescales of tens of seconds to minutes, governed by the cabin air exchange time constant $\tau = V/Q_l$. Shorter windows failed to capture these delayed responses, while longer windows degraded performance by spanning heterogeneous driving conditions within a single sequence.

3.4 Approach I: Data-Driven Model

The first approach uses deep learning to predict cabin CO_2 concentration directly from CAN bus signals, without any explicit physical model. The model learns the relationship between HVAC signals, ambient conditions, and CO_2 concentration purely from data.

3.4.1 Feature Selection

Feature selection was based on the CO_2 mass-balance model of the vehicle cabin, as detailed in Section 2.3. The primary adjustable parameter is the fresh-air exchange rate Q_l: any signal affecting Q_l is causally relevant to the predicted target. Based on this, seven feature groups covering ambient conditions, recirculation, blower, humidity, air distribution, and vehicle dynamics were retained from the CAN bus signals, giving a total of **19 base features**. These are summarized in Table 3.2.

Group	**Key signals**	**Physical relevance**
Ambient	`Ambient Temperature`	Determines the HVAC operating regime. At low ambient temperatures, the system prioritizes heating and limits fresh-air intake, causing CO_2 to accumulate. At high temperatures, heavy recirculation reduces cooling load with the same effect.
Recirculation	`Recirculation ratio`, `Fresh air fraction`	The recirculation flap determines how much cabin air is re-conditioned versus replaced by outside air. A high recirculation ratio stops CO_2 dilution and causes rapid build-up [21, 31].
Blower	`Blower Speed`, `Blower Current`, `Blower Power`	The blower drives air through the HVAC system. Higher speeds increase fresh-air delivery and CO_2 dilution when the fresh-air intake is open.
HVAC Thermal	`Evaporator Temp.`, `Heat Exchange Rate`, `Cabin Temp.`	Reflect the current heating or cooling mode and indicate whether the system is likely to restrict fresh-air admission to maintain the thermal set point.
Humidity	`Cabin Humidity`, `Air Humidity Temp.`	Occupant respiration raises both CO_2 and humidity simultaneously, providing a fast indirect signal of occupant load.
Air Distribution	Vent, floor, and defrost actuator positions	The active distribution mode determines where conditioned air is directed and how effectively fresh air displaces accumulated CO_2.
Vehicle	`Vehicle Speed`, `Driver Door Open Status`, `Passenger Door Open Status`	Higher speed increases air infiltration through body seals. Door-open events introduce a sudden fresh-air exchange that rapidly dilutes cabin CO_2 [1, 30].

Table 3.2: Feature groups and their physical relevance to cabin CO_2 dynamics.

3.4.2 Deep Learning Architectures

The recurrent and convolutional architectures evaluated in this study are described in detail in Chapter 2. This section specifies the configurations used. Altogether, seven models were evaluated: RNN, Stacked RNN, GRU and LSTM according to the formulations in Sections 2.2.1–2.2.4, and the TCN, TCN-LSTM, and TCN-GRU described in Sections 2.2.6–2.2.7.
All models were trained using the Adam optimizer [29] with a learning rate of 1×10^{-4} and weight decay of 1×10^{-3}. Adam was chosen for its adaptive learning rates and reliable convergence across different architectures. Mean absolute error (L1 loss) was used as the training objective rather than mean squared error, as it is less sensitive to outliers that occasionally survive the preprocessing stage described in Section 3.3, making it more robust during training without requiring a perfectly clean target signal.

3.4.2.1 Model Configurations

The configurations in Table 3.3 were selected following a preliminary architecture search in which hidden size, number of layers, and dropout rate were varied; the combination yielding the lowest validation MAE was retained for each model family. Dropout of 0.20 was applied to all multi-layer models, and all models were trained for up to 50 epochs with a batch size of 32.

Model	Hidden	Layers	Dropout	Batch	Optimizer	Epochs	Params
RNN	16	1	—	32	Adam	50	609
Stacked RNN	16	3	0.20	32	Adam	50	1,697
GRU	16	2	0.20	32	Adam	50	3,425
LSTM	16	1	0.20	32	Adam	50	2,385
TCN	16	3	0.20	32	Adam	50	5,281
TCN-LSTM	(24, 32)	3+2	0.20	32	Adam	50	16,849
TCN-GRU	(16, 32)	4	0.20	32	Adam	50	15,889

Table 3.3: Summary of model architectures and training hyper-parameters. For hybrid models, the layer count denotes TCN blocks followed by recurrent layers.

3.5 Approach II: Physics-Based Model

3.5.1 Physics-Based Modelling Framework

The second approach uses the cabin CO_2 mass balance model introduced in Section 2.3 as the physical basis for estimating cabin CO_2 concentration. Instead of directly predicting cabin CO_2 concentration from CAN bus signals, this approach first estimates the effective cabin body leakage flow rate $Q_l(t)$, which controls the balance between occupant CO_2 generation and CO_2 exchanged by outside air.

As shown in Section 2.3, the equilibrium cabin CO_2 concentration is determined by the occupant source term S, the ambient concentration C_{amb}, and cabin body leakage Q_l. For a given time-varying leakage estimate $Q_l(t)$, the corresponding moving equilibrium concentration is therefore

$$C_{\mathrm{eq}}(t) = C_{\mathrm{amb}} + \frac{S}{Q_l(t)}. \tag{3.4}$$

Here, $C_{\mathrm{eq}}(t)$ should not be interpreted as the instantaneous cabin CO_2 concentration.Instead it is the concentration that the cabin would asymptotically approach if the current HVAC conditions were to remain. Since real driving conditions are highly time-varying and the initial cabin concentration is uncertain, directly applying the analytical transient solution at every timestamp would be sensitive to error accumulation. Therefore, this study uses the equilibrium concentration as a more stable and interpretable physics-derived representation of the cabin CO_2 concentration state.

The key remaining task is to estimate $Q_l(t)$. Following the steady-state relationship derived in Section 2.3, Q_l can be inferred when the cabin CO_2 concentration is approximately at equilibrium. Therefore, the first step of the physics-based approach is to identify quasi-steady-state segments from the measured CO_2 time series and use them to obtain target values for leakage-flow modelling. This procedure is described in the next subsection.

3.5.2 Steady-State Inference of cabin body leakage

When the cabin reaches steady state, Equation (3.4) holds. Under this condition, the leakage flow rate $Q_l(t)$ can be determined from the cabin CO_2 concentration. Therefore, if steady-state points can be identified, the corresponding leakage flow rate Q_l can be inferred from the cabin CO_2 concentration measured at those points.

In practice, perfectly steady-state conditions are difficult to achieve in real driving data. Therefore, quasi-steady-state points were identified from the measured time series using rolling-window statistics of the cabin CO_2 signal. Only data segments with sufficiently small CO_2 variation and slope were retained so that the steady-state assumption remained approximately valid. The inferred Q_l values at these quasi-steady-state points were then used as target values for the subsequent modeling of the time-varying leakage flow rate.

The quasi-steady-state points were identified according to the following criteria:

- A rolling window of 120 samples was applied to the cabin CO_2 signal to characterize its local temporal behavior.
- Within each window, the rolling mean cabin CO_2 concentration $\bar{C}_{cab}$, the rolling standard deviation σ_{CO_2}, and the local CO_2 slope $\mathrm{d}C/\mathrm{d}t$ in ppm/min were calculated.
- A sample was considered quasi-steady-state only if the CO_2 variation within the window was sufficiently small:

$$\sigma_{CO_2} < 15 \text{ ppm}. \tag{3.5}$$

- In addition, the local rate of change of cabin CO_2 was required to be small:

$$\left|\frac{\mathrm{d}C}{\mathrm{d}t}\right| < 2\ \text{ppm/min}. \tag{3.6}$$

- To avoid unstable leakage estimates when the cabin CO_2 concentration was too close to the ambient level, only samples satisfying

$$\bar{C}_{cab} > C_{\text{amb}} + 50 \tag{3.7}$$

  were retained.

- Consecutive samples satisfying all of the above criteria were grouped into candidate quasi-steady-state segments. Only segments with sufficient duration were retained, and one representative midpoint from each segment was selected as a steady-state point for subsequent leakage inference. Furthermore, files containing too many steady-state points were down-sampled to prevent data imbalance across files. One file can only have at most 20 steady point to be chosen.

Finally we get 735 steady points in total, and then using Equation (2.23) to calculate Q_l of these steady points as target.

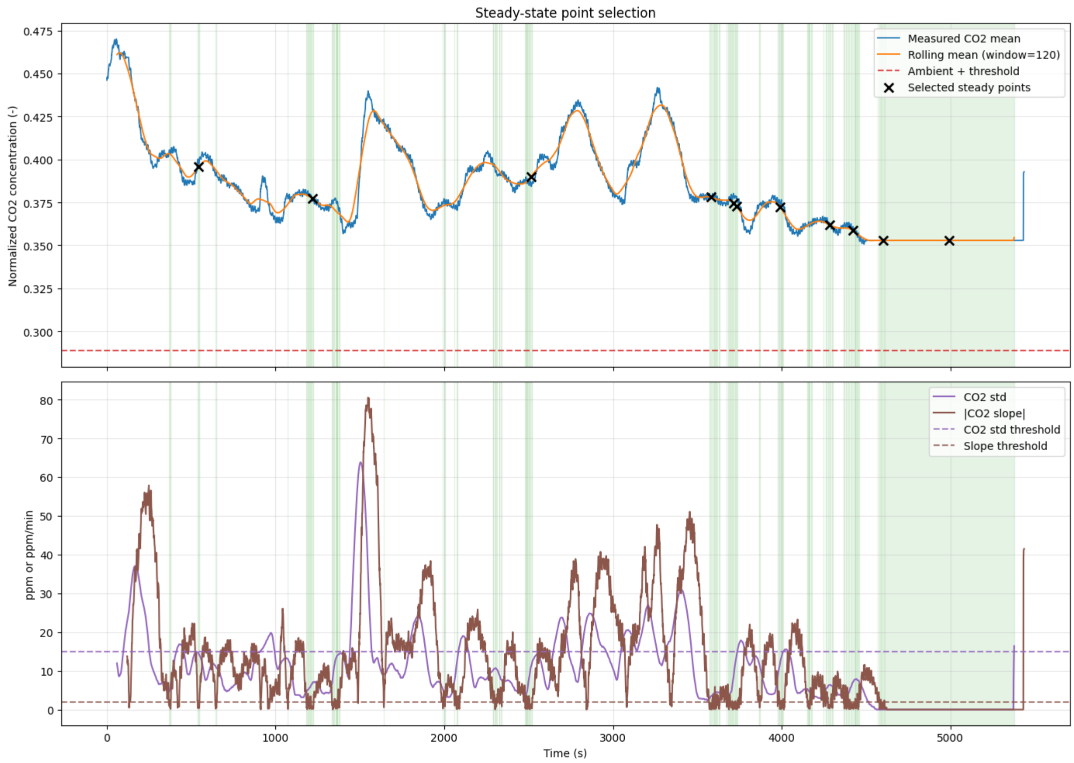

Figure 3.4: Steady-state point selection for cabin CO_2 analysis. Top: normalized measured cabin CO_2 concentration, 120-sample rolling mean, normalized ambient CO_2 threshold, and selected quasi-steady-state points, where 3000 ppm is scaled to 1. Bottom: rolling CO_2 standard deviation in ppm and absolute CO_2 slope in ppm/min, which are used to identify quasi-steady-state periods. The green regions indicate candidate time intervals where the CO_2 signal satisfies the steady-state selection criteria, and one representative point is selected from each interval that meets the minimum length requirement.

3.5.3 Data-Driven Estimation of Time-Varying cabin body leakage

Previous studies by Chang et al. have shown that cabin body leakage is affected by vehicle speed and ventilation fan speed.[30] In addition, recirculation ratio also affects the effective cabin body leakage, since a higher recirculation ratio reduces the amount of fresh outside air introduced into the cabin.[31] Therefore, we choose these three feature to fit cabin body leakage Q_l.

- **Vehicle speed:** Vehicle speed affects the pressure difference between the cabin and the surrounding environment, and therefore influences the air exchange rate through leakage paths. In general, a higher vehicle speed tends to increase cabin body leakage.
- **Blower speed:** Blower speed affects cabin body leakage by changing the pressure distribution inside the cabin ventilation system. When the blower operates at a higher speed, the air velocity near the fan inlet increases, which leads to a larger local pressure drop. This pressure difference enhances the

driving force for air exchange through leakage paths in the cabin body, thereby increasing cabin body leakage.

- **Recirculation ratio:** The recirculation ratio determines the proportion of cabin air that is recirculated within the HVAC system instead of being replaced by outside air. A higher recirculation ratio generally reduces the amount of fresh air entering the cabin, and thus affects cabin body leakage.

These three features were then used as input variables, while the calculated Q_l values at the identified steady points were used as target labels. Several machine learning methods were tested to model the relationship between the input features and Q_l. The steady-state dataset was randomly split, with 80% used for training and the remaining 20% used for testing. As shown in Table 3.4, the Gradient Boosting Decision Tree model achieved the best predictive performance among the evaluated methods. Figure 3.5 shows the difference between the model fitted Q_l and the calculated Q_l.

Table 3.5 show that blower speed is the most important variable for predicting Q_l, followed by recirculation ratio and vehicle speed. This indicates that blower operation has the strongest influence on the cabin body leakage, while recirculation also plays an important role. In comparison, vehicle speed has a smaller contribution in the current model.

After establishing the fitting model for the steady points, the trained GBDT model can be applied to estimate Q_l at each timestamp, since each file in the original dataset contains vehicle speed, blower speed, and recirculation ratio.

Model	Train		Test	
	MAE	RMSE	MAE	RMSE
Polynomial Ridge degree 2	49.58	70.72	45.66	60.27
Polynomial Ridge degree 3	44.88	66.41	41.16	56.24
Random Forest	30.12	48.97	28.18	40.25
Extra Trees	39.26	61.65	36.56	52.41
Gradient Boosting	**20.18**	**31.27**	**28.38**	**39.43**
Hist Gradient Boosting	20.72	33.37	28.83	39.52
SVR RBF	42.56	68.15	39.84	55.73

Table 3.4: Performance comparison of machine learning models for Q_l prediction

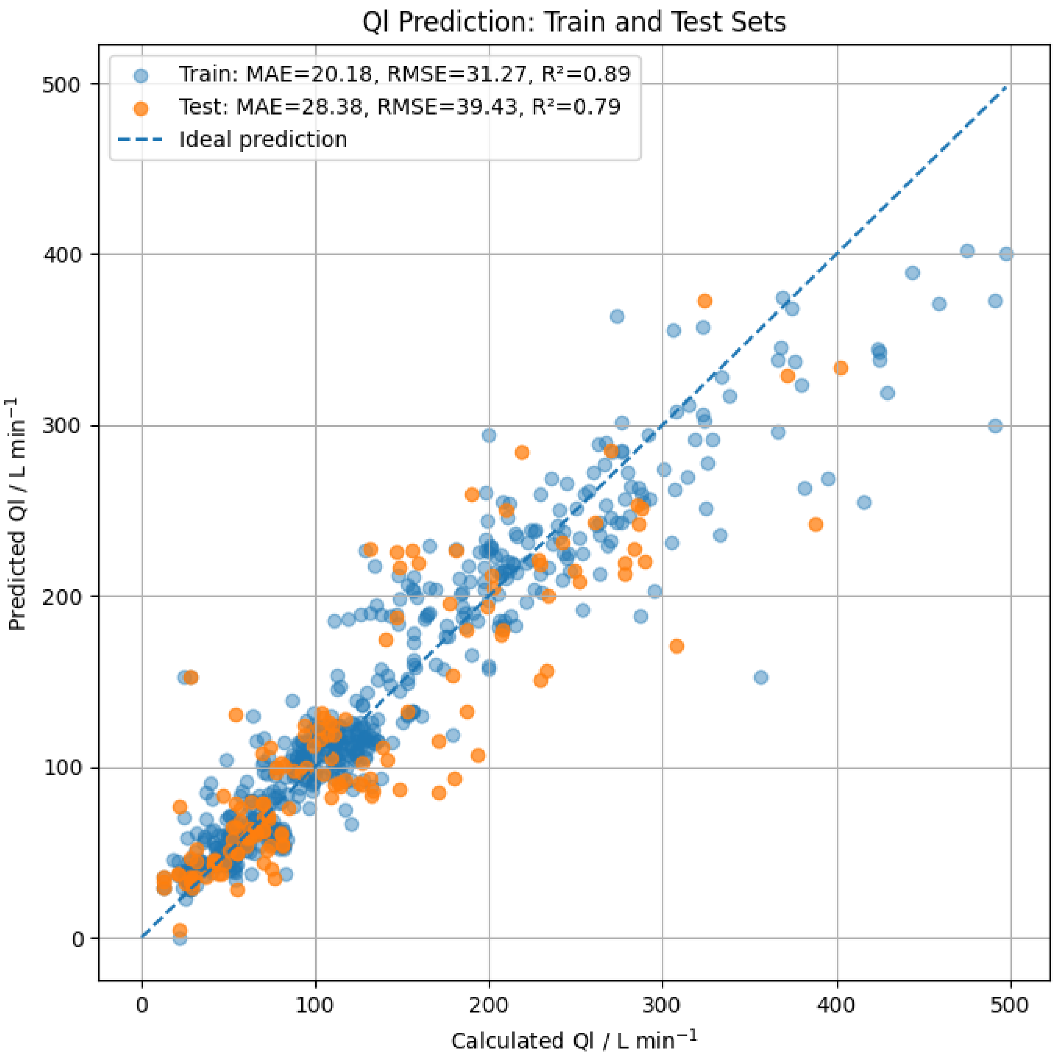

Figure 3.5: Q_l prediction of training set and test set from GBDT model

Feature	Importance
Blower speed	0.516694
Recirculation ratio	0.338140
Vehicle speed	0.145166

Table 3.5: Feature importance of the GBDT model for Q_l prediction

3.5.4 Interpretation of Equilibrium CO_2

From section 3.5.3, we already got a GBDT model to predict Q_l in every timestamp. For a given cabin body leakage $Q_l(t)$, the corresponding equilibrium cabin CO_2 concentration can be written as

$$C_{\mathrm{eq}}(t) = C_{\mathrm{amb}} + \frac{S}{Q_l(t)}. \tag{3.8}$$

Physically, $C_{\mathrm{eq}}(t)$ represents the asymptotic CO_2 concentration that would be reached if the current operating conditions remained unchanged for a sufficiently long period of time. In other words, it is not the instantaneous cabin CO_2 concentration itself, but the target level toward which the cabin CO_2 concentration tends to evolve under the current balance between occupant emissions and ventilation.
This interpretation can also be understood directly from Equation (2.17). After simplify this equation, we can get:

$$\frac{dC(t)}{dt} = \frac{Q(t)}{V}\left(C_{\mathrm{eq}}(t) - C(t)\right) \tag{3.9}$$

If the current cabin CO_2 concentration $C(t)$ is higher than $C_{\mathrm{eq}}(t)$, the ventilation effect dominates and the cabin CO_2 concentration tends to decrease. Conversely, if $C(t)$ is lower than $C_{\mathrm{eq}}(t)$, the internal CO_2 source dominates and the cabin CO_2 concentration tends to increase. Therefore, the difference between $C(t)$ and $C_{\mathrm{eq}}(t)$ determines the instantaneous direction of CO_2 evolution.
One may ask why the analytical solution of the mass balance equation(Equation (2.20)), was not directly used to calculate the cabin CO_2 concentration at different timestamps. The main reason is that this approach involves too many uncertain variables and is highly sensitive to error accumulation. In particular, the initial cabin CO_2 concentration is not accurately known, and the leakage flow rate Q_l varies over time rather than remaining constant. As a result, it is difficult to determine the instantaneous cabin CO_2 concentration reliably from this equation alone. Therefore, in this study, the equilibrium CO_2 concentration was used instead, as it provides a more stable and interpretable representation of the cabin ventilation state.
In practice, since $Q_l(t)$ varies with vehicle operating conditions, the equilibrium concentration is also time-dependent. As a result, $C_{\mathrm{eq}}(t)$ should be interpreted as a moving equilibrium level rather than a fixed steady-state value. Cabin CO_2 concentration does not immediately coincide with $C_{\mathrm{eq}}(t)$, but instead approaches it gradually according to the first-order dynamic response described by the mass balance equation.

For this reason, $C_{eq}(t)$ can be used as a physically interpretable diagnostic variable. It helps explain whether the model predicts an increasing or decreasing CO_2 trend at a given time and provides additional insight into the interaction between occupant emissions and effective ventilation.

Method	MAE	RMSE	MAPE
Raw equilibrium CO_2	167.22	260.28	17.80 %
Filtered equilibrium CO_2	**158.11**	**249.01**	**16.94** %

Table 3.6: Overall performance of equilibrium CO_2 against measured CO_2

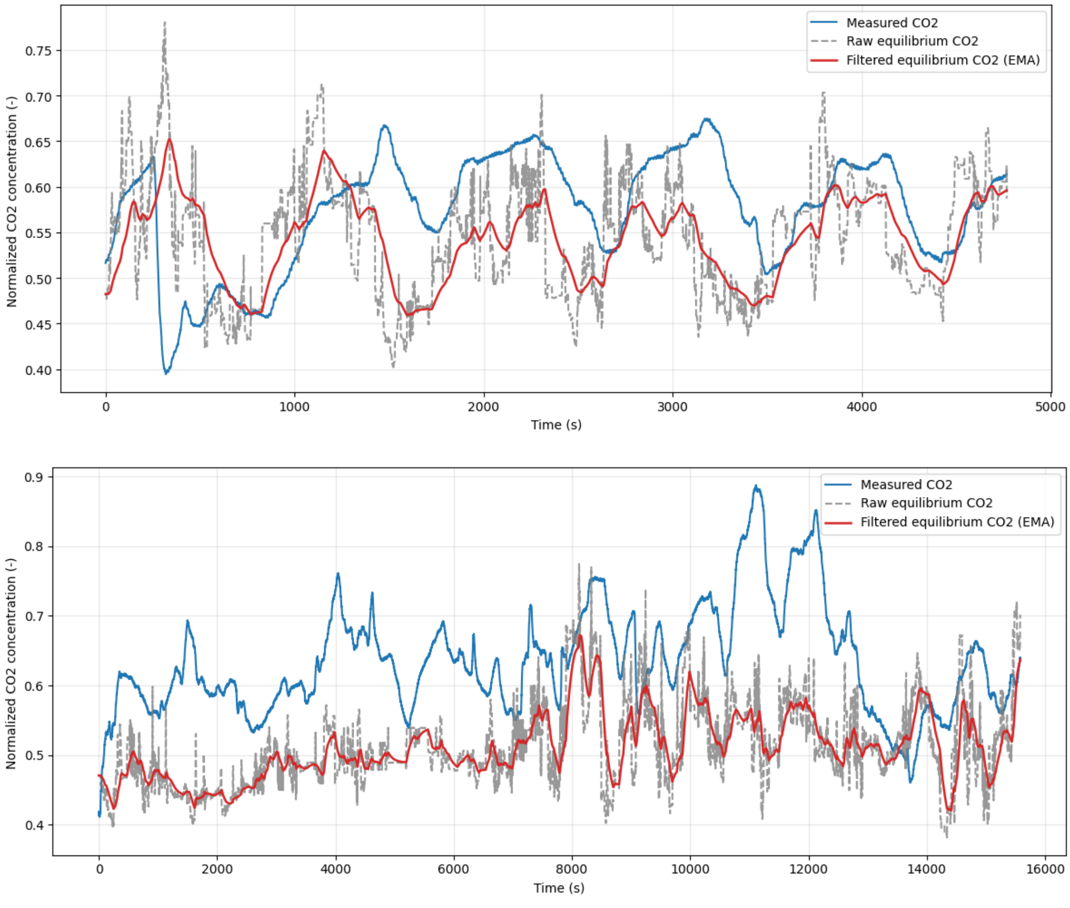

Figure 3.6: Comparison of measured and equilibrium CO_2 concentration for two selected files. Each panel shows the measured cabin CO_2 concentration (blue solid line), the raw equilibrium CO_2 concentration (grey dashed line), and the EMA-filtered equilibrium CO_2 concentration (red solid line).

The equilibrium CO_2 concentration was calculated at each time step in the dataset and compared with the measured cabin CO_2 concentration, as summarized in Table 3.6. The raw equilibrium CO_2 estimate yielded a MAPE of 17.80%. To reduce short-term fluctuations, an exponential moving average (EMA) was applied to the raw equilibrium CO_2 sequence separately for each file. A span of 180 samples was used, corresponding to a smoothing factor of $\alpha = 2/(180 + 1) \approx 0.01105$. The EMA-filtered equilibrium CO_2 concentration was computed recursively as

$$C_t^{\mathrm{EMA}} = \alpha C_t^{\mathrm{eq}} + (1 - \alpha) C_{t-1}^{\mathrm{EMA}},$$

where C_t^{eq} is the raw equilibrium CO_2 concentration and C_t^{EMA} is the filtered value at time step t. The first filtered value was initialized using the first raw equilibrium CO_2 value. After EMA filtering, the MAPE was reduced to 16.94%.
For visualization, two files were selected and plotted in Figure 3.6. Each panel shows the measured CO_2 concentration, the raw equilibrium CO_2 concentration, and the EMA-filtered equilibrium CO_2 concentration. It can be observed that the EMA-filtered equilibrium CO_2 curve follows a trend similar to that of the measured CO_2 concentration.

3.6 Approach III: Physics-Informed Data-Driven Model

The third approach combines the two previous methods by using the output of the physics-based model as an additional input feature to the data-driven model. The motivation is that the physics model encodes known cabin dynamics explicitly, providing the deep learning model with a physically grounded prior that may help it correct for errors in the simulation caused by uncertain ventilation rate estimates or unmodelled effects such as air infiltration. Rather than relying on the data-driven model to rediscover the mass balance from scratch, this approach provides the physics prediction as a starting point and lets the model learn what the physics signal alone cannot capture.

3.6.1 Combined Feature Set

The physics model output, filtered equilibrium CO_2 is appended as an additional feature column to the CAN bus feature matrix described in Section 3.4.1, giving a combined feature set of **20 features**. All other preprocessing, scaling, and sequencing steps described in Section 3.3 are applied identically. The combined input tensor is therefore:

$$\mathbf{X}^{\mathrm{comb}} \in \mathbb{R}^{N \times 120 \times 20} \tag{3.10}$$

3.6.2 Training and Evaluation

The combined model uses the same deep learning architectures and training configuration described in Section 3.4.2. The best architecture identified in the architecture search is retrained on the combined feature set. This allows a direct comparison between three configurations on the same held-out test set: the data-driven model trained on CAN bus features only, the physics-based model running without any learned parameters, and the combined model trained on CAN bus features augmented with the physics prediction. The comparison reveals whether the physics model output provides useful additional signal beyond what the CAN bus features alone convey to the deep learning model.

3.7 Post-Processing

Model predictions can exhibit short-term oscillations that are not physically meaningful, since cabin CO_2 changes slowly. A light smoothing step was therefore applied to the raw predictions at evaluation time using an exponentially weighted moving average (EWMA) with a span of 3 samples:

$$\hat{y}_t^{\text{smooth}} = \alpha \, \hat{y}_t + (1 - \alpha) \, \hat{y}_{t-1}^{\text{smooth}}, \qquad \alpha = \frac{2}{\text{span} + 1} \tag{3.11}$$

where $\hat{y}_t$ is the raw prediction at time t and $\alpha = 0.5$ is the smoothing factor. Smoothing was applied separately within each session and is included when computing all reported evaluation metrics.

3.8 Evaluation Criteria

To compare the performance of all three approaches using the predicted results and actual values, four complementary metrics were used. These metrics and their interpretations are summarised in Table 3.7.

Table 3.7: Evaluation metrics and their interpretation.

Metric	Formula	Interpretation
MAPE	$\frac{100}{N} \sum_{i=1}^{N} \left\lvert \frac{y_i - \hat{y}_i}{y_i} \right\rvert$	Scale-independent relative error (%). Primary ranking criterion.
MAE	$\frac{1}{N} \sum_{i=1}^{N} \lvert y_i - \hat{y}_i \rvert$	Mean absolute error in ppm. Directly interpretable in physical units.
RMSE	$\sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$	Penalises large deviations more heavily than MAE. Sensitive to outliers.

MAPE is expressed as a percentage, so a value closer to zero indicates higher prediction accuracy. This is particularly useful for comparing prediction values at different scales and is used as the primary ranking criterion, consistent with related work on virtual cabin sensors [19]. MAE provides the average absolute error in the same unit as the predicted values (ppm), helping to understand model errors in a physically interpretable way. RMSE is additionally used because it is more sensitive to large

deviations, penalising the high-error events that are most critical from a cabin air quality perspective.
In addition to aggregate metrics computed over the full test set, each model was evaluated separately within the three temperature conditions (cold, mild, warm) to expose condition-specific failure modes. This disaggregated view is important because performance requirements and HVAC operating regimes differ substantially across conditions.

3.9 Model Selection

After all three approaches were evaluated on the hold-out test set, models were ranked by ascending MAPE (primary criterion) and the best-ranked model was selected as the final virtual CO_2 sensor.

4
Results

4.1 Overview

This chapter presents the experimental results obtained by evaluating the three approaches described in Chapter 3 on the held-out test set. Section 4.2 reports the results of Approach I, the purely data-driven deep learning model, including the architecture search, overall performance, per-condition breakdown, and feature importance analysis. The results of Approach II, the physics-based model, were reported as part of the methodology in Chapter 3, as they constitute intermediate steps in the model pipeline rather than final predictions. Section 4.3 presents the results of Approach III, the physics-informed data-driven model, and compares it directly against both previous approaches.
All three approaches were evaluated on the same held-out test set using the metrics defined in Section 3.8: MAPE, MAE and RMSE. MAPE is used as the primary ranking criterion throughout.

4.2 Approach I: Data-Driven Model

4.2.1 Overall Performance

Table 4.1 reports the performance of all models on the held-out test set after full training, before the EWMA post-processing described in Section 3.7 is applied. Models are ranked by ascending MAPE.
Table 4.2 reports the same models after applying the EWMA smoothing filter with a span of 3 (Section 3.7). Smoothing consistently improves all metrics across all models and the ranking by MAPE is fully preserved, with Hybrid TCN-LSTM remaining the best model across all three metrics.

4.2.2 Per-Condition Breakdown

The results displayed in Table 4.3 provide a breakdown of the performance of the model that performed the best across all three temperature conditions.
The mild condition yields the lowest error across all three metrics, with a MAPE of 7.03% and an MAE of 61.63 ppm. This is expected since at moderate ambient temperatures, CO_2 dynamics are stable, with no extremes in heating demand or recirculation. The warm condition is the most challenging for the model, showing the highest MAPE (10.13%), MAE (129.72 ppm), and RMSE (163.28 ppm). The

Model	MAPE (%)	MAE (ppm)	RMSE (ppm)
Hybrid TCN-LSTM	**8.35**	**83.20**	**117.58**
Hybrid TCN-GRU	8.75	86.92	120.69
TCN	8.92	89.93	126.25
GRU	9.47	93.29	129.83
LSTM	9.47	94.22	133.18
Stacked RNN	10.32	101.54	141.07
RNN	11.39	109.93	162.01

Table 4.1: Hold-out test set performance of all data-driven models before post-processing smoothing, ranked by ascending MAPE. The best model per metric is shown in bold.

Model	MAPE (%)	MAE (ppm)	RMSE (ppm)
Hybrid TCN-LSTM	**7.89**	**77.95**	**108.88**
Hybrid TCN-GRU	8.39	82.58	112.69
TCN	8.46	84.45	117.23
GRU	8.89	87.02	119.54
LSTM	9.03	89.33	123.89
Stacked RNN	9.85	96.54	131.93
RNN	10.87	104.35	153.40

Table 4.2: Hold-out test set performance after EWMA smoothing (span = 3), ranked by ascending MAPE. Bold indicates the best value per metric.

Condition	MAPE (%)	MAE (ppm)	RMSE (ppm)
Cold ($T_{\mathrm{amb}} < 0\,^{\circ}\mathrm{C}$)	8.71	74.74	114.31
Mild ($0 \leq T_{\mathrm{amb}} \leq 20\,^{\circ}\mathrm{C}$)	7.03	61.63	82.04
Warm ($T_{\mathrm{amb}} > 20\,^{\circ}\mathrm{C}$)	10.13	129.72	163.28
Overall	**8.35**	**83.20**	**117.58**

Table 4.3: Per-condition performance of the Hybrid TCN-LSTM on the hold-out test set before post-processing smoothing.

mean CO_2 concentration in this regime is the highest in the dataset and the wider concentration range inflates the absolute errors. The cold condition falls between the two, with a MAPE of 8.71% and an RMSE of 114.31 ppm, indicating that cold-start transients and variable heating demand increase prediction difficulty relative to the mild regime, but less so than the sustained high-concentration periods seen in warm conditions.

4.2.3 Prediction Visualisation

Figure 4.1 compares the predictions of the best and worst performing models on the hold-out test set after smoothing. The top panel shows the Hybrid TCN-LSTM predictions against the measured cabin CO_2, the middle panel shows the RNN predictions, and the bottom panel overlays both models.

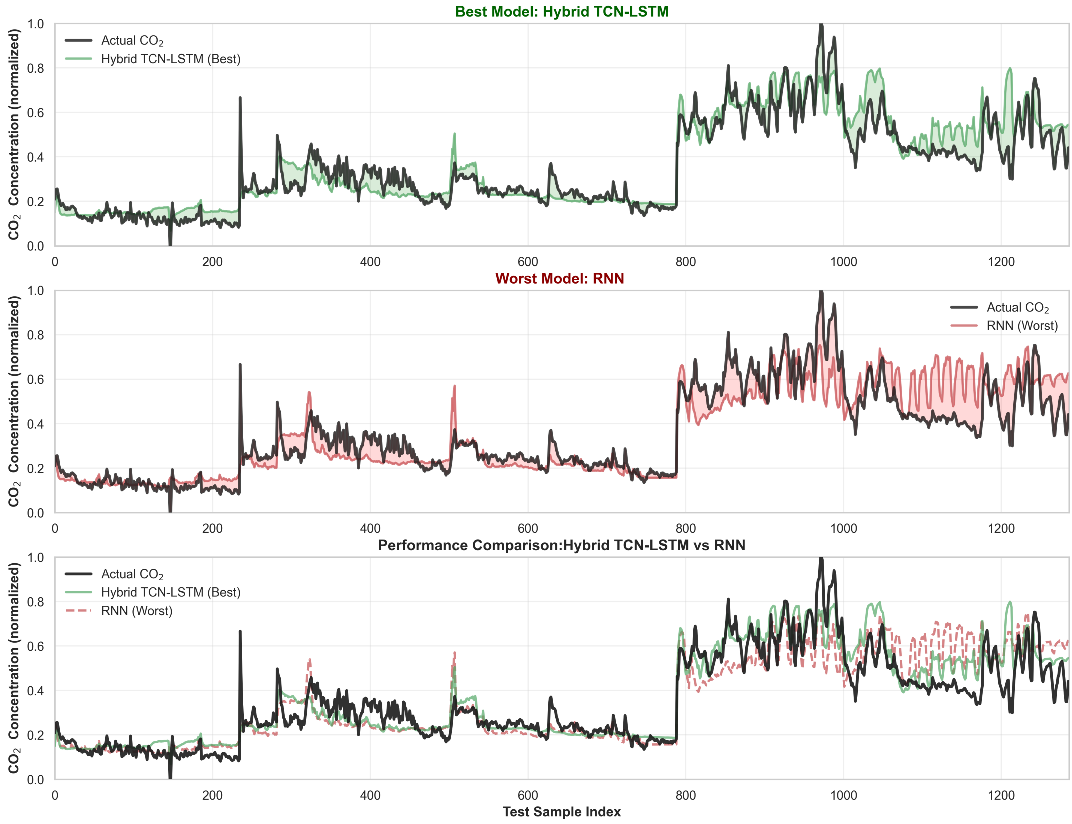

Figure 4.1: Prediction comparison on the hold-out test set after EWMA smoothing. *Top:* Hybrid TCN-LSTM (MAPE 7.89%, MAE 77.95 ppm). *Middle:* RNN (MAPE 10.87%, MAE 104.35 ppm). *Bottom:* Overlay of both models. Shaded areas indicate absolute prediction error.

Both models follow the general trend of the measured signal across the full test set, covering CO_2 levels from around 600 ppm in low-recirculation segments to over 1800 ppm in high-recirculation warm sessions. The Hybrid TCN-LSTM tracks the measured signal closely throughout, including sharp spikes and the sustained high-concentration period after sample 800. The shaded error region in the top panel is consistently narrow, indicating small and well-distributed errors. The RNN, by contrast, overshoots during the spikes around samples 250 and 500, and underestimates the measured signal in the high-concentration region after sample 800, where the wide shaded region reflects the largest errors of any model. The overlay in the bottom panel makes the difference between the two models clear.

4.3 Approach III: Physics-Informed Data-Driven Model

4.3.1 Overall Performance

Table 4.4 reports the performance of all models under Approach III on the held-out test set before smoothing. The equilibrium CO_2 signal $C_{\mathrm{eq}}(t)$ estimated by the physics-based model (Approach II) is appended as an additional input feature, giving a combined feature set of 20 features. All models use the same architecture and training configuration as in Approach I to enable a direct comparison.

Model	MAPE (%)	MAE (ppm)	RMSE (ppm)
Hybrid TCN-LSTM	**7.93**	**78.38**	**110.20**
Hybrid TCN-GRU	8.41	84.17	119.10
TCN	8.52	85.29	122.72
GRU	8.78	88.62	130.50
LSTM	8.83	89.11	130.56
RNN	9.05	90.57	130.70
RNN-Stacked	10.31	104.05	143.08

Table 4.4: Approach III hold-out test set performance (before smoothing), ranked by ascending MAPE. The physics model output $C_{\mathrm{eq}}(t)$ is included as an additional input feature. Bold indicates the best value per metric.

The EWMA smoothing described in Section 3.7 was then applied to all Approach III predictions. As with Approach I, smoothing improved every model across all three metrics and preserved the original ranking. The Hybrid TCN-LSTM remained the best model, with its MAPE dropping from 7.93% to 7.70% and its MAE from 78.38 to 75.75 ppm. The improvement of approximately 0.23 percentage points in MAPE is comparable to the gain seen in Approach I, confirming that the smoothing step is equally effective regardless of whether the physics signal is present in the input features.

Table 4.5 reports Approach III performance after EWMA smoothing (span = 3).

Model	MAPE (%)	MAE (ppm)	RMSE (ppm)
Hybrid TCN-LSTM	**7.70**	**75.75**	**105.69**
Hybrid TCN-GRU	8.11	80.78	113.46
TCN	8.16	81.30	116.26
GRU	8.44	84.81	124.87
LSTM	8.47	85.33	124.84
RNN	8.70	86.87	124.43
RNN-Stacked	9.96	100.49	137.79

Table 4.5: Approach III hold-out test set performance after EWMA smoothing, ranked by ascending MAPE.

4.3.2 Three-Way Comparison

Table 4.6 provides a unified comparison of the best model from each of the three approaches on the same held-out test set.

Approach	Model	MAPE (%)	MAE (ppm)	RMSE (ppm)
I	Hybrid TCN-LSTM	8.35	83.20	117.58
II	Filtered $C_{\text{eq}}(t)$	16.94	158.11	249.01
III	Hybrid TCN-LSTM (combined)	**7.93**	**78.38**	**110.20**

Table 4.6: Three-way performance comparison of the best model from each approach on the hold-out test set (before smoothing).

Adding $C_{\text{eq}}(t)$ as an input feature consistently improved the Hybrid TCN-LSTM across all three metrics compared to Approach I, reducing MAPE from 8.35% to 7.93%, MAE from 83.20 to 78.38 ppm, and RMSE from 117.58 to 110.20 ppm. This suggests that the physics-based equilibrium signal provided useful additional context that the model couldn't fully derive from the raw CAN bus features alone. Additional results, including per-condition breakdowns for all architectures and extended prediction visualizations, are provided in Appendix A.

5 Discussion

This study evaluated three approaches to predicting cabin CO_2 concentration in a heavy-duty truck from CAN bus signals: a purely data-driven deep learning model, a physics-based mass balance model, and a combined approach in which the physics model output is used as an additional input feature. The results show that all three approaches are feasible, with the combined approach producing the best overall performance.

Among the data-driven models evaluated in Approach I, the Hybrid TCN-LSTM achieved the lowest MAPE of 8.35% before smoothing and 7.89% after, making it the best-performing model. The consistent ranking across all conditions and both smoothed and unsmoothed evaluations suggests that this performance difference is not coincidental but reflects genuine architectural advantages. These can be attributed to two main factors.

Complementary feature extraction. The TCN encoder uses dilated causal convolutions applied in parallel to the full 120 second input window, allowing it to capture local temporal patterns at different time scales simultaneously. The LSTM decoder integrates these features sequentially, enabling it to keep track of the evolution of the ventilation state over time. Neither component alone achieves the same result: the plain TCN is ranked third, and the LSTM alone is ranked fourth, while their combination is ranked first. This is in line with the findings of Bai et al. [32], who showed that convolutional architectures can match or outperfom recurrent ones on sequence modelling tasks.

Gradient stability and long-range dependencies. Among the recurrent models, GRU and LSTM both clearly outperform the simple and stacked RNN. The gating mechanisms described in Sections 2.2.3 and 2.2.4 allow these models to retain relevant context over the full 120-step input window without the vanishing gradient problem that plagues the simple RNN. The stacked RNN improves on the simple RNN but still lags behind the gated architectures, confirming that depth alone does not solve the gradient problem without gating.

Per-condition performance. The model performs best in the mild condition (MAPE 7.03%) where CO_2 dynamics are primarily driven by recirculation without extreme HVAC demand. The warm condition is the hardest (MAPE 10.13%), which is expected given that it has the highest mean concentration (1,228 ppm) and the broadest distribution, as shown in Figure 3.3. The cold condition falls in between (MAPE 8.71%), where cold-start transients and variable heating demand add difficulty. Importantly, the architecture ranking is stable across all three conditions, which means the advantage of the Hybrid TCN-LSTM is general rather than specific to one temperature regime.

The physics-based model (Approach II) achieved a stand-alone MAPE of 16.94%, which is roughly double the worst data-driven model. This level of error is too high for direct use as a virtual sensor, but this is not unexpected. The model assumes a fixed per-occupant CO_2 emission rate and a perfectly mixed cabin, and the leakage flow Q_l is estimated from only three variables: blower speed, recirculation ratio, and vehicle speed. The GBDT feature importance results show that blower speed is the most influential predictor, accounting for 51.7% of the predictive weight, followed by recirculation ratio at 34% and vehicle speed at 15%. This confirms that the model is primarily driven by mechanical ventilation and has limited sensitivity to the infiltration effects captured by vehicle speed.

Despite these limitations, the equilibrium signal $C_{\text{eq}}(t)$ tracks the general trend of the measured concentration. The 735 quasi-steady-state points used to infer Q_l represent segments where the cabin state is stable enough for the mass balance assumption to hold approximately, and the GBDT model generalizes this relationship to the full driving record. The result is a signal that encodes the cumulative effect of past ventilation decisions on the current equilibrium level, which is not directly available from any single CAN bus measurement.

Approach III adds the filtered $C_{\text{eq}}(t)$ signal as a 20th input feature to the data-driven models. The improvement over Approach I is modest but consistent: roughly 0.4 percentage points in MAPE across all seven architectures and both smoothed and unsmoothed evaluations. The best model, Hybrid TCN-LSTM, improves from 7.89% to 7.70% MAPE after smoothing.

The consistency of the improvement across all models and all temperature conditions is notable. If $C_{\text{eq}}(t)$ were simply redundant with the existing CAN bus features, the improvement would be expected to be negligible or model-specific. The fact that every architecture benefits suggests the physics signal is providing genuinely complementary information, most likely the integrated ventilation history that determines the current equilibrium level. The 19 CAN bus features describe the instantaneous ventilation state well, but the equilibrium signal summarizes where that state has been heading over time, which is information the model otherwise has to infer indirectly from the raw sequence.

Deployment Considerations

A key motivation for this work is that the virtual sensor should be deployable on production vehicle hardware without requiring additional compute infrastructure. The results confirm that all architectures evaluated in this study are feasible in this regard. Every model produces predictions at well under 1 ms per sample, which is comfortably within the 1 Hz CAN bus sampling interval and leaves ample headroom for integration into an existing electronic control unit (ECU). The best-performing model, Hybrid TCN-LSTM, has 16,849 trainable parameters, which corresponds to approximately 66 kB of memory when stored in 32-bit floating point precision, well within the memory budget of modern automotive-grade microcontrollers. Inference times on an automotive-grade ECU will scale by a constant factor relative to the values reported in Table 5.1, but the relative ranking across architectures is expected to hold.

Model	**Params**	**ms / sample**	**Memory (kB)**
RNN	609	0.0073	2.4
Stacked RNN	1,697	0.0075	6.6
GRU	3,425	0.0047	13.4
LSTM	2,385	0.0042	9.3
TCN	5,281	0.0044	20.6
Hybrid TCN-LSTM	16,849	0.0046	65.8
Hybrid TCN-GRU	15,889	0.0050	62.1

Table 5.1: Inference time per sample and memory footprint for each architecture, measured on a 12th Gen Intel Core i7-12850HX CPU. Memory assumes 32-bit floating point parameter storage.

Among the architectures evaluated, the TCN stands out as the most attractive from a deployment perspective. It achieves a MAPE of 8.92%, only 0.57 percentage points behind the best model, with only 5,281 parameters and an inference time of 0.0044 ms per sample. This makes it a strong candidate for resource-constrained deployments, where memory and compute are limited. The Hybrid TCN-LSTM remains the preferred choice where accuracy is the primary concern, with the best MAPE across all conditions and a memory footprint of 65.8 kB, which remains within the budget of modern automotive-grade microcontrollers. The simple and stacked RNN architectures, despite having smaller parameter counts, exhibit marginally higher inference times. This is because their purely recurrent computation cannot leverage the parallelisation available to convolutional and hybrid architectures. In the latter case, the TCN encoder processes the full input window in parallel before passing its output to the recurrent decoder.

Limitations

This study used using data from a single truck platform over approximately 123 hours of driving. The model was not tested for generalisation to different cabin geometries, HVAC configurations or truck platforms, and future work should assess performance across multiple vehicle types to determine robustness. Most driving sessions involved only the driver, which is typical of heavy duty truck operation, but means the model has not been tested with the higher CO_2 generation rates that occur when multiple passengers are present..

The physics model uses a fixed per-occupant emission rate taken from the literature [21]. In reality, this rate varies with breathing intensity and activity level. A model that estimates the emission rate online, for example from the CO_2 rise rate during transient events, could improve both the stand-alone physics model and the benefit it provides in Approach III. Similarly, the GBDT leakage model uses only three features and relies on 735 quasi-steady-state points, which represent a small fraction of the full dataset. Expanding the feature set or using a more data-efficient leakage

estimation method could improve the quality of $C_{\mathrm{eq}}(t)$.
The prediction horizon of 30 seconds was selected based on empirical evaluation but may not suit all applications. For predictive ventilation control, a longer horizon may be needed. The EWMA smoothing improves MAPE by approximately 0.4 percentage points but introduces a small temporal lag that would need to be considered in real-time deployments.

6
Conclusion

This thesis developed and evaluated a virtual CO_2 sensor for heavy-duty truck cabins using signals available on the vehicle CAN bus. Three approaches were investigated: a purely data-driven deep learning model, a physics-based mass balance model, and a combined approach in which the physics model output is used as an additional input feature.

The results show that CAN bus signals contain sufficient information to predict cabin CO_2 concentration with a MAPE below 9% across all evaluated architectures, and below 8% for the best model after smoothing. The Hybrid TCN-LSTM consistently outperformed all other architectures across temperature conditions and evaluation metrics. After smoothing, it achieved an MAPE of 7.89% and an MAE of 77.95 ppm in Approach I, and an MAPE of 7.70% and an MAE of 75.75 ppm in Approach III. The consistent outperformance of hybrid convolutional-recurrent architectures over pure convolutional and pure recurrent models demonstrates that cabin CO_2 dynamics exist on multiple timescales and that capturing both local patterns and sequential dependencies are important for accurate prediction. The proposed virtual sensor shows a promising accuracy relative to the physical CO_2 sensors currently used in vehicles. The accuracy of existing automotive physical CO_2 sensors typically ranges from around 50 ppm to 200 ppm depending on the sensor type, operating range, and environmental condition [37][38], whereas our Hybrid TCN-LSTM achieved an MAE below 80 ppm. However, this comparison should be interpreted carefully. The target CO_2 used in this study was measure by physical CO_2 sensors, not representing the actual CO_2 concentration. Therefore, the reported MAE reflects the deviation from the reference sensor measurement, not the absolute error relative to the actual CO_2 concentration. Thus, it cannot be said that the virtual sensor is more accurate than the physical sensor itself, but the results show that it can reproduce the reference sensor signal with a practically usable level of accuracy.

The physics-based model mentioned in Approach II alone was not accurate enough to serve as a virtual sensor, with a stand-alone MAPE of 16.94%. However, the equilibrium CO_2 signal it produces encodes ventilation history in a form that the data-driven model cannot fully derive from the raw CAN bus signals. Approach III improved performance for all seven architectures and all temperature conditions when adding this signal as an additional input feature, showing that even a simplified physics model can bring useful complementary information in a hybrid approach.

The three research questions posed in Section 1.2 can now be answered directly. First, cabin CO_2 concentration can be accurately predicted from CAN bus signals alone, with the Hybrid TCN-LSTM performing best. Second, the physics-based mass balance model is not accurate enough on its own under real driving conditions, primarily due to the assumption of a fixed occupant emission rate and a limited leakage flow model. Third, incorporating the physics model output as an input feature does improve prediction performance, with a consistent improvement of approximately 0.4 percentage points in MAPE across all models.

Future Work

Future work could explore several directions.
First, the dataset needs to be expanded to include more cabin and ventilation scenarios such as window open and close, sunroof open and close, and various occupant numbers. These conditions significantly affect cabin air exchange rates and carbon dioxide production, but current datasets do not adequately reflect this.
Second, occupant CO_2 emission could be estimated online, rather than using a fixed value. In real-world driving, emissions can vary with occupant physical activity and metabolic status. For example, if the driver returns to the vehicle after exiting, their CO_2 emissions may be higher than when driving in a normal seated position. Therefore, future systems could combine vehicle signals with additional information, such as seat occupancy, occupant numbers, or wearable health data (e.g., heart rate or activity levels on a smartwatch), to more accurately estimate real-time CO_2 generation. This would allow ventilation systems to respond more intelligently to changes in occupant status, but privacy and data protection require careful consideration.
Third, testing the models on multiple truck platforms and under multi-occupant conditions would provide a clearer picture of generalisation. Investigating whether the computational cost of the hybrid architectures can be reduced through model compression would be important for deployment on embedded ECU hardware. Finally, extending the prediction horizon beyond 30 seconds could make the virtual sensor more suitable for predictive ventilation control applications.

Bibliography

[1] Hudda, N. and Fruin, S. (2018) “Carbon dioxide accumulation inside vehicles: The effect of ventilation and driving conditions.” In: *Science of the Total Environment*, 610–611, pp. 1448–1456. DOI: https://doi.org/10.1016/j.scitotenv.2017.08.105.

[2] Zhou, L., He, Y., Zhang, Q. and Zhang, L. (2021) “Carbon dioxide sensor module based on NDIR technology.” In: *Micromachines*, 12(7), p. 845. DOI: https://doi.org/10.3390/mi12070845.

[3] Lee, P. S. and Zhu, J. (2018) “In-cabin air quality during driving and engine idling in air-conditioned private vehicles in Hong Kong.” In: *International Journal of Environmental Research and Public Health*, 15(4), p. 611. DOI: https://doi.org/10.3390/ijerph15040611.

[4] Han, J., Lin, H. and Qin, Z. (2023) “Prediction and comparison of in-vehicle CO_2 concentration based on ARIMA and LSTM models.” In: *Applied Sciences*, 13(19), p. 10858. DOI: https://doi.org/10.3390/app131910858.

[5] Matthaios, V. N., Harrison, R. M., Koutrakis, P. and Bloss, W. J. (2023) “In-vehicle exposure to NO_2 and $PM_{2.5}$: A comprehensive assessment of controlling parameters and reduction strategies.” In: *Science of the Total Environment*, 891, p. 165537. DOI: https://doi.org/10.1016/j.scitotenv.2023.165537.

[6] Goh, C. C., Kamarudin, L. M., Shukri, S., Abdullah, N. and Zakaria, A. (2016) “Monitoring of carbon dioxide (CO_2) accumulation in vehicle cabin.” In: *Proceedings of the 2016 3rd International Conference on Electronic Design (ICED)*, pp. 427–432. DOI: https://doi.org/10.1109/ICED.2016.7804682.

[7] Gabriel, M. and Auer, T. (2023) “LSTM deep learning models for virtual sensing of indoor air pollutants: A feasible alternative to physical sensors.” In: *Buildings*, 13(7), p. 1684. DOI: https://doi.org/10.3390/buildings13071684.

[8] Esteban Bueno, S. (2021) “Vehicle HVAC system modeling and controlling.” M.A. thesis, Escola Tècnica Superior d’Enginyeria de Telecomunicació de Barcelona, Barcelona, Spain.

[9] Hallac, D., Bhooshan, S., Chen, M., Abida, K., Sosič, R. and Leskovec, J. (2018) “Drive2Vec: Multiscale state-space embedding of vehicular sensor data.”

In: *Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3233–3238. DOI: https://doi.org/10.1109/ITSC.2018.8569550.

[10] Volvo Trucks (n.d.) "Volvo FM – Den mångsidiga specialisten." *Volvo Selected*. Available at: https://volvoselected.com/se/truckModels/volvoFM (Accessed: 01 Jun. 2026).

[11] Ludwig, B., Ehrhardt, J. and Niggemann, O. (2025) "Creating virtual sensors using neural networks." In: *Proceedings of the 2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8. DOI: https://doi.org/10.1109/etfa65518.2025.11205672.

[12] Zaidan, M. A., Motlagh, N. H., Fung, P. L., Lu, D., Timonen, H., Kuula, J., Niemi, J. V., Tarkoma, S., Petäjä, T., Kulmala, M. and Hussein, T. (2020) "Intelligent calibration and virtual sensing for integrated low-cost air quality sensors." In: *IEEE Sensors Journal*, 20(22), pp. 13638–13652. DOI: https://doi.org/10.1109/jsen.2020.3010316.

[13] Huber, J., Weber, C., Eberhardt, A. and Wöllenstein, J. (2016) "Photoacoustic CO_2 sensor for automotive applications." In: *Procedia Engineering*, 168, pp. 3–6.

[14] Lemort, V., Olivier, G. and De Pelsemaeker, G. (2023) *Thermal Energy Management in Vehicles*. John Wiley & Sons.

[15] Daly, S. (2006) *Automotive Air Conditioning and Climate Control Systems*. Elsevier.

[16] Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. MIT Press.

[17] Box, G. E. P. and Cox, D. R. (1964) "An analysis of transformations." In: *Journal of the Royal Statistical Society, Series B*, 26(2), pp. 211–252. DOI: https://doi.org/10.1111/j.2517-6161.1964.tb00553.x.

[18] Steininger, M., Kobs, K., Davidson, P., Krause, A. and Hotho, A. (2021) "Density-based weighting for imbalanced regression." In: *Machine Learning*, 110(8), pp. 2187–2211. DOI: https://doi.org/10.1007/s10994-021-06023-5.

[19] Lee, H., Na, W. and Park, S. (2026) "Development of virtual sensors for vehicle in-cabin temperature prediction using deep learning." In: *Applied Sciences*, 16(1), p. 300. DOI: https://doi.org/10.3390/app16010300.

[20] Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y. and Tu, X. M. (2014) "Log-transformation and its implications for data analysis." In: *Shanghai Archives of Psychiatry*, 26(2), pp. 105–109. DOI: https://doi.org/10.3969/j.issn.1002-0829.2014.02.009.

[21] Jung, H. (2013) "Modeling CO_2 concentrations in vehicle cabin." In: *SAE Technical Paper* 2013-01-1497. DOI: https://doi.org/10.4271/2013-01-1497.

[22] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) “Dropout: A simple way to prevent neural networks from overfitting.” In: *Journal of Machine Learning Research*, 15(1), pp. 1929–1958. URL: `https://jmlr.org/papers/v15/srivastava14a.html`.

[23] Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) “Learning representations by back-propagating errors.” In: *Nature*, 323(6088), pp. 533–536. DOI: `https://doi.org/10.1038/323533a0`.

[24] Pascanu, R., Mikolov, T. and Bengio, Y. (2013) “On the difficulty of training recurrent neural networks.” In: *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 1310–1318. URL: `https://proceedings.mlr.press/v28/pascanu13.html`.

[25] Graves, A., Mohamed, A. and Hinton, G. (2013) “Speech recognition with deep recurrent neural networks.” In: *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649. DOI: `https://doi.org/10.1109/ICASSP.2013.6638947`.

[26] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) “Learning phrase representations using RNN encoder–decoder for statistical machine translation.” In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. DOI: `https://doi.org/10.3115/v1/D14-1179`.

[27] Hochreiter, S. and Schmidhuber, J. (1997) “Long short-term memory.” In: *Neural Computation*, 9(8), pp. 1735–1780. DOI: `https://doi.org/10.1162/neco.1997.9.8.1735`.

[28] Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014) “Empirical evaluation of gated recurrent neural networks on sequence modeling.” In: *NIPS 2014 Workshop on Deep Learning*. URL: `https://arxiv.org/abs/1412.3555`.

[29] Kingma, D. P. and Ba, J. (2014) “Adam: A method for stochastic optimization.” In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. URL: `https://arxiv.org/abs/1412.6980`.

[30] Chang, T.-B., Sheu, J.-J. and Huang, J.-W. (2017) “Vehicle air leakage ventilation and its effect on cabin indoor air quality.” In: *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 231(6), pp. 1226–1234. DOI: `https://doi.org/10.1177/0954408916664549`.

[31] Grady, M. L., Jung, H., Kim, Y. C., Park, J. K. and Lee, B. C. (2013) “Vehicle cabin air quality with fractional air recirculation.” In: *SAE 2013 World Congress & Exhibition*, 201503. DOI: `https://doi.org/10.4271/2013-01-1494`.

[32] Bai, S., Kolter, J. Z. and Koltun, V. (2018) “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling.” In: *arXiv preprint arXiv:1803.01271*. URL: `https://arxiv.org/abs/1803.01271`.

[33] Lea, C., Flynn, M. D., Vidal, R., Reiter, A. and Hager, G. D. (2017) "Temporal convolutional networks for action segmentation and detection." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 156–165. DOI: `https://doi.org/10.1109/CVPR.2017.113`.

[34] Dudukcu, H. V., Taskiran, M., Taskiran, Z. G. C. and Yildirim, T. (2023) "Temporal convolutional networks with RNN approach for chaotic time series prediction." In: *Applied Soft Computing*, 133, Article 109945. DOI: `https://doi.org/10.1016/j.asoc.2022.109945`.

[35] Zhou, S., Song, C., Wang, T., Pan, X., Chang, W. and Yang, L. (2022) "A short-term hybrid TCN-GRU prediction model of bike-sharing demand based on travel characteristics mining." In: *Entropy*, 24(9), Article 1193. DOI: `https://doi.org/10.3390/e24091193`.

[36] Xiang, X., Li, X., Zhang, Y. and Hu, J. (2024) "A short-term forecasting method for photovoltaic power generation based on the TCN-ECANet-GRU hybrid model." In: *Scientific Reports*, 14(1), Article 6744. DOI: `https://doi.org/10.1038/s41598-024-56751-6`.

[37] Innovaer Technologies (n.d.) "AIS-8100 automotive air quality sensor for CO_2 and particulate matter measurement." Available at: `https://innovaertech.com/product/ais-8100/` (Accessed: 5 Jun. 2026).

[38] Amphenol Advanced Sensors (2023) "Internal CO_2 sensor for automotive applications." Available at: `https://www.amphenol-sensors.com/hubfs/AAS-920-721D-Telaire-Internal-Auto-CO2-Sensor-061423-web.pdf` (Accessed: 5 Jun. 2026).

A
Appendix

This appendix presents per-condition performance tables for all models evaluated in Approach I, together with representative prediction time series for each temperature condition comparing Approach I and Approach III using the best-performing architecture, Hybrid TCN-LSTM.

Approach I: Per-Condition Performance

Model	**Condition**	**MAPE (%)**	**MAE**	**RMSE**
Hybrid TCN-LSTM	Cold	8.71	74.74	114.31
	Mild	7.03	61.63	82.04
	Warm	10.13	129.72	163.28
Hybrid TCN-GRU	Cold	10.31	87.98	122.73
	Mild	6.40	55.35	73.58
	Warm	10.80	138.62	171.26
TCN	Cold	9.97	85.81	115.73
	Mild	6.50	57.23	79.28
	Warm	11.71	149.86	187.72
GRU	Cold	10.54	92.66	137.99
	Mild	8.21	70.35	92.86
	Warm	10.27	132.55	167.40
LSTM	Cold	11.82	105.79	155.90
	Mild	6.68	57.82	79.39
	Warm	11.29	141.24	170.10
Stacked RNN	Cold	11.03	100.12	153.48
	Mild	9.04	76.87	100.22
	Warm	11.59	144.68	178.88
RNN	Cold	12.84	112.84	164.22
	Mild	10.87	94.86	161.50
	Warm	10.48	131.70	160.13

Table A.1: Approach I per-condition performance of all models before smoothing. MAE and RMSE are reported in normalized units.

Approach I vs Approach III: Predictions

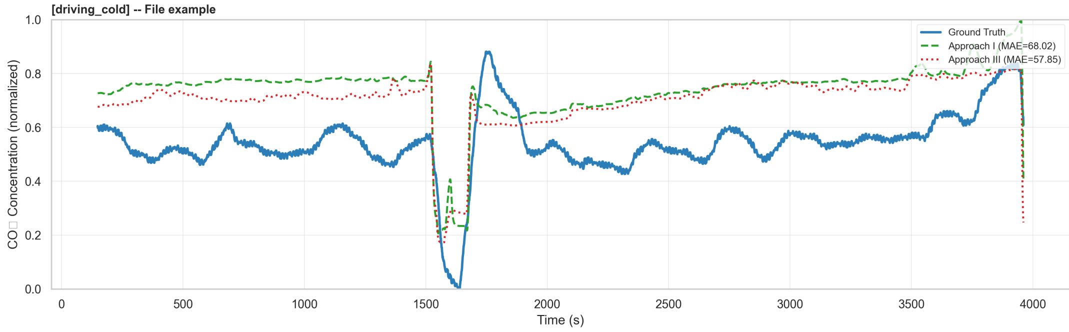

Figure A.1: Representative prediction time series for the cold condition, comparing Approach I and Approach III using the Hybrid TCN-LSTM. CO_2 concentration is shown on a normalized scale.

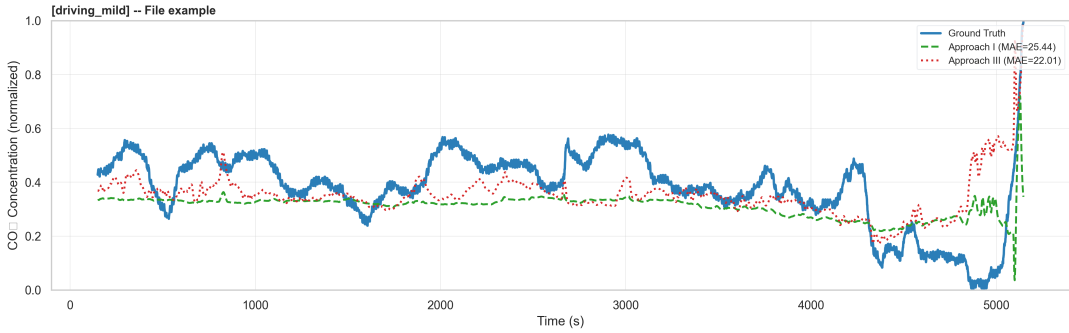

Figure A.2: Representative prediction time series for the mild condition, comparing Approach I and Approach III using the Hybrid TCN-LSTM. CO_2 concentration is shown on a normalized scale.

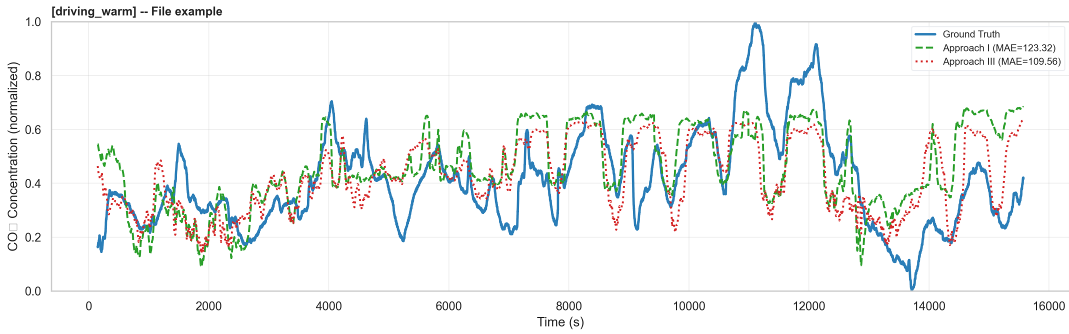

Figure A.3: Representative prediction time series for the warm condition, comparing Approach I and Approach III using the Hybrid TCN-LSTM. CO_2 concentration is shown on a normalized scale.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY