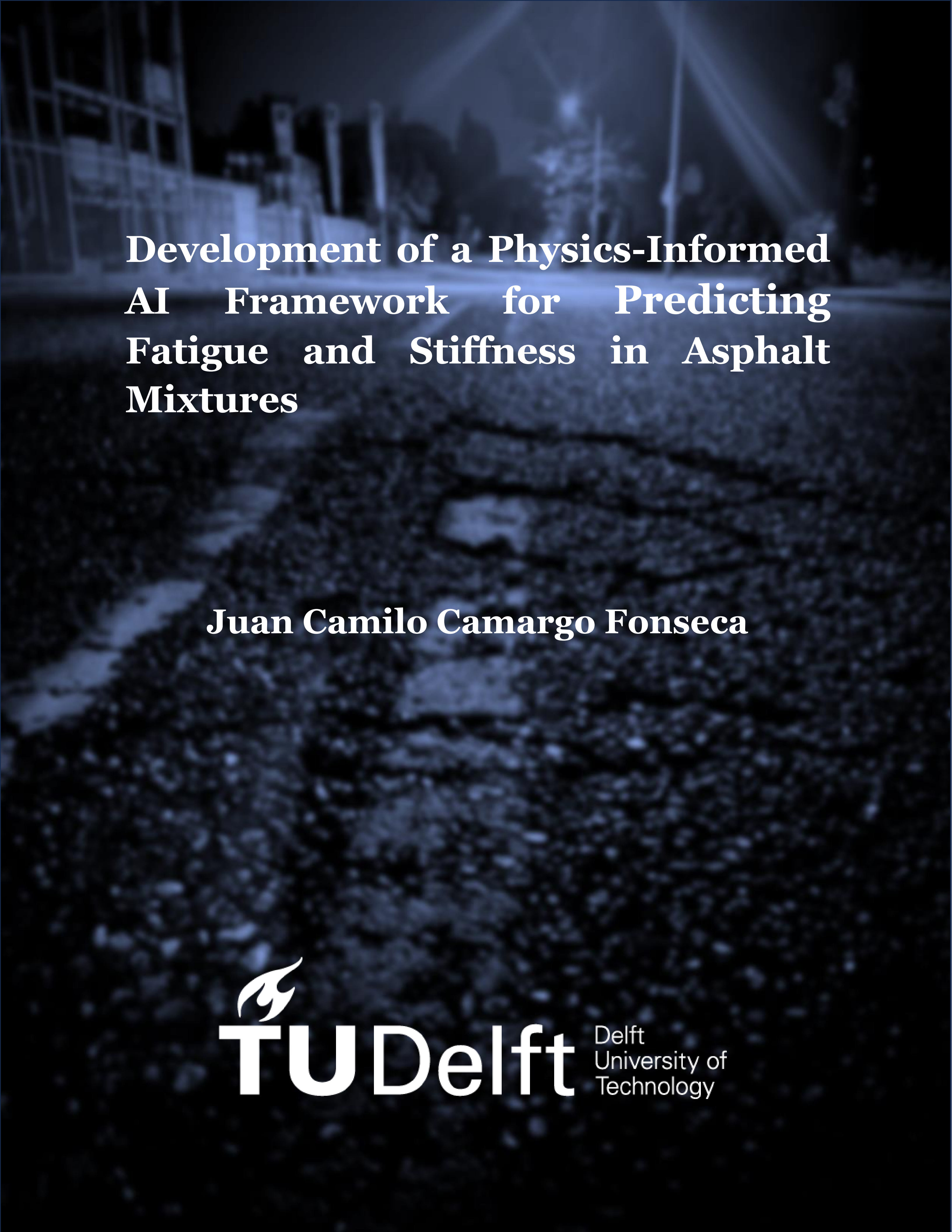


Development of a Physics-Informed AI Framework for Predicting Fatigue and Stiffness in Asphalt Mixtures

Juan Camilo Camargo Fonseca

TU Delft
Delft University of Technology

Development of a Physics-Informed AI Framework for Predicting Fatigue and Stiffness in Asphalt Mixtures

By

Juan Camilo Camargo Fonseca

Master thesis submitted to the Delft University of Technology in partial fulfilment of the requirements for the degree of

Master of Science in Civil Engineering – Traffic and Transportation Engineering

To be defended publicly on August 26th, 2024, at 14:30 (CET)

Graduation Committee

Graduation Chair	Dr. Kumar Anupam	TU Delft
Supervisor	Dr. Alfredo Núñez Vicencio	TU Delft
Advisor	Mohammadjavad Berangi MSc.	TU Delft

Acknowledgements

What once seemed like a faraway dream is now a reality. The completion of this Master Thesis marks the end of my journey at TU Delft, a journey that I feel immensely privileged and blessed to have had. I strongly believe that no one achieves anything alone. Therefore, I would like to express my gratitude to everyone who helped me reach this important milestone in my life.

I would like to start by addressing my graduation committee. Dr. Anupam, thank you for your valuable guidance, your insights, and for always pushing me to do better. Dr. Núñez, thank you for your constructive feedback and your valuable contributions to improve the outcome of this thesis. Mohammad, thank you for your commitment in every stage on the project and for your constant belief in my ability to overcome the challenges encountered. I am thankful to you all for entrusting me with this research project, which has significantly enhanced my coding, modelling, critical thinking, and writing skills.

I can't imagine my life in the Netherlands without the incredible friends I have made here. I want to thank you all for every conversation, birthday, dance, game, holiday, trip, bike ride, lunch, coffee break, night out, and meme we shared together. These ordinary but meaningful moments have been the most cherished part of my time at TU Delft, and I will always keep them with me. I would also like to thank my roommates for always making me feel at home, for the spontaneous dinner nights, and for the comforting conversations after a long day. I am deeply grateful that our paths crossed, and I will carry these meaningful memories with me as I move forward to the next stage in my life.

Por último, quiero expresar mi más profundo agradecimiento a mi familia. No hay palabras suficientes para expresar lo afortunado que me siento de tenerlos en mi vida. Gracias por estar siempre a mi lado, por su constante apoyo, por cada videollamada que me hizo sentir como en casa, por ponerme en sus oraciones y por hacerme sentir como si nunca me hubiera ido cada vez que fui de visita. Para mis papás tengo un párrafo aparte. Les quiero agradecer desde el fondo de mi corazón su amor, su entrega y sus sacrificios para que yo lograra mis metas y llegara lejos en la vida. Muchísimas gracias por las oportunidades que me dieron y por creer ciegamente en mí. Su ejemplo y convicción me han enseñado que soy capaz de lograr mucho más de lo que alguna vez imaginé. Este logro es tan mío como de ustedes, siempre llevaré con orgullo todo lo que me han enseñado.

Juan Camilo Camargo Fonseca
Delft, August 2024

Abstract

The Netherlands boasts an extensive road network that requires meticulous maintenance and preservation. Dutch asphalt pavements are assessed through functional properties such as mix stiffness and resistance to fatigue. However, current testing practice for these properties is intensive in time and resources, leading to the exploration of alternative methods for performance prediction. Recently, Artificial Intelligence (AI) has emerged as a tool for performance prediction in pavement engineering. Despite its potential, the application of AI is constrained by its limited interpretability and inconsistency with known physical laws. To enhance consistency and interpretability in AI predictive models, Physics Informed AI (PIAI) emerges as a promising approach.

This research develops a PIAI framework for physics infusion in pavement performance predictions. This infusion is accomplished through a Physics-Informed Loss Function balancing data and physics components in model training. The data component assures model predictions approximate the targets, whereas the physics component enforces a subset of features to follow a preset physical model. These components are also present during feature selection, where the physical model is used to guide the inclusion of important features in a PIAI model.

Using the developed framework, this research presents two PIAI prediction models based on the NL-LAB datasets. These models infuse homogenization theory and energy dissipation theory to enhance interpretability and consistency in stiffness and fatigue predictions. The results obtained on both models suggest that physics infusion is feasible without compromising prediction accuracy, balancing physical and statistical knowledge when predicting pavement performance. These findings also indicate that the PIAI framework is a promising approach for infusing physics into AI prediction models. Physics infusion can potentially enhance the acceptance and trust of AI within the pavement engineering community. Furthermore, the developed framework has the potential to accelerate pavement performance assessments by reducing the need for extensive material testing. Its flexibility also supports the incorporation of new physical models, fostering innovation and sustainability in pavement engineering.

Table of Contents

Glossary

4PBB	4-point Bending Beam
AI	Artificial Intelligence
CIFR	Common Important Feature Ratio
CNN	Convolutional Neural Network
DSR	Dynamic Shear Rheometer
GNN	Graph Neural Network
HMA	Hot Mix Asphalt
IRI	International Roughness Index
LTPP	Long-Term Pavement Performance
MAPE	Mean Average Prediction Error
MSE	Mean Square Error
NL-LAB	NederLands Langjarig Asfalt Bemonsteringsprogramma
ODE	Ordinary Differential Equation
OGM	Open Graded Mixes
PDE	Partial Differential Equation
PDI	Partial Differential Inequality
PI	Penetration Index
PIAI	Physics Informed Artificial Intelligence
PINN	Physics Informed Neural Network
PMB	Polymer Modified Bitumen
PV	Plateau Value
RAP	Recycled Asphalt Pavement
RDEC	Ratio of Dissipated Energy Change
ReLU	Rectified Linear Unit
RFE	Recursive Feature Elimination
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SHAP	Shapley Additive Explanation
TPE	Tree Parzen Estimator
XGB	XGBoost Gradient Boosting Library

1. Introduction

The opening chapter of this thesis introduces the research topic and delineates the structure of the report. It begins by presenting the motivation for investigating Physics Informed AI (PIAI) in pavement engineering, followed by a detailed problem statement. The chapter then defines the research objectives and questions and concludes with an overview of the research methodology and an outline of the subsequent chapters.

1.1. Research Motivation

A well-developed road network fosters economic development by facilitating passenger and freight transportation [1]. Hence, investing in road construction and asset management is a fundamental policy decision for governments worldwide [2]. The Netherlands has one of the densest road networks in the European Union [3]. A road network this extensive is a valuable asset that must be meticulously maintained, preserved, and improved [4]. This is reflected in the significant budgetary expenditures undertaken in road construction and maintenance by the Dutch government [5].

In the Netherlands, asphalt pavements are assessed through functional properties related to field performance [6]. Understanding this relationship is crucial for pavement engineering researchers and practitioners. For this purpose, an extensive research program named NederLands Langjarig Asfalt Bemonsteringsprogramma (NL-LAB) [6] was initiated. The NL-LAB program aimed to better understand the relationships between asphalt mix composition, construction, and field performance in the Netherlands [6].

A primary concern of asphalt field performance is bearing capacity [6], as traffic loads must be adequately dissipated through the pavement structure. Accordingly, the NL-LAB datasets contain two functional properties related to bearing capacity: mix stiffness and fatigue resistance [6]. These two properties are measured through the 4-point bending beam (4PBB) test [6]. The 4PBB test applies a periodic bending with constant strain to an asphalt mix beam, and records the resulting stress, strain, and phase angle per cycle [7], [8]. Mix stiffness is defined by the stress-strain ratio at the 100th load cycle [7], whereas fatigue resistance is measured via the initial strain corresponding to a fatigue life of $1x10^6$ cycles (ε_6) [8].

Although the 4PBB is a routine test for assessing mix stiffness and fatigue resistance, its execution can be complicated and time-consuming [9]. Pavement performance prediction models are an alternative approach to predicting fatigue resistance and stiffness. Prediction models in pavement engineering have traditionally followed three approaches, each with benefits and limitations [10]. Empirical models base on experimentation results and relate statistically pavement performance and explanatory variables, but often lack strong physical foundations [10]. Mechanistic models leverage a strong physical or mechanical foundation to model pavement functional properties, but their application is restricted by the complex nature of pavement engineering problems [10]. Mechanistic-Empirical models leverage empirical relationships between pavement performance and physics-based calculated pavement responses, infusing physics and statistics in a single modelling approach [10]. However, Mechanistic-Empirical models incorporate local calibration factors that require extensive fine-tuning procedures [11].

In current days, Artificial Intelligence (AI) is seen as an alternative tool in pavement engineering that can potentially bridge some limitations of traditional modelling approaches. AI applications in pavement engineering encompass distress detection, distress quantification, performance prediction, and maintenance programming and scheduling [12]. Although AI applications can potentially bring benefits, AI usage also entails a set of limitations [13]. Purely data-driven AI models are limited to the scope of the training data used and have limited generalizability [13]. Likewise, predictions made by data-driven models can be physically inconsistent, often earning the label of 'black-box' models [13].

This 'black-box' label relates to the challenges regarding interpretability of purely data-driven AI models [13]. These challenges can be potentially addressed via physics infusion in AI models [13]. This modelling approach is often known as Physics Informed AI (PIAI) and aims to develop physically consistent AI models [14]. It is noted that PIAI models have been developed for different fields in scientific computing such as fluid mechanics, solid mechanics, and material science [13]. However, limited applications in the field of pavement engineering are found in the past literature ([15], [16], [17], [18]). And, to the best of the researcher's knowledge, a framework to generate PIAI models for different pavement functional properties is underdeveloped.

1.2. Problem Statement

Research in pavement engineering is ongoing since the 18th century, with significant advancements and an increased research output emerging in the 1950s [19]. Consequently, a myriad of physical and empirical relationships — commonly referred as "physical models" — have been popularly employed in pavement engineering ever since. PIAI is a promising approach towards infusing physical models and AI capabilities for enhancing predictions. However, it remains uncertain whether such physics infusion is feasible in AI models predicting pavement performance based on the NL-LAB datasets. It is also unknown if a physical model should meet a set of conditions to facilitate physics infusion. Moreover, it is desired to understand the impact of this potential physics infusion on the accuracy, interpretability, and consistency of model predictions.

1.3. Research Objective and Questions

The main objective of this research is to design and validate a novel PIAI framework for predicting pavement functional properties based on the NL-LAB stiffness and fatigue datasets. Accordingly, the corresponding main research question reads as follows.

> Can physics be infused to enhance AI prediction models for stiffness and fatigue of asphalt mixtures?

The development of the PIAI framework requires the selection of an adequate physics infusion method. This method should be suitable for the characteristics of the datasets and the physical model to infuse. Hence, the first research sub-question reads as follows.

> Which method for infusing physics in AI models better suits the characteristics of the NL-LAB stiffness and fatigue datasets?

This research acknowledges that the choice of a physical model for infusion into AI predictions is influenced by the characteristics of the NL-LAB dataset and the physics infusion method. To address this, clear selection criteria for physical models within the developed framework are defined, as outlined in the second research sub-question, which reads as follows.

> What conditions are required to infuse a physical model into a PIAI prediction model for the NL-LAB stiffness and fatigue datasets?

The PIAI models developed for stiffness and fatigue predictions are evaluated using a multi-criteria performance assessment considering accuracy, interpretability, and consistency. Accuracy is measured by the PIAI models' ability to closely predict the target values [20]. Interpretability assesses the extent to which the model predictions adhere to the constraints imposed by the physical model [21]. Furthermore, given that AI model initializations are often performed randomly [22], consistency examines how well the physical constraints are maintained across different initializations [15]. The effects of physics infusion on these performance criteria are investigated, as addressed by the third and final research sub-question, which reads as follows.

How does infusing physics impact model accuracy, interpretability, and consistency?

1.4. Research Methodology

Figure 1-1 shows the 4-stage methodology adopted to attain the research aim and answer the research questions. The first stage refers to data collection, exploration, and provenance. The data provenance review enables an understanding of the testing conditions under which the dataset instances were generated [23]. The second stage contains a review and selection of a method for physics infusion in AI prediction models. The third stage dives into the development of a PIAI framework for prediction models based on the NL-LAB fatigue and stiffness datasets. This stage elaborates on physical model selection, data preparation, feature selection analysis, and model training and optimization for physics infusion. The fourth and last stage assesses the obtained PIAI models accuracy, interpretability and consistency.

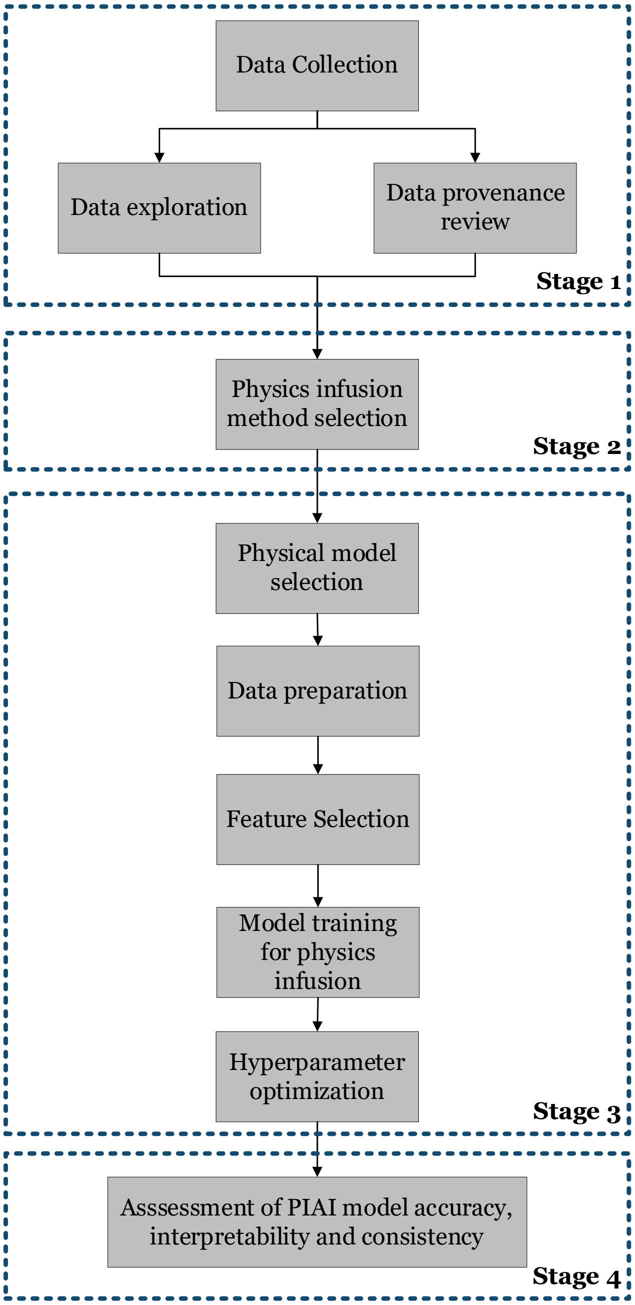

Data Collection
Data exploration
Data provenance review
Stage 1
Physics infusion method selection
Stage 2
Physical model selection
Data preparation
Feature Selection
Model training for physics infusion
Hyperparameter optimization
Stage 3
Asssessment of PIAI model accuracy, interpretability and consistency
Stage 4

Figure 1-1. Research methodology.

1.5. Report Outline

This report contains five chapters, with the remaining four chapters being structured as follows. Chapter 2 presents a literature review concerning the NL-LAB datasets and physics infusion method in AI models. Chapter 3 presents a conceptual introduction to the developed PIAI framework. Chapter 4 presents and discusses the results obtained for the developed PIAI fatigue and stiffness models. Finally, Chapter 5 presents the conclusions and future recommendations.

2. Literature Review

This research performed a literature review to identify a suitable method for infusing physics into AI models. This infusion method should fit the characteristics of the NL-LAB datasets and consider the properties of the physical model to infuse. Accordingly, this literature review is organized as follows. The review starts with Section 2.1, which focuses on possible AI physics infusion methods and PIAI applications developed in pavement engineering and related fields. Section 2.2 introduces the NL-LAB program and the stiffness and fatigue datasets. Then, Section 2.3 focuses on the testing procedures and physical models for predicting asphalt stiffness and fatigue resistance. Finally, Section 2.4 concludes the literature review with an introduction to the selected physics infusion method, answering the first research sub question.

2.1. Infusing physics into AI models

With PIAI, researchers strive to improve the performance of AI predictive models by leveraging prior domain knowledge in the form of a physical model [24]. Physics infusion is possible via a Physics-Informed Loss Function [14], a Physics-Informed Architecture [14], Physics-Informed Pre-Training [14], and hybrid methods [25]. The following subsection details physics infusion via a Physics-Informed Loss Function.

2.1.1. Physics-Informed Loss Function

In statistical learning, a model loss is a scalar value that quantifies the closeness between model predictions and targets [26]. This model loss can be conceptualized as a function of the model parameters (ϕ) [26], obtaining the loss function L[ϕ] [26]. Training an AI model means finding a set of model parameters that map each training input to its associated target as closely as possible [26]. When a model is trained, the loss function reached a local minimum with an associated set of optimal model parameters [26].

A Physics-Informed Loss Function incorporates a penalty term into the model loss [14]. Since this penalty term guides the model towards reaching physically consistent solutions [14], it can be interpreted as a physics learning bias [24]. However, a model with a penalty term will approximate but not guarantee the infuse physical model [24]. Therefore, Physics is weakly incorporated in the predictive model through a Physics Informed Loss-Function [24].

Equation 2-1 shows an example of a Physics-Informed Loss Function. The exemplified loss function has one data-driven component (*$Loss_D$*) and three physical components ($Loss_{P1}$, $Loss_{P2}$, and $Loss_{P3}$). All components have associated weight factors (λ_i), which can be interpreted as model hyperparameters that require tuning [27].

$$Loss = \lambda_1 \cdot Loss_D + \lambda_2 \cdot Loss_{P1} + \lambda_3 \cdot Loss_{P2} + \lambda_4 \cdot Loss_{P3} + ... \qquad \text{Equation 2-1}$$

where:

λ_i: Weight factor for component i.
$Loss_D$: Data-driven component of the loss function
$Loss_{P1}$: (Possible) first physical component of the loss function.
$Loss_{P2}$: (Possible) second physical component of the loss function.
$Loss_{P3}$: (Possible) third physical component of the loss function.

An application of a Physics-Informed Loss Function in pavement engineering was developed by Deng et al. [15] for rut depth prediction in asphalt mixtures. The author enforced rut depth predictions to follow a monotonically increasing trend with respect to increasing temperature and number of wheel passes in the Hamburg Wheel Tracking test [15]. The author infused the desired physical behaviour through a loss function with two physics-based components containing ReLU functions that penalized non-monotonic predictions [15]. The author demonstrated that this implementation improved model stability and rationality [15]. Noticeably, the author's research focused on tabular data without spatial or temporal relationships between instances. The NL-LAB datasets present similar characteristics (see Section 2.2).

Another application of a Physics-Informed Loss Function in pavement engineering was developed by Han et al. [18] for asphalt fatigue prediction. The author leveraged a dual pathway model architecture [18] to predict the damage characteristic (C-S) curve of asphalt mixtures [18]. The obtained model predicts the pseudo-stiffness (C) value with an average error of 5.2% for different temperatures and frequencies and infusing visco-elastic continuum dynamics [18]. Although the author reported that the obtained model had improved precision accuracy and generalizability, model interpretability is lacking [18]. Moreover, the model developed takes data in the form of time series, which differs from the characteristics of the NL-LAB datasets.

Physics-Informed Neural Networks (PINNs) are a sub category of Physics-Informed Loss Function [13]. Researchers mostly use PINNs when the physical model or relationship to infuse is in the form of partial differential equations (PDEs) and partial differential inequalities (PDIs) [25]. Considering the example of Equation 2-1, PINNs can incorporate physical components in the loss function relating to the residuals of physical equations, boundary conditions, and initial conditions [13]. PINNs can also include a data-driven component in the loss function when observations are available [13].

PINNs work by leveraging automatic differentiation, a computational method to differentiate the outputs of a neural network with respect to its inputs [13]. Through automatic differentiation, PINNs can solve forward and inverse problems involving PDEs [27]. In forward problems, PINNs can be used to compute solutions to PDEs [27]. Whereas, in an inverse problem, PINNs are used to discover unknown PDE parameters using observational data [27].

Despite various promising developments using PINNs, they entail several limitations. First, they can be affected by vanishing and exploding gradients, as with any deep learning task [26]. Also, neural networks tend to learn lower frequency solutions first and be biased towards smooth functions, which is known as spectral bias [28]. Notably, PINNs need to be retrained for every instance of a problem, for example, with a change in initial or boundary conditions.

To the best of the researcher's knowledge, PINN applications have been developed in engineering fields related to pavement engineering. In structural engineering, Kapoor et al. [29] investigated PINN applications for forward and inverse problems in complex beam systems. The author used three physics-driven components in the loss function for the forward problem, accounting for residuals of the governing PDE, boundary conditions, and initial conditions [29]. Moreover, a data-driven component was incorporated when solving the inverse problem [29]. The author demonstrated that PINNs can solve nondimensionalized Euler-Bernoulli and Timoshenko complex beam systems [29]. Although the application developed showed promising results, the NL-LAB dataset does not contain spatial or temporal series. Therefore, the relevance of the findings towards predicting pavement performance is limited.

Another PINN application was developed by Kapoor et al. [30] for moving load problems in beams. The author found that PINNs can solve forward and inverse problems considering Euler-Bernoulli beam theory [30]. Additionally, the author found that modelling a point load as a Gaussian function instead of a Dirac delta function prevents instability when training vanilla PINNs [30]. This work provides interesting insights into PINN applications for structural engineering. However, the characteristics of the NL-LAB datasets are not compatible with problems involving temporal or spatial discretization.

By incorporating a physics-informed loss function, physics is approximated without changing the model architecture. However, it is possible to modify such architecture to meet physical constraints, as detailed in the next subsection.

2.1.2. Physics-Informed Architecture

Model architecture is a term is used to describe the arrangement of layers, connections, and components that condition data flow through an AI model [31]. Model architecture choice often depends on the characteristics of the task at hand [26]. Thus, it is possible to encode physical or empirical dependencies in the core architecture of a model [13] [14].

Conventional neural networks [32] propagate information only in the forward direction, from inputs to targets, lacking mechanisms to learn dependencies among targets [33]. Previous knowledge about target dependencies can be infused in neural networks as inductive biases [26]. For example, Convolutional Neural Networks (CNNs) are designed to handle data on regular grids, assuming nearby data in space is highly correlated [26]. Likewise, Recurrent Neural Networks (RNNs) are tailored to process sequential and/or dynamic data, assuming high correlation between two inputs close in time [32]. Similarly, Graph Neural Networks (GNN) are suited to represent data in irregular graphs.

Despite its efficiency, a physics-informed architecture method has several limitations [24]. Task with relatively simple and well-defined physics tend to perform better than tasks with more complex physics [24]. Besides, this method requires careful elaboration and implementation, especially regarding the appropriate selection of an inductive bias [24]. Finally, implementation scaling or extension to more complex tasks is challenging, as the underlying physics is not well understood or is hard to encode in the network [24].

Kapoor et al. [33] introduced a novel framework to enhance generalization in physics-informed models by combining PINNs and neural oscillators. A neural oscillator is a specific network type that involves building RNN architectures based on Ordinary Differential Equations (ODEs) [33]. The author demonstrated that their proposed framework enables an AI model to learn the long-time dynamics of solutions to the governing PDEs [33]. The findings are promising for enhancing generalization performance in Physics-Informed AI models. However, the results are based on numerical experiments and not on an existing dataset. Moreover, none of the PDEs solved by the author correlates to a performance indicator modelled in this research.

Another plausible method for infusing physics in AI models involves training a data-driven model with physics-compliant inputs and targets. The following subsection discusses this method and its applications to real-world data.

2.1.3. Physics-Informed Pre-Training

With Physics-Informed Pre-Training, a data-driven model is initialized using physically consistent inputs and targets [13]. In a later stage, the model can be fine-tuned with observational or real-world data [25]. Although the initialization procedure may speed up model convergence to consistent solutions [14], physics is weakly infused in the prediction model [24].

Despite its effectiveness, a Physics-Informed Pre-Training framework brings several limitations. Pre-training often requires a large amount of data which can be computationally expensive to collect and process [24]. Moreover, the generated models are purely data-driven and thus are prone to learn the trend and noise in the pre-training set rather than the underlying physics [13]. Consequently, model generalizability beyond the pre-training set is challenging, and there is no guarantee of satisfying initial and boundary conditions [13].

PIAI applications using Physics-Informed Pre-Training have been developed in pavement engineering. Kargah-Ostadi et al. [16] developed a Physics-Informed Pre-Training framework for predicting the International Roughness Index (IRI). The author pre-trained a neural network using theoretical solutions of the Quarter-Car model from LTPP [34] road profile measurements and vehicle suspension properties, speed and acceleration response [16]. Then, the author fine-tuned the model to predict the IRI measurements contained in the LTPP database [34]. The final model showed good accuracy, precision, and generalization potential for smooth road profiles. Yet, the author acknowledged that sparse training samples from rougher road profiles in the LTPP database [34] limit model generalizability [16]. The framework developed highlights the potential of Physics-Informed Pre-Training for diverse pavement engineering problems. However, input features must be consistent between the pre-trained and fine-tuned models [16]. For the NL-LAB datasets, this consistency means that a physical or empirical model considering all features should exist, which is seldom the case.

The methods discussed for physics infusion in AI are not mutually exclusive [25]. Hence, they can be combined in a single PIAI framework, as detailed in the following subsection.

2.1.4. Hybrid Methods

Hybrid methods for physics infusion refer to combining different infusion methods in a single pipeline, aiming to enhance model generalizability [24]. To the best of the researcher's knowledge, hybrid physics-infusion methods have been developed for engineering domains other than pavement engineering.

Kapoor et al. [35] developed a hybrid modelling framework between transfer learning and causality-respecting PINNs for Euler-Bernoulli and Timoshenko beams on a Wrinkler foundation. The author first trained a causality-respecting model and then used the trained parameters as initialization for similar problems [35]. This framework outperformed other PINN implementation frameworks in predicting displacement and rotations in Wrinkler beams [35]. Moreover, the author concludes that this framework addresses the need to re-train a model when the initial conditions or computational domain change [35]. The hybrid modelling framework developed enhances AI generalization capabilities for extended temporal and spatial domains. However, the characteristics of the NL-LAB datasets do not match with the spatial and temporal domains in the work by Kapoor et al. [35].

Daw et al. [36], developed a hybrid modelling framework combining Physics-Informed Pre-Training and a Physics-Informed Loss Function to predict lake temperature profiles. The author first trained a surrogate model [37] to predict the output of the General Lake Model for a given set of input drivers [36]. The author added a Physics-Informed Loss Function to the surrogate model to enforce consistency between the predicted temperature, water density, and lake depth [36]. The model loss function had one physical component that penalized temperature predictions inconsistent with a monotonic increase in water density with depth [36]. The author concluded that the obtained model showed better generalizability and produced physically meaningful results compared to conventional data-driven models [36]. Although the model developed is distant from pavement engineering, a similar approach can be leveraged to enforce monotonical relationships in predictions based on the NL-LAB datasets.

This section presented four different methods for infusing physics in prediction models according to the relevant literature. The application examples enable understanding possible relationships between the physics-infusion method, the available data and the physical model to infuse. Hence, the following section introduces the NL-LAB program, and the characteristics of the datasets generated for stiffness and fatigue prediction in asphalt mixtures.

2.2. The NL-LAB program

The NL-LAB [6] program started in 2012 aiming for a better understanding of the relationships between functional properties and field performance of Dutch asphalts [38]. Under the NL-LAB [6] program, samples were collected from six road construction projects in the Netherlands [39]. These samples correspond to intermediate and base asphalt layers with recycled asphalt pavement (RAP) [40]. Data collection in the NL-LAB [6] program occurred in phases. Each phase indicates a distinct combination of asphalt mix and compaction setups, as shown in Table 2-1. It is noted that phase 3 includes samples collected at various time intervals, incorporating aging as an additional variable.

Table 2-1. NL-LAB phase overview ([41]).

Phase	Mixing	Compaction	Time Interval	Component Assessed
1	Lab	Lab	After construction	Asphalt and Bitumen
2	Plant			
3	Field	Field		
3a			6 months	Bitumen
3b			1 year	
3c			2 years	Asphalt and Bitumen
3d			6 years	

Table 2-2 details the tests standards used the NL-LAB program to assess pavement functional properties. The testing results, along with mix design and sample identification information, comprise the NL-LAB datasets [6]. The program generated datasets for four functional properties: resistance to fatigue, stiffness, resistance to rutting, and water sensitivity. In line with the scope of this research, Appendix A shows an overview of the raw (unprocessed) features contained in the NL-LAB fatigue and stiffness datasets.

Table 2-2. Testing standards for pavement functional properties.

Material	Test Name	Standard	Temperature	Frequency
Bitumen	Needle penetration	NEN-EN 1426 [42]	25°C	N/A
	Softening point via the ring and ball method	NEN-EN 1427 [43]	N/A	N/A
	Complex shear modulus and phase angle using the Dynamic Shear Rheometer (DSR)	NEN-EN 14770 [44]	20°C	10 rad/s
Asphalt Mix	Bulk density of bituminous specimens	NEN-EN 12697-6 [45]	N/A	N/A

Material	Test Name	Standard	Temperature	Frequency
	Stiffness	NEN-EN 12697-26 [7] Method B	20°C	8 Hz
	Resistance to fatigue	NEN-EN 12697-24 [8] Method D	20°C	8 Hz
	Cyclic compression test	NEN-EN 12697-25 [46] Method B	40°C	1 Hz
	Water sensitivity	NEN-EN 12697-12 [47] Method A	15°C	N/A

Identifying possible physical models for infusion requires a prior understanding of the testing conditions under which the pavement functional properties were obtained. The following section details these conditions, along with possible physical models developed for stiffness and fatigue prediction in asphalt mixtures.

2.3. Pavement functional properties

Testing in pavement engineering is essential to quantify functional properties of bitumen and asphalt mix impact pavement performance [48]. Since bitumen is crucial for resisting tensile stresses within an asphalt mix, bitumen functional properties focus on assessing its capacity to withstand tensile strain without failure [48]. Moreover, testing in asphalt mixtures tests aim to assess the material quality and the suitability of the mixture components [48]. The following subsection elaborates on the tests performed on bitumen samples.

2.3.1. Bitumen Testing

Under the NL-LAB program, tests on bitumen samples included: i). Penetration, ii). Softening point, and iii). Dynamic shear rheometer. This subsection provides an explanation of the corresponding test procedures.

The penetration test measures the distance a standard needle penetrates vertically into a bitumen sample after 5 seconds of loading [42]. The usual measurement units in this test are tenths of a millimetre (1/10 mm) [42]. Higher penetration values indicate a softer bitumen, while lower penetration values are associated with a stiffer bitumen [49]. Although penetration grading was the first standardized bitumen grading system [48], the test approximates bitumen consistency empirically and does not measure any fundamental bitumen property [50].

The softening point of bitumen is determined by the Ring and Ball test [43]. In this test, two 3.5-gram steel balls are placed on bitumen discs in a water bath and heated steadily until they fall 25.0 ± 0.4 mm [43]. The temperature at which this fall occurs is recorded as the bitumen softening point [43]. The bitumen Penetration Index (PI) [51] integrates the results of the penetration test and softening point test [51]. When calculating the PI, it is assumed that bitumen penetration at the softening point is 800 1/10 mm [51]. Lower PI values indicate high-temperature susceptibility, whereas higher PI values indicate low-temperature susceptibility [49]. Since the PI performs bitumen characterization over a small temperature range [49], the understanding of bitumen rheological properties is limited.

The Dynamic Shear Rheometer (DSR) is used to characterize bitumen rheological properties [49]. In the DSR test, an oscillatory shear stress is applied to a bitumen sample sandwiched between two parallel plates [49]. The resulting shear deformation is also oscillatory, with the same frequency as the applied shear stress and a phase lag, as shown in Figure 2-1. The complex shear modulus norm (|G*|) is the ratio between the maximum applied shear stress and the maximum shear strain [49]. The phase angle (δ) is the measured lag between the shear stress and strain plots [49]. The phase angle enables decomposing the complex modulus into the storage (G') and loss modulus (G''), as shown in Figure 2-2.

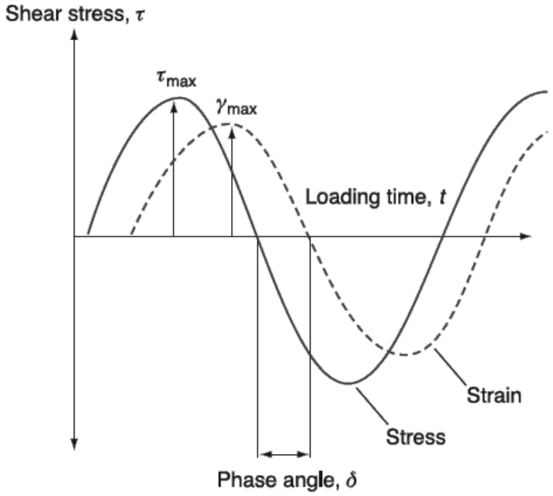


Figure 2-1. Components of the complex modulus ([49]).

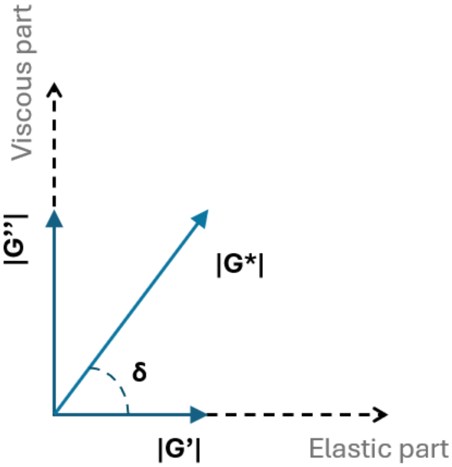


Figure 2-2. Complex modulus decomposition ([48]).

In every load cycle in the DSR test, a hysteresis loop forms in the strain-stress domain [52]. The area enclosed under this hysteresis loop indicates the amount of dissipated energy and is calculated via Equation 2-2 [52].

$$W = \pi \tau_0 \gamma_0 \sin \delta \qquad \text{Equation 2-2}$$

where:

W : dissipated energy

τ_0 : shear stress amplitude

γ_0 : shear strain amplitude

A bitumen sample is subjected to strain-controlled sinusoidal loading in fatigue testing [52]. Assuming all the strain energy dissipates to control fatigue cracking, Equation 2-2 can be rewritten as Equation 2-3. From Equation 2-3, it is noted that a lower value of |G*|sinδ reduces the dissipated energy. This observation led to the introduction |G*|sinδ as a parameter to minimize for controlling fatigue cracking in the Superpave design method [52].

$$W = \pi \gamma_0^2 \left| G^* \right| \sin \delta \qquad \text{Equation 2-3}$$

Although bitumen functional properties are relevant to assess pavement performance, further testing in asphalt mixtures is fundamental for a comprehensive analysis [48]. The following subsection presents the test procedures for asphalt mix under the NL-LAB program.

2.3.2. Asphalt Mix Testing

Testing procedures on asphalt mix samples under the NL-LAB program included: i). Density and air voids, ii). Fatigue resistance, iii). Stiffness, iv). Rutting resistance, and v). Water sensitivity. Given the scope of this research, the testing procedures for rutting resistance and water sensitivity are not detailed in this section.

2.3.2.1. Density and Air Voids

The air voids content of an asphalt mix is the ratio between the volume of air voids in the mix and the bulk volume of the compacted mix [48]. The air voids content was determined using the bulk density of the compacted mix and the theoretical maximum mix density [53]. The bulk density of the compacted mix corresponds to a Saturated Surface Dry (SSD) condition [45], whereas the theoretical maximum mix density was obtained via the volumetric procedure specified in NEN-EN 12697 [54]. Data regarding density and air voids content in the NL-LAB datasets was collected by the contractors involved in the program [6].

Density and air voids content often correlate to pavement performance. Researchers [48] related low air voids content with higher rutting probability and high air voids content with premature cracking and ravelling in asphalt mixtures. The effects of premature cracking on pavement life were assessed by Linden et al. [55], finding that an increase of 1% in air voids content reduced pavement service life by up to 10%. On the contrary, a lower air void content is linked to higher asphalt mix densities [48]. Mogawer et al. [56] found higher asphalt mix densities to increase the mix dynamic modulus for different temperatures and frequencies [56].

2.3.2.2. Mix Stiffness

Stiffness quantifies the ability of an asphalt mix to dissipate an applied load [49]. In a pavement structure, two material layers with the same thickness but different stiffness will transfer different load magnitudes to the underlying layer [49]. Notably, a stiffer asphalt layer will be more prone to fatigue cracking, whereas a layer with low stiffness tends to be more affected by rutting. Hence, stiffness is a key functional property for designing an adequate pavement structure for specific loading, environmental, and site conditions.

An elementary definition of stiffness is given by the stress-strain ratio shown in Equation 2-4 [7]. Since an asphalt mix is a viscoelastic material, its loading response is temperature and frequency-dependent [7]. Hence, for a linear visco-elastic material, stiffness is defined in terms of a complex modulus (E*) and a phase angle (δ), as shown in Equation 2-5.

$$E = \frac{\sigma}{\varepsilon}$$ Equation 2-4

where:

E : stiffness modulus
σ : Maximum applied stress
ε : Maximum measured strain response

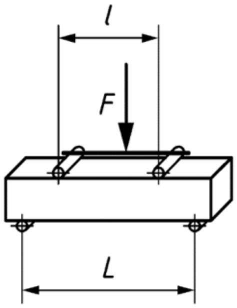
$$E^* = \left|E^*\right| \cdot \left(\cos\left(\delta\right) + i \cdot \sin\left(\delta\right)\right)$$ Equation 2-5

where:

$\left|E^*\right|$: Norm of the complex modulus, defined by the stress strain ratio.

δ : Phase angle.

Under the NL-LAB program stiffness was determined using the 4PBB test [7], following the setup shown in Figure 2-3. This test applies periodic bending with constant amplitude (see Figure 2-4) to an asphalt sampled conditioned at a given temperature [7]. At the 100th load cycle, the norm of the complex modulus is determined via Equation 2-4 [7]. The resulting value is the target of the NL-LAB stiffness dataset and corresponds to the mix stiffness at 20°C and 8 Hz [40].



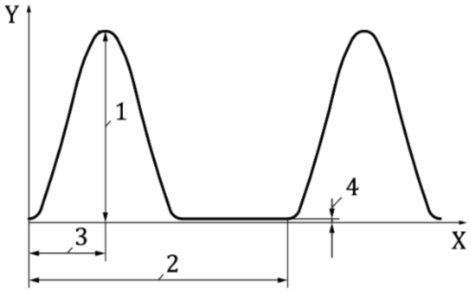
where:

F: Applied load.

L: Distance between supports.

l: Distance between loading points.

Figure 2-3. 4PBB test schematization. Source: [7].



where:

1. Peak load
2. Pulse repetitions period
3. Loading time
4. Minimum Load

X. Time

Y. Force

Figure 2-4. Applied load in the 4PBB test. Source: [7].

Research on asphalt stiffness modulus prediction has been extensively carried out since the 1950s [57], with continuous development to the present day. A set of stiffness prediction models incorporates homogenization theory, where the effective stiffness of a composite material is determined based on the stiffness of its different phases [58]. Examples of this model family are the Christensen model [59], the modified Christensen model [60], and the model proposed by Zhang et al. [58]. Other relevant stiffness prediction models include the developments by Shell [49], the University of Nottingham [61], the U.S. Asphalt Institute [62], and Witczak & Fonseca [63]. It is noted that none of the reviewed physical models include all features of the NL-LAB datasets.

2.3.2.3. Fatigue Resistance

Asphalt fatigue is a pavement distress characterized by a series of interconnected cracks [51]. Fatigue cracking initiates at the bottom of the asphalt layer, where the tensile strain is highest, and propagates upwards as one or more parallel longitudinal cracks [51]. Fatigue cracking indicates pavement structural deficiencies, especially in areas subjected to traffic loading [51].

Under the NL-LAB program, fatigue resistance is quantified using the 4PBB test [40]. After determining the mix initial stiffness, the test applies periodic bending until reaching a failure condition [8]. This failure condition corresponds to a 50% reduction in the initial stiffness [8]. The fatigue life of a sample is the number of cycles needed to reach the failure condition (Nf_{50}) [8]. After testing multiple samples, it is possible to plot a fatigue line relating the applied initial strain and the fatigue life of each sample in the logarithmic plane [8]. Fitting a linear regression to the fatigue line eases the computation of the initial strain corresponding to a fatigue life of $1x10^6$ cycles. This initial strain is labelled as ε_6 and is the target of the NL-LAB fatigue dataset.

Different authors correlated fatigue resistance and other bitumen or mix properties. Bahia et al. [64] studied the relationship between the Superpave fatigue parameter ($|G^*|\sin\delta$) and the fatigue life in mixtures with Polymer Modified Bitumen (PMB). The author found weak correlations considering different aggregate types and mix compositions, which they linked to measuring fatigue only in the linear viscoelastic range [64]. Ishaq & Giustozzi [65] developed a correlation between bitumen fatigue life and HMA fatigue life in the 4PBB test, obtaining an R^2 value of 0.82. Shen & Carpenter [66] developed a model correlating the fatigue life with the Plateau Value (PV) from energy dissipation theory. The author also devised a model to correlate the PV with the HMA tensile strain, flexural stiffness, volumetric composition, and aggregate characteristics [66]. Similar to mix stiffness, physical models for fatigue resistance prediction do not incorporate all features in the NL-LAB datasets.

This subsection concludes the review on the test protocols and physical models for stiffness and fatigue resistance in asphalt mixtures. It was found that the reviewed physical models do not include the entirety of the features in the stiffness and fatigue datasets. In the following section, the literature review is concluded by selecting a physics infusion suitable to the characteristics of the NL-LAB datasets and the physical models reviewed.

2.4. Conclusion

Table 2-3 summarises the PIAI studies reviewed in this research. Literature suggests a relationship between framework for physics infusion, the data available, and the physical model to infuse. Using a Physics-Informed Loss function is possible when handling tabular inputs, as the effect of a subset of features on the model output can be constrained. However, PINNs are preferred when a model contains partial differential equations, as exemplified in [29], [30], [33] and [35]. Hence, to leverage PINNs, data instances should be correlated in time or space. Moreover, specifying a Physics-Informed Architecture is crucial to account for possible correlations between data instances. The example shown in [33] devises an architecture fitted to capture the temporal dynamics of the problem at hand. Physics-informed pre-training can handle tabular data, with or without temporal or spatial correlation between data instances. The application developed in [16] demonstrated that Physics-Informed Pre-Training is possible when the features for physics pre-training and data-driven fine tuning are the same.

Table 2-3. Examples of PIAI applications.

Author	Framework Used	Physical/Empirical Model Infused	Features	Targets
Deng et al. [15]	Physics-Informed Loss Function	Monotonic increasing constraint of rut depth predictions considering temperature and number of wheel passes.	Tabular data containing bitumen and aggregate properties. No correlations in time or space identified.	Rut depth
Han et al. [18]	Physics-Informed Loss Function	Visco-Elastic Continuum Dynamics.	Historical data of fatigue C-S curve as a time series. Data instances are temporally related.	Pseudo-Stiffness (C).
Kapoor et al. [29]	Physics-Informed Loss Function Specifically, PINN	Euler-Bernoulli beam equation. Timoshenko beam equation.	Numerically generated data points to verify compliance of governing PDE, boundary conditions and initial conditions.	Beam displacements.
Kapoor et al. [30]	Physics-Informed Loss Function Specifically, PINN	Euler-Bernoulli beam equation.	Numerically generated data points to verify compliance of governing PDE, boundary conditions and initial conditions.	Beam displacements.
Kapoor et al. [33]	Physics-Informed Architecture	Viscous Burgers equation. Allen-Cahn equation.	Numerically generated data points to verify compliance of governing PDE, boundary	Unspecified.

Author	Framework Used	Physical/Empirical Model Infused	Features	Targets
		Nonlinear Schrödinger equation. Euler-Bernoulli beam equation.	conditions and initial conditions. Also, numerically generated data to validate model generalization.	
Kargah-Ostadi et al. [16]	Physics-Informed Pre-Training	Quarter-Car Model.	Tabular data of vehicle properties, FHWA LTPP road profiles, and synthetic vertical acceleration responses. Input data is correlated in space and time.	IRI values.
Kapoor et al. [35]	Hybrid Approach Physics-Informed Pre-Training and PINN	Euler-Bernoulli beam equation. Timoshenko beam equation.	Numerically generated data points to verify compliance of governing PDE, boundary conditions and initial conditions.	Beam displacement.
Daw et al. [36]	Hybrid Approach Physics-Informed Pre-Training and Physics-Informed Loss Function	General Lake Model. Monotonic increasing relation of water density with respect to depth.	Tabular inputs relating to environmental and time-of-the -year conditions. Lake depth is also an input. Thus, instances are spatially correlated.	Lake temperature profile.

The characteristics of the NL-LAB datasets limit the applicability of some reviewed infusion methods. These datasets contain instances that are not correlated in time or space, which hinders the use of PINNs and a Physics-Informed Architecture. Moreover, models developed in pavement engineering for fatigue and stiffness predictions rarely contain all features present in the datasets, challenging a Physics-Informed Pre-Training. However, some fatigue and stiffness physical models contain a subset of features present in the datasets. Hence, by specifying the effect of a subset of features in AI model predictions, physics can be infused in AI model predictions. In consequence, a Physics-Informed Loss Function is the selected physics-infusion method.

The literature review concludes with the selection of a suitable physics-infusion method for the NL-LAB datasets' characteristics. However, the development of a PIAI framework for stiffness and fatigue predictions necessitates further considerations regarding the desired physical model and PIAI model training. These considerations are detailed in the following chapter.

3. PIAI Framework for the NL-LAB datasets

This chapter presents the developed PIAI framework for prediction models based on the NL-LAB datasets, corresponding with Stage 3 of the research methodology (see Figure 1-1). Figure 3-1 shows the four stages for model development under the PIAI framework. These stages are explained across four distinct sections in this chapter. Section 3.1 presents the developed criteria for selecting a physical model, answering the second research sub question. Section 3.2 introduces the different steps performed for data preparation. Then, Section 3.3 introduces the physics and data-driven feature selection method incorporated in the PIAI framework. Finally, Section 3.4 provides a detailed description of the procedures for model training and optimization under the development framework.

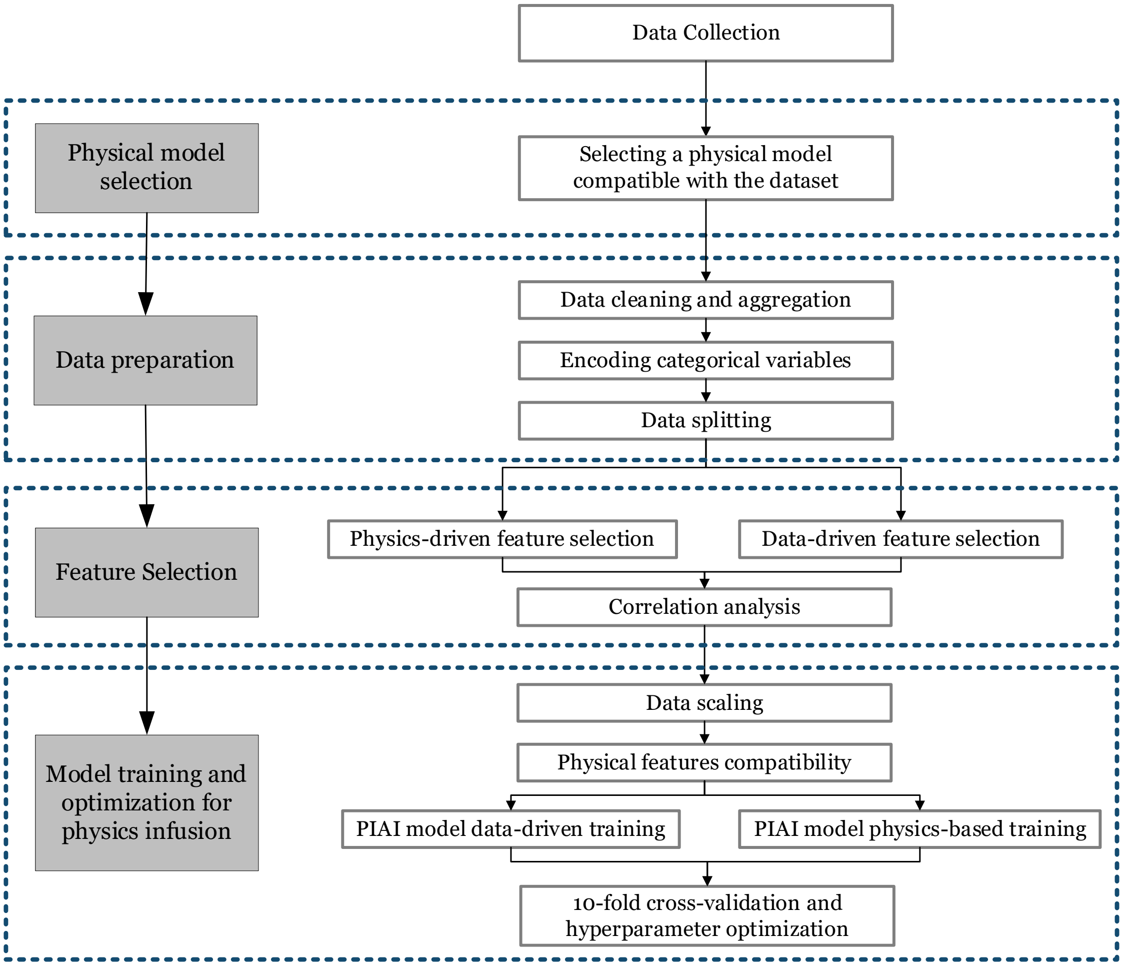

Data Collection
Physical model selection
Selecting a physical model compatible with the dataset
Data preparation
Data cleaning and aggregation
Encoding categorical variables
Data splitting
Feature Selection
Physics-driven feature selection
Data-driven feature selection
Correlation analysis
Model training and optimization for physics infusion
Data scaling
Physical features compatibility
PIAI model data-driven training
PIAI model physics-based training
10-fold cross-validation and hyperparameter optimization

Figure 3-1. Proposed PIAI model development framework for the NL-LAB datasets.

3.1. Selection of a Physical Model

Datasets generated under the NL-LAB program consider a substantial number of features that aim to explain pavement performance (see Appendix A – Table A 1). To the best of the researcher's knowledge, there is seldom a physical model in pavement engineering containing the same features and targets as the NL-LAB datasets. Hence, this research established the following compatibility criteria for selecting a physical model:

- The dependent and independent variables of the physical model must be among the dataset features. Otherwise, it must be possible to calculate the physical model variables from the dataset features.

- The physical model must include the target variable of the dataset. Preferably, the dataset target should be included explicitly as a variable in the physical model. However, physics infusion is also possible using a physical model considering a variable calculated from the dataset target.

- The testing conditions of the physical model should closely match the testing conditions of the dataset. If not, it should be possible to implement a correction. Examples of corrections include using master curves to account for different temperature and loading conditions. Failure to meet this requirement does not prevent physics infusion in AI models but can lead to inconsistencies.

- The physical model should be a validated development in the pavement engineering domain. Like in the previous requirement, failing to meet this requirement does not prevent physics infusion in AI models. However, it poses a risk of infusing an inadequate physical behaviour.

These compatibility criteria are further explained through an example. Consider a dataset with 5 features (X1,..., X5) and a target (Y) as shown in Table 3-1. Each feature and the target contain k entries ($X1_1$,..., $X1_k$). Consider also the candidate physical model presented in Equation 3-1. The independent variable of the candidate physical model corresponds to the target of the toy dataset. Additionally, the toy candidate physical model calculates the independent variable as a linear combination of functions of a subset of the features in the dataset (X1, X2). Under the assumption that the features X1, and X2 are independent and uncorrelated, it is possible to infuse the toy candidate model in an AI prediction model.

Table 3-1. Example dataset for explaining physical model compatibility.

Input Features (X_D)					Target
X1	X2	X3	X4	X5	Y
$X1_1$	$X2_1$	$X3_1$	$X4_1$	$X5_1$	Y_1
$X1_2$	$X2_2$	$X3_2$	$X4_2$	$X5_2$	Y_2
$X1_3$	$X2_3$	$X3_3$	$X4_3$	$X5_3$	Y_3
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
$X1_k$	$X2_k$	$X3_k$	$X4_k$	$X5_k$	Y_k

$$Y = C_1 \cdot f(X_1) + C_2 \cdot g(X_2) + C_3 \qquad \text{Equation 3-1}$$

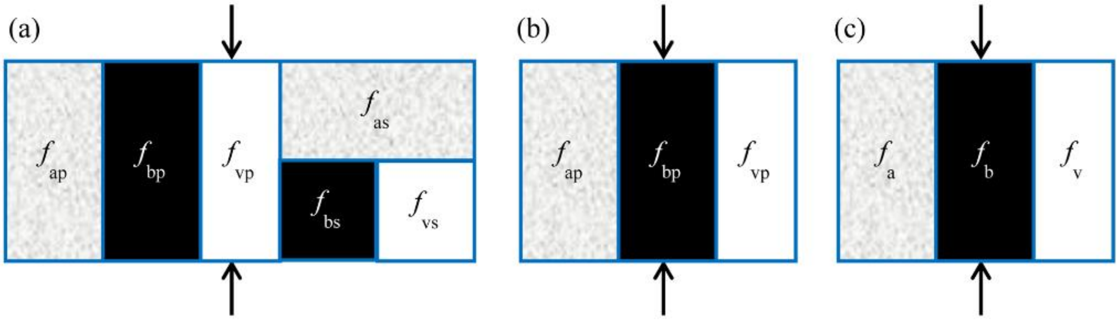
where:

Y: Independent variable of the dataset.
f(X_1): Function of feature X_1.
g(X_2): Function of feature X_2.
C_1, C_2, C_3: Model constants.

Selecting a physical model conforming to the abovementioned compatibility criteria is fundamental for enhancing pavement performance predictions. In the following subsections, the selected physical models to infuse in the PIAI prediction models for stiffness and fatigue are presented.

3.1.1. Physical model for stiffness prediction based on micromechanics

The selected physical model to infuse in the PIAI stiffness prediction model was proposed by Zhang et al. [58] for Open Graded Mixes (OGMs). This model belongs to a family of models based on homogenization theory [58]. In homogenization theory, the effective micromechanical properties of a composite material are determined based on the properties of its different phases [58]. In an asphalt mix three phases can be identified [59]: bitumen, aggregates, and air voids. Figure 3-2 shows three phase arrangements used by researchers ([58], [59], [60]) in homogenization models.



where:

f: Phase.	a: Aggregate.
p: Parallel component.	b: Bitumen.
s: Series component.	v: Air voids.

Figure 3-2. Arrangement of the Christensen model for asphalt mixes. From [58].
a). Original arrangement [59], b). Simplified arrangement [60], c). Proposed modified arrangement [58].

Christesen et al. [59] developed the first homogenization theory application in pavement engineering, devising the phase arrangement shown in Figure 3-2(a) for improved accuracy in mix stiffness predictions. Later, Christensen & Bonaquist [60] found the series component of this initial arrangement to be futile for mix stiffness prediction, and proposed the parallel arrangement shown in Figure 3-2(b).

Zhang et al. [58] proposed a revised parallel arrangement to better account for aggregate contact interaction. This revised arrangement considers the total volume of bitumen, aggregates, and air voids, as shown in Figure 3-2(c). Moreover, the author introduced the Aggregates Organization Factor (P_a) to describe the frequency and temperature-dependent contribution of the aggregate phase in mix stiffness prediction. The resulting model for mix stiffness prediction is shown in Equation 3-2.

$$\left|E^*\right|_{mix}(f) = P_a(f)\, f_a E_a + 3 f_b \left|G^*\right|_b (f) \qquad \text{Equation 3-2}$$

where:

$|E^*|_{\text{mix}}(f)$: Mix stiffness, frequency dependent.

$P_a(f)$: Aggregate organization factor, frequency dependent.

f_a: Volume fraction of the aggregate phase.

E_a: Aggregate Young's modulus.

f_b: Volume fraction of the bitumen phase.

$|G^*|_b(f)$: Dynamic shear modulus of the bitumen phase, frequency dependent.

Zhang et al. [58] validated the proposed model by developing master curves for mix stiffness and bitumen shear modulus. Through this validation, the author demonstrated that P_a is frequency dependent, approximating a sigmoidal function when frequency increases [58]. Thus, the author proposed the regression model shown in Equation 3-3.

$$P_a = a + (1-a)\cdot \frac{\exp\left(b + c\ln(f_b)/(f_b + f_v)\cdot \left|G^*\right|_b + d(f_b + f_v)\right)}{1+\exp\left(b + c\ln(f_b)/(f_b + f_v)\cdot \left|G^*\right|_b + d(f_b + f_v)\right)} \quad \text{Equation 3-3}$$

where:

f_v: Volume fraction of the air voids phase.

a, b, c, d: Regression coefficients. $a = 0.0017, b = 0.62, c = 0.72, d = -0.17$

$|G^*|_b$, and f_b are described in Equation 3-2.

The model developed by Zhang et al. [58] satisfied the compatibility criteria for physical model selection. The target of the NL-LAB stiffness dataset is incorporated in Zhang et al [58]'s model as independent variable. The model variables f_a, f_b and $|G^*|_b$ correspond to the features "VA", "Volume_Target_Bitumen" and "bit2_Gstar" of the NL-LAB stiffness dataset. Furthermore, This research assumed 53000 MPa as the Young's modulus of the aggregate phase (E_a) [58]. Finally, the volume fraction of the aggregate phase was calculated using Equation 3-4, and stored in a new feature named "Volume_Agg_Fraction".

$$f_a = 1 - f_b - f_v \quad \text{Equation 3-4}$$

f_a, f_b, and f_v are described in Equation 3-2 and Equation 3-3.

After performing the abovementioned calculations, P_a was obtained via Equation 3-3 and stored in a new feature named "Pa". With P_a, the expected mix stiffness was calculated via Equation 3-2. The PIAI stiffness model was trained to approximate this expected mix stiffness in the physics component.

The selection of Zhang et al.'s [58] model as the physical model to infuse in stiffness predictions entailed three limitations for this research. The first limitation was related to the stiffness test procedure. While Zhang et al. [58] used the Cyclic uniaxial compression test [67] to obtain mix stiffness, the stiffness values reported in the NL-LAB dataset correspond to the 4PBB test [7]. Therefore, it was assumed that the stiffness values obtained through both testing procedures are comparable. The second limitation was brought by the regression model introduced by Zhang et al. [58] (see Equation 3-3) being applicable only to OGMs. Hence, to bridge this limitation it was assumed that the samples contained in the NL-LAB stiffness dataset correspond to OGMs. The third and last limitation was related to the calculation of fa, as it was implicitly assumed that no bitumen was absorbed by the aggregates.

3.1.2. Physical model for fatigue prediction based on energy dissipation theory

The selected physical model to infuse in the PIAI fatigue model was developed by Shen & Carpenter [66]. Using energy dissipation theory, the authors proposed a prediction model relating the Plateau Value (PV) to the initial flexural stiffness, and parameters referring to mix volumetrics and aggregate gradation [66].

Energy dissipation theory states that the area under the stress-strain curve represents the energy applied to a material under loading conditions [66]. This applied energy fully recovers when the unloading curve follows the same path as the loading curve [66]. When the loading and unloading paths differ, a hysteresis loop is formed. In viscoelastic materials, the as the area of this hysteresis loop corresponds to the dissipated energy (see Equation 3-5).

$$W_i = \pi \cdot \sigma_i \cdot \varepsilon_i \cdot \sin\left(\phi_i\right) \qquad \text{Equation 3-5}$$

where:

W_i: Energy dissipated in cycle i.

σ_i: Stress level in cycle i.

ε_i: Strain level in cycle i.

ϕ_i: Phase angle in cycle i.

Material damage is related to changes in dissipated energy [66]. Two different hysteresis loops in successive load cycles indicate material damage occurred [66]. Only the relative amount of energy dissipation created by each additional load cycle will produce further material damage [66]. In consequence, the relative change of dissipated energy has a direct relationship with damage accumulation [66]. Hence, the Ratio of Dissipated Energy Change (RDEC) (see Equation 3-6) is often used as a testing parameter to assess HMA fatigue damage [66].

$$RDEC = \frac{DE_{i+1} - DE_i}{DE_i} \qquad \text{Equation 3-6}$$

where:

RDEC: Ratio of Dissipated Energy Change.

DE_i: Energy dissipated in cycle i.

A curve with three distinct zones (see Figure 3-3) is obtained when plotting the RDEC and the number of loading cycles [68]. The Plateau Value (PV) corresponds to the constant RDEC value shown in zone II. The PV is often determined as the RDEC value corresponding to the load cycle in which the initial stiffness is reduced by 50% [66].

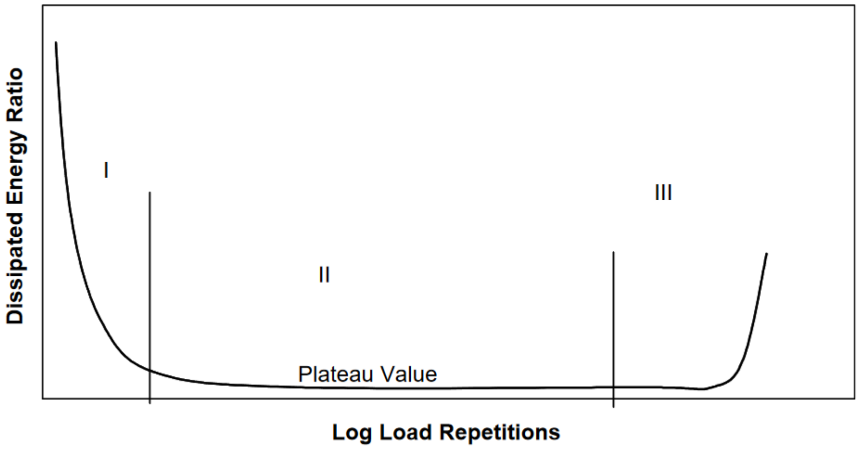


Figure 3-3. RDEC vs Load Cycles. From [68].

Shen & Carpenter [66] proposed a PV prediction model using 19 mixtures from the Illinois Department of Transportation [66]. These mixtures presented various air void contents, different gradations, and both neat and polymer-modified bitumen [66]. Mixtures were compacted using a rolling wheel compactor and tested via a four-point bending beam fatigue test according to AASHTO T321-03 [66]. This test applies a constant strain amplitude to the mix at a frequency of 10 Hz and a temperature of 20°C [66]. Equation 3-7 shows the developed PV prediction model.

$$PV = 44.422\varepsilon^{5.140} S^{2.993} VP^{1.850} GP^{-0.4063} \qquad \text{Equation 3-7}$$

where:

PV: Plateau Value.
ε: Tensile strain.
S: HMA initial flexural stiffness (20°C, 10Hz) [MPa].
VP: Volumetric parameter, described in Equation 3-8.
GP: Aggregate gradation parameter, described in Equation 3-9.

$$VP = \frac{AV}{AV + V_b} \qquad \text{Equation 3-8}$$

where:

VP: Volumetric parameter.
AV: Mix air voids.
V_b: Bitumen content by volume.

$$GP = \frac{P_{NMS} - P_{PCS}}{P_{200}} \qquad \text{Equation 3-9}$$

where:

GP: Aggregate gradation parameter
P_{NMS}: Percentage of aggregate passing the nominal maximum sieve size.
P_{PCS}: Percentage of aggregate passing the primary control sieve (PCS = 0.22·NMS).
P_{200}: Percentage of aggregate passing the No. 200 (75μm) sieve size.

Shen & Carpenter [66] also proposed a relationship between the PV and the number of load cycles to failure, shown in Equation 3-10. The authors verified this relationship to be unique for different mixtures, loading modes, and testing conditions at normal damage levels [66]. Moreover, the authors employed the same failure condition used in the NL-LAB fatigue dataset [66].

$$Nf = 0.4801 \cdot PV^{-0.9007} \qquad \text{Equation 3-10}$$

where:

N_f: Fatigue life, i.e. number of cycles for a 50% stiffness reduction.
PV: Plateau value.

The model developed by Shen & Carpenter [66] satisfied the compatibility criteria for physical model selection. The target of the NL-LAB fatigue dataset is one of the dependent variables of the selected physical model. Moreover, the initial stiffness was retrieved from the NL-LAB stiffness dataset. Additionally, mix stiffness was obtained by. The volumetric parameter (see Equation 3-8) was calculated using "VA" and "Volume_Target_Bitumen". The calculated value for this parameter was stored in a new feature named "VP". Finally, the aggregate gradation parameter (see Equation 3-9) was obtained by constructing aggregate gradation curves from the target composition values (see Appendix B). The calculated value for this parameter was stored in a new feature named "GP".

Since the target of the NL-LAB fatigue dataset is ε_6, the number of cycles to failure is $N_f = 1\text{x}10^6$. With the number of cycles to failure, the PV was calculated via Equation 3-10. Furthermore, Equation 3-7 was reworked to isolate ε from the calculated PV. Hence, the PIAI model for fatigue prediction was trained to approximate this expected stiffness in the physics component.

This research incurred in two limitations when selecting Shen & Carpenter's [66] model as the physical model to infuse in fatigue predictions. The first limitation is related to the testing procedure. Although the 4PBB test was used in both the physical model and the NL-LAB dataset, the reference cycle for obtaining the initial stiffness differs. In the NL-LAB dataset, initial stiffness was determined at the 100th load cycle [7]. However, Shen & Carpenter's model uses an initial stiffness value obtained at the 50th load cycle [69]. Therefore, it is assumed that mix stiffness values determined via the 4PBB test at the 50th and 100th load cycle are comparable. The second limitation is related to the determination the Nominal Maximum Aggregate Sieve Size (NMAS). This research found no information about the NMAS of the mixtures tested in the NL-LAB program. Therefore, a NMAS of 16mm was assumed based on the aggregate target composition, as illustrated in Appendix B.

This subsection introduced the selected physical models for infusion in the PIAI stiffness and fatigue models. Although selecting a physical model is fundamental for PIAI model development, raw data needs to be prepared for use in predictive models. The following section presents the data preparation procedures performed in this research.

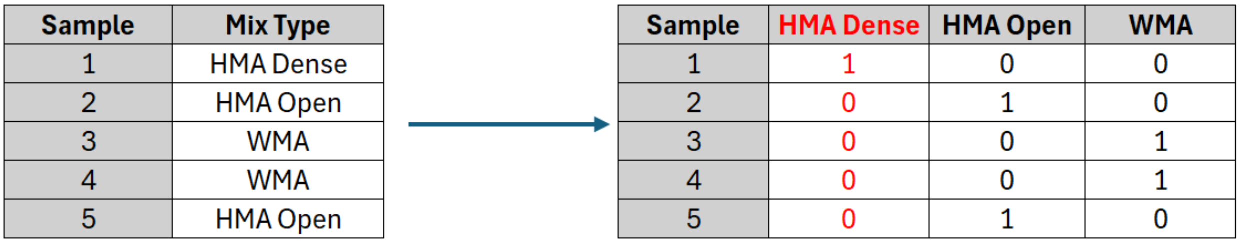
3.2. Data Preparation

Data preparation is a fundamental step prior to model training as raw data is rarely suitable for direct use in AI predictive models [70]. This research introduced different data preparation procedures to ensure that input data was usable for PIAI model training. Data preparation comprised data cleaning and aggregation, encoding categorical features, and data splitting.

Data cleaning involved removing informational features (see Appendix A - Table A 1) and instances containing missing entries from the datasets. The categorical features “work” and “phase” were removed as each “work” and “phase” combination can be represented by a “mix_setup” and “comp_setup” combination, as seen in Appendix C.

Furthermore, data aggregation included the generation of gradation curves from the target aggregate mass composition values in the datasets. With the gradation curves, the volume percentages of gravel (retained in 4.75 mm sieve [71]), and sand (passing the 4.75 mm sieve but retained in the 0.075 mm sieve [71]) were determined. These percentages were included in the datasets as “Volume_Target_Gravel” and “Volume_Target_Sand”, in lieu of the target aggregate composition values per sieve size.

Encoding categorical features is another crucial data preparation step. Categorical features contain class values instead of numerical values [70]. These class values are often text or numerical inputs referring to categories [70]. Encoding a categorical feature strives to assign numerical values to represent each category contained in the feature [70]. Methods for encoding a categorical feature include one-hot encoding, ordinal encoding, feature hashing, and target encoding or bin counting [70], [72]. After consideration of the drawbacks and benefits of each encoding method, this research used one-hot encoding for handling categorical variables. Although one-hot encoding generates additional binary features for each possible category, it provides better interpretability over the effects of each category in model predictions. Figure 3-4 illustrates an example of one-hot encoding. In the example, “Mix Type” is a categorical feature with 3 (k = 3) possible categories. One-hot encoding creates k binary features, corresponding to k possible categories. However, one-hot encoding creates a linear dependency between the binary features, as knowing the values of k-*1* features enables deducing the value of the k^{th} feature [72]. Therefore, it is necessary to remove one feature which serves as a benchmark or reference category. In the example, the category “HMA Dense” is the benchmark for the “Mix Type” feature.



Sample	Mix Type
1	HMA Dense
2	HMA Open
3	WMA
4	WMA
5	HMA Open

Sample	HMA Dense	HMA Open	WMA
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	1
5	0	1	0

Figure 3-4. Example of One-Hot Encoding.

Data cleaning, data aggregation, and encoding categorical variables involved the creation and removal of features in the NL-LAB datasets. Table 3-2 summarizes the changes in the dataset features and a brief explanation of the justification behind these changes. It is noted that the data preparation procedures resulted in a net addition of 2 features to the dimensionality of the NL-LAB datasets. Moreover, Appendix A (Table A 2) provides a list of the obtained features after data preparation, along with relevant descriptive statistics.

Table 3-2. Added and removed features from each dataset.

Dataset	Features	Added / Removed	Justification
Stiffness	"Volume_Aggregate_Fraction", "Pa"	Added	Needed for the stiffness physical model
Fatigue	"VP", "GP",	Added	Needed for the fatigue physical model
Both	"Volume_Target_Gravel", "Volume_Target_Sand"	Added	Aggregated gradation features.
Both	"Volume_Target_C22_4", "Volume_Target_C16", "Volume_Target_C11_2", "Volume_Target_C08", "Volume_Target_C05_6", "Volume_Target_C002mm", "Volume_Target_C00063mu"	Removed	Contained in the aggregated gradation features.
Both	"work", "phase"	Removed	Reflected by specific "mix_setup" and "comp_setup" combinations
Both	"mix_setup"	Removed	Categorical feature
Both	"Forced action mixer", "Planetory mixer"	Added	Categories of the "mix_setup" feature. Benchmark category: "Asphalt plant"
Both	"comp_setup"	Removed	Categorical feature

Dataset	Features	Added / Removed	Justification
Both	“Hand roller”, “Mini roller”, “Segment compactor”, “Shear box”	Added	Categories of the “comp _setup” feature. Benchmark category: “Field roller”
Both	“PMB”	Added	Since work 6 included and the “work” feature was removed, this binary indicator accounts for PMB presence in the asphalt mix.

Data splitting is the final step of the data preparation stage. Training an AI model using all available data may lead to overfitting and undermine model performance [70]. Hence, splitting data into training, validation, and test sets can prevent overfitting. This research utilized a ratio of 80%-10%-10%, as shown in Figure 3-5. The training, validation, and test subsets play different roles in AI model development. The training set helps obtain the optimal model parameters [73]. The validation set allows finding the optimal model architecture through hyperparameter tuning [73]. The test set enables the acquisition of unbiased performance metrics [73].

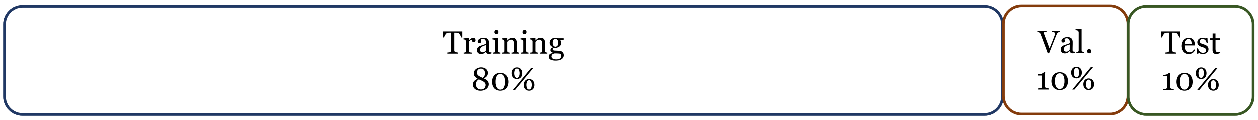


Figure 3-5. Training-validation-test split.

Through data preparation, raw features in the NL-LAB datasets were transformed to facilitate incorporation in PIAI prediction models. Data preparation resulted in 22 and 23 features for the NL-LAB stiffness and fatigue datasets respectively. Since the datasets contain 425 instances, it was decided to reduce the number of input features to prevent overfitting in the PIAI models. This research reduced the number of input features with a feature selection procedure, as explained in the following section.

3.3. Feature Selection

Figure 3-6 illustrates the principle of the feature selection method performed in this research. The outcome of the feature selection analysis is the set of important features. The important feature set is a subset of the feature space and is formed by the union of the sets of important features from a physics and a data-driven perspective.

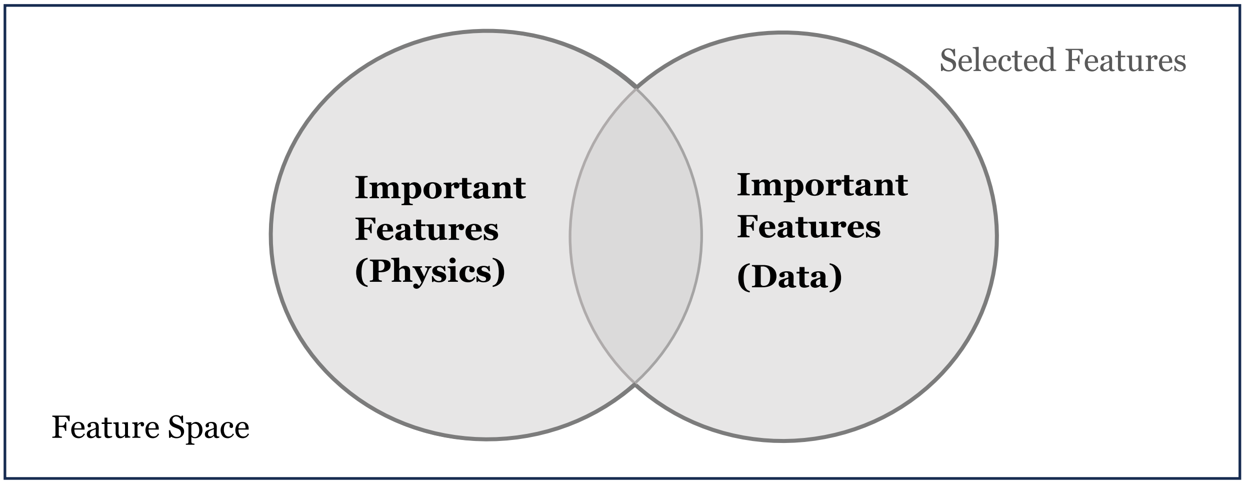


Figure 3-6. Illustration of the feature importance analysis.

This research introduced a feature selection method comprising physics and data-driven components. While the physics component aims to preserve all features needed to compute the outputs of the physical model, the data component aims to preserve features with strong statistical relationships to the target [70]. A feature required to compute the output of a physical model must not be removed irrespective of the results of the data component. The following section provides a description of the selected method for feature selection in the data-driven component.

3.3.1. Data-driven feature selection through BorutaShap

This research used BorutaShap [74] as the data-driven feature selection method. BorutaShap is a wrapper [70] method that brings together the Boruta [75] feature selection algorithm with SHAP [76] values. Boruta is an iterative selection method that compares the importance score [70] of the features in a dataset with the maximum importance score achievable by pure randomness [77]. The importance score in BorutaShap is obtained with SHAP [76] values. SHAP values leverage a sound mathematical foundation on cooperative game theory to quantify the contribution of each feature in model predictions (see Appendix D) [78].

By incorporating statistical testing to determine feature importance, BorutaShap eliminates the need for presetting thresholds related to minimum model performance or a desired number of features [74]. Such thresholds are typically required in other wrapper methods, including Forward Selection [79] and Recursive Feature Elimination (RFE) [79]. The principles underlying BorutaShap, as well as its relevance to this research, are best illustrated through an example.

Consider a dataset with 5 features (X1, ..., X5), 1 target (Y) and 5 instances, as shown in Table 3-3. After fitting a tree-based regressor to this dataset, feature importance scores are calculated using the mean SHAP value across all predictions [80], as shown in Figure 3-7. These importance scores can be integrated into different feature selection methods. In Forward selection, a prediction model is constructed using the most important feature (X1) and the target (Y) [79]. If this model fails to meet the preset performance criteria, the next most important feature (X2) is added, and the model is retrained [79]. Conversely, RFE removes the least important feature (X5) first and retrains the models [79]. With RFE, the least important feature is successively removed until the model reaches the desired performance threshold or until the feature number is reduced to the preset minimum [79].

Table 3-3. Example dataset for BorutaShap illustration.

Dataset Features (X_D)					Target
X1	X2	X3	X4	X5	Y
$X1_1$ $X1_2$ $X1_3$ . . . $X1_k$	$X2_1$ $X2_2$ $X2_3$ . . . $X2_k$	$X3_1$ $X3_2$ $X3_3$ . . . $X3_k$	$X4_1$ $X4_2$ $X4_3$ . . . $X4_k$	$X5_1$ $X5_2$ $X5_3$ . . . $X5_k$	Y_1 Y_2 Y_3 . . . Y_k

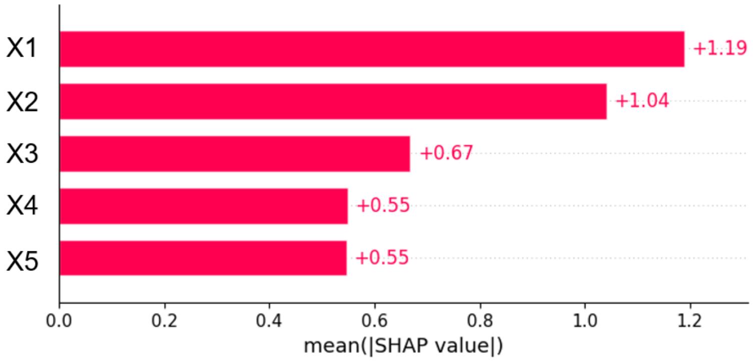


Figure 3-7. Example SHAP values for BorutaShap illustration (*[81]*).

BorutaShap differs from Forward Selection and RFE by identifying as important only those features that have been statistically proven to enhance model performance [74]. BorutaShap extends the original feature space by creating one shadow feature per original feature using random permutation [82], as shown in Table 3-4. A tree-based regressor is then fitted to this extended feature set and the target variable [74]. This trained regressor is used to compute importance scores for both the original and shadow features using SHAP values [76], as shown in Table 3-5. An original feature is considered to have scored a “hit” if its importance score is higher than the maximum importance score of all shadow features [82]. In the given example, X1, X2 and X3 scored a “hit”.

Table 3-4. Example dataset extended with shadow features.

Extended Features						Target
X1	X2	X3	Shad. X1	Shad. X2	Shad. X3	Y
$X1_1$	$X2_1$	$X3_1$	$X1_2$	$X2_1$	$X3_2$	Y_1
$X1_2$	$X2_2$	$X3_2$	$X1_4$	$X2_5$	$X3_1$	Y_2
$X1_3$	$X2_3$	$X3_3$	$X1_5$	$X2_4$	$X3_5$	Y_3
$X1_4$	$X2_4$	$X3_4$	$X1_1$	$X2_3$	$X3_3$	Y_4
$X1_5$	$X2_5$	$X3_5$	$X1_3$	$X2_2$	$X3_4$	Y_5

Table 3-5. Example of SHAP importance scores and hits per feature.

Feature	X1	X2	X3	Shad. X1	Shad. X2	Shad. X3
SHAP Importance Score	0.37	0.13	0.17	0.12	0.16 (max)	0.05
Hit (This iteration)	1	0	1	N/A	N/A	N/A

This procedure is repeated across several iterations to account for different random permutations in the shadow features [82]. After *n* iterations, the experiment is modelled with a binomial distribution with $p = 0.5$ [74]. BorutaShap then performs a two-sided equality test based on this distribution to determine a feature’s importance from the number of hits it accumulated [74]. This iterative process enhances the robustness of the feature selection method by ensuring that only the features consistently deemed important across multiple iterations are retained.

By default, BorutaShap uses a Random Forest as the base regression model from which the importance scores are calculated. However, extending the feature space through shadow features may incorporate noise and result in overfitting during feature selection [83]. Since overfit models exhibit little bias and high variance [26], minor data changes may result in different important features [83]. Hence, a regressor exhibiting low variance is desired for enhanced robustness in feature selection.

As mentioned in previous paragraphs, the default base regression model in BorutaShap is a Random Forest [84]. However, BorutaShap enables the setting of different base regression models. This research compared different base regression models such as XGB

[85], CatBoost [86] and Extra-Trees [87] to find the model with the lowest variance in the selected features. The hyperparameters of each base regression model were not tuned in this analysis. Each base regression model was run considering different seed values [22] as the importance score is calculated on the test set, which results from a split made at random [74]. As seen in Table 3-6 and Table 3-7, for each seed, the number of important features identified by each model can differ. Additionally, the number of iterations required to achieve convergence varies with different seeds.

To enhance the robustness of feature selection, a stopping criterion was established in the BorutaShap analysis. This criterion required that the set of accepted features must not be zero. Additionally, it required that the number of accepted features must either remain the same in the last two runs or that the number of tentative features be zero, whichever occurred first.

BorutaShap was run multiple times for each base regression model and seed combination. Each run had a different number of iterations, starting with “200” iterations for the first run. For successive runs, the number of iterations was doubled if the stopping criterion was not met. This iterative approach aimed to determine the number of iterations needed to stabilize the number of accepted features. Therefore, the number of iterations reported in Table 3-6 and Table 3-7 corresponds to the final run, where the number of accepted features remained constant over the last two runs.

To evaluate the variance of different base regression models, this research introduced the average Common Important Feature Ratio (CIFR), as shown in Equation 3-11. The CIFR of each base model assesses variance by measuring the proportion of accepted features preserved over different random seeds. The CIFR ranges between 0 (no features preserved) and 1 (all features preserved), with higher values indicating reduced variance. Table 3-6 and Table 3-7 presents the results of the BorutaShap sensitivity analysis on the stiffness and fatigue dataset. This research selected Extra Trees [87] as BorutaShap base regression model, as it indicated lower variance through a higher CIFR and better accuracy through a higher mean R² on the test set. Similarly, this research selected Random Forest [84] as the base regression model for the BorutaShap analysis on the fatigue dataset.

$$\mathrm{CIFR}_i = \frac{1}{n} \cdot \sum_{j=1}^{n} \frac{\#\ \mathrm{Common\ Important\ Features}_i}{\#\ \mathrm{Important\ features}_{i,j}} \qquad \text{Equation 3-11}$$

where:

CIFR$_i$ = Common Important Feature Ratio of base model *i*.

n: Number of seeds per base model.

# Common Important Features: Number of common important features over all seeds for base model *i*.

# Important features: Number of important features of base model *i* considering seed *j*

Table 3-6. Results of the BorutaShap sensitivity analysis for the stiffness dataset.

Base Model	Seed	Iterations	R^2 Test set	Important Features	Common Important Features	Mean R^2 Test set	CIFR
Random Forest [84]	0	1600	0.77	12	9	0.80	0.70
	7	6400	0.80	15			
	31	1600	0.84	12			
	42	1600	0.77	13			
XGB Regressor [85]	0	200	0.77	7	6	0.77	0.76
	7	400	0.68	8			
	31	400	0.86	8			
	42	400	0.76	9			
CatBoost Regressor [86]	0	1600	0.77	18	15	0.83	0.88
	7	400	0.88	17			
	31	400	0.84	16			
	42	1600	0.83	17			
Extra-Trees Regressor [87]	0	1600	0.82	21	21	0.85	0.98
	7	1600	0.86	21			
	31	1600	0.88	21			
	42	400	0.83	23			

Table 3-7. Results of the BorutaShap sensitivity analysis for the fatigue dataset.

Base Model	Seed	Iterations	R^2 Test set	Important Features	Common Important Features	Mean R^2 Test set	CIFR
Random Forest [84]	0	1600	0.77	11	10	0.68	0.89
	7	6400	0.56	12			
	31	1600	0.70	11			
	42	1600	0.68	11			
XGB Regressor [85]	0	200	0.59	5	4	0.60	0.73
	7	400	0.45	8			
	31	400	0.72	5			
	42	400	0.63	5			
CatBoost	0	1600	0.72	8	7	0.62	0.79

Base Model	Seed	Iterations	R^2 Test set	Important Features	Common Important Features	Mean R^2 Test set	CIFR
Regressor [86]	7	400	0.47	11			
	31	400	0.73	9			
	42	1600	0.58	8			
Extra-Trees Regressor [87]	0	1600	0.75	11	8	0.66	0.72
	7	1600	0.49	13			
	31	1600	0.70	10			
	42	400	0.73	11			

The introduced data and physics driven feature selection method enabled reducing the number of input features by filtering out unimportant features for model performance or physical model computation. However, this method does not account for the correlation between the selected features. By analysing the correlation between features, the number of input features can be further reduced. The following section introduces the correlation analysis method leveraged by this research.

3.3.2. Correlation Analysis

This research used Spearman's correlation rank (see Equation 3-12) to assess correlation between the remaining features after the physics and data-driven feature selection. This analysis aims to further reduce the feature space and preserve the assumptions required for infusing a physical model (see Section 3.1).

$$\rho = 1 - \frac{6 \cdot \sum d_i^2}{n \cdot (n^2 - 1)} \qquad \text{Equation 3-12}$$

where:

ρ : Spearman correlation coefficient.

$d_i = x_i \cdot y_i$

x_i , y_i : Observation i of variables x and y

n: Total number of observations

The Spearman correlation rank measures the strength of a monotonic relationship between two variables. A positive Spearman correlation coefficient indicates that variable y tends to increase as variable x increases [88]. A negative value indicates that variable y tends to decrease when variable x increases [88]. A correlation coefficient of zero indicates no tendency for y when x changes [88]. The Spearman correlation rank was selected because the NL-LAB datasets include continuous and categorical features, and relationships between variables are non-linear [39], [40]. It is noted that no feature required for computing the output of a physical model was removed after correlation analysis.

The correlation analysis is the last step in the feature selection process. The outcome of this process is a reduced set of input features to be used in PIAI model training. The following section presents the considerations for PIAI model training and optimization under the framework developed.

3.4. Model Training and Optimization

PIAI Model training in this research serves a twofold purpose. The PIAI model must generate appropriate mappings between the selected features and targets and approximate the behaviour dictated by the chosen physical model. Deng et al. [15] suggests that neural networks can be leveraged to infuse physics in prediction models for tabular data in pavement engineering. Hence, the PIAI framework developed in this research uses neural networks as the base model architecture.

Since neural networks fit targets using a weighted sum of input variables, they require prior scaling of features and targets [70]. Scaling brings robustness and stability to model training by accounting for differences between measurement units of features [70]. Scaling is possible through normalization and standardization [70]. This research used normalization as the scaling method. Normalization rescales the data so that all values range between 0 and 1 [70] as shown in Equation 3-13. Normalization was preferred over standardization as it preserves the binary features obtained through one-hot encoding.

$$x_s = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \qquad \text{Equation 3-13}$$

where:

x_s: Scaled feature value.

x_i: Original feature value.

x_{max}: Maximum feature value.

x_{min}: Minimum feature value.

Figure 3-8 illustrates the developed PIAI framework, which contains a data component and a physics component. The data component aims to generate predictions ($\hat{y}_D$) that closely match the targets (y_D), as with conventional AI applications. The physics component strives to enhance the predictions by infusing physical knowledge from pavement engineering. This physical knowledge is infused within the PIAI framework through a physical model, from which the physics targets (y_P) are calculated. Subsequently, the physics component of the PIAI framework model is trained to generate predictions ($\hat{y}_P$) closely matching the physics targets.

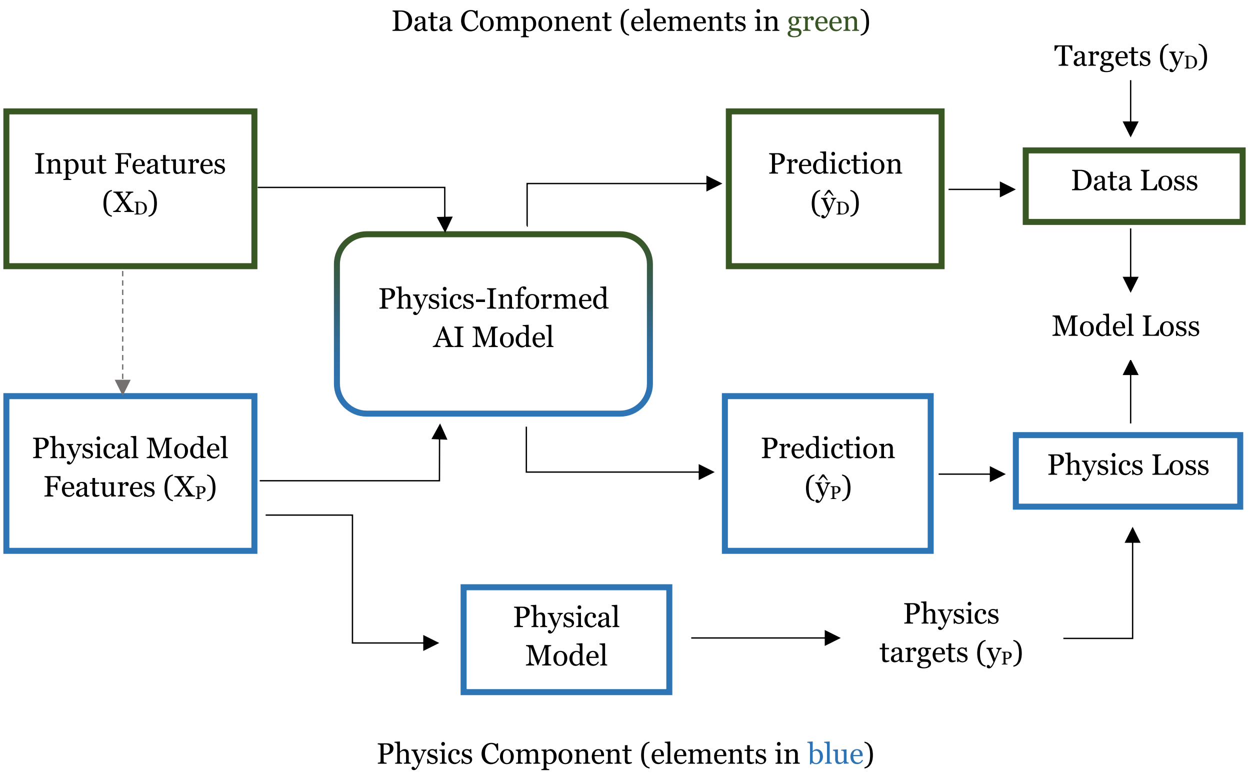
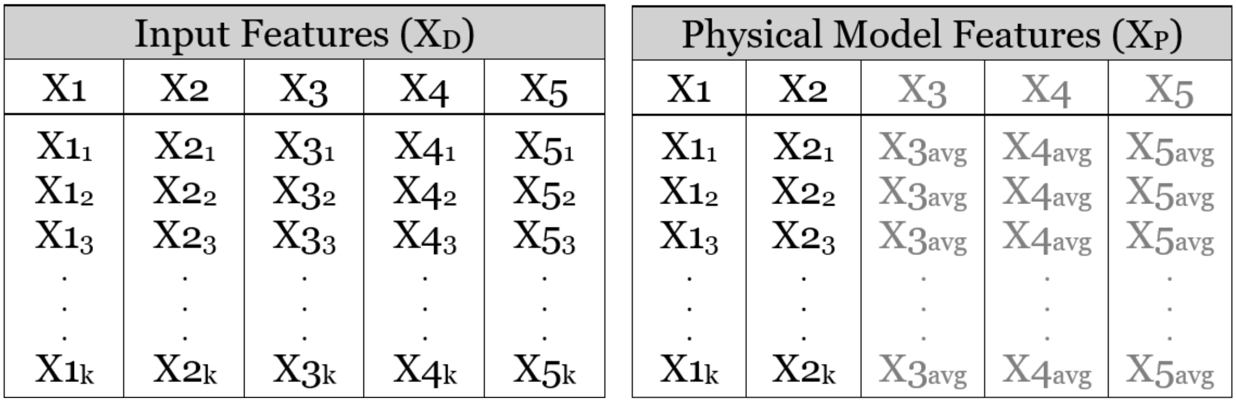


Figure 3-8. Developed PIAI framework.

The PIAI framework utilizes two feature sets: the Input Features (X_D) and the Physical Model Features (X_P). X_D and X_P have identical dimensions to ensure compatibility during PIAI model training [15]. Although X_P is derived from X_D, it is specifically tailored to enable training in the physics component. X_P preserves only the features needed to compute the physical model output and averages the remaining features [15]. While X_D leverages an extensive set of relevant features to enhance PIAI model accuracy, X_P facilitates the infusion of physical knowledge for enhanced PIAI model consistency and interpretability. Figure 3-9 illustrates an example of X_P generation using a toy dataset of 5 features. The example assumes that X1 and X2 are required to compute the output of a selected physical model. Consequently, X1 and X2 are preserved in X_P in the same order they had in X_D, while X3, X4, and X5 are averaged.



Input Features (X_D)				
X1	X2	X3	X4	X5
$X1_1$	$X2_1$	$X3_1$	$X4_1$	$X5_1$
$X1_2$	$X2_2$	$X3_2$	$X4_2$	$X5_2$
$X1_3$	$X2_3$	$X3_3$	$X4_3$	$X5_3$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
$X1_k$	$X2_k$	$X3_k$	$X4_k$	$X5_k$

Physical Model Features (X_P)				
X1	X2	X3	X4	X5
$X1_1$	$X2_1$	$X3_{avg}$	$X4_{avg}$	$X5_{avg}$
$X1_2$	$X2_2$	$X3_{avg}$	$X4_{avg}$	$X5_{avg}$
$X1_3$	$X2_3$	$X3_{avg}$	$X4_{avg}$	$X5_{avg}$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
$X1_k$	$X2_k$	$X3_{avg}$	$X4_{avg}$	$X5_{avg}$

Figure 3-9. Example of Physical model features definition

In the framework illustration shown in Figure 3-8, a model loss is introduced. This loss is described by Equation 3-14. Three elements compose the model loss: a data loss (L_D), a physics loss (L_P), and a physics tuning hyperparameter (λ). The physics tuning hyperparameter controls how much importance is given in the training process to the physics loss. A significantly low value of λ will result in a model with low physics compliance in its predictions, while a high value of λ will result in physics-compliant predictions that diverge from the observations (y_D). Furthermore, the researcher selected the mean square error (MSE) loss to calculate the data and physics losses. By selecting MSE, the implicit assumption is that the conditional distribution of the targets for a given input value is Gaussian [89]. Hence, the MSE loss is minimized when the model predicts the mean of the conditional target distribution regression problems [89].

$$L = L_D + \lambda \cdot L_P$$

Equation 3-14

where:

L_D: Data loss.

L_P: Physics loss.

λ: Physics tuning hyperparameter.

Equation 3-15

$$\text{MSE} = \frac{1}{N} \cdot \sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2$$

where:

y_i : Target.

$\hat{y}_i$: Prediction.

N : Number of instances.

Training the PIAI model consists in minimizing the model loss (see Equation 3-14). This research used the PyTorch [90] library for model training. Figure 3-10 shows a flow diagram of the training loop devised in this research. Defining a model class in PyTorch [90] involves determining the layer type, activation function, and neural network parameter initialization method. This research used Xavier's Uniform [91] initialization, which has been successfully leveraged in other PIAI applications [35]. Then, a model instance was created by specifying hyperparameters as the number of hidden layers, the number of nodes per hidden layer, and the dropout rate. After creating a model instance, an instance of the Adam [92] optimizer was also initialized. The optimizer finds a minimum loss after iterations through multiple epochs. An epoch refers to a single pass through the entire training dataset [26]. Each epoch involved making predictions, calculating the data and physics loss, computing loss gradients, performing backpropagation [26], and updating model parameters. Within the training loop, the researcher implemented an early stopping strategy to prevent overfitting. This strategy consisted in monitoring the decrease in the validation loss with every epoch. Hence, training was halted when the validation loss had not decreased for a given number of epochs [89]. Then, the model parameters yielding the lowest validation loss were kept as the optimal model parameters [89].

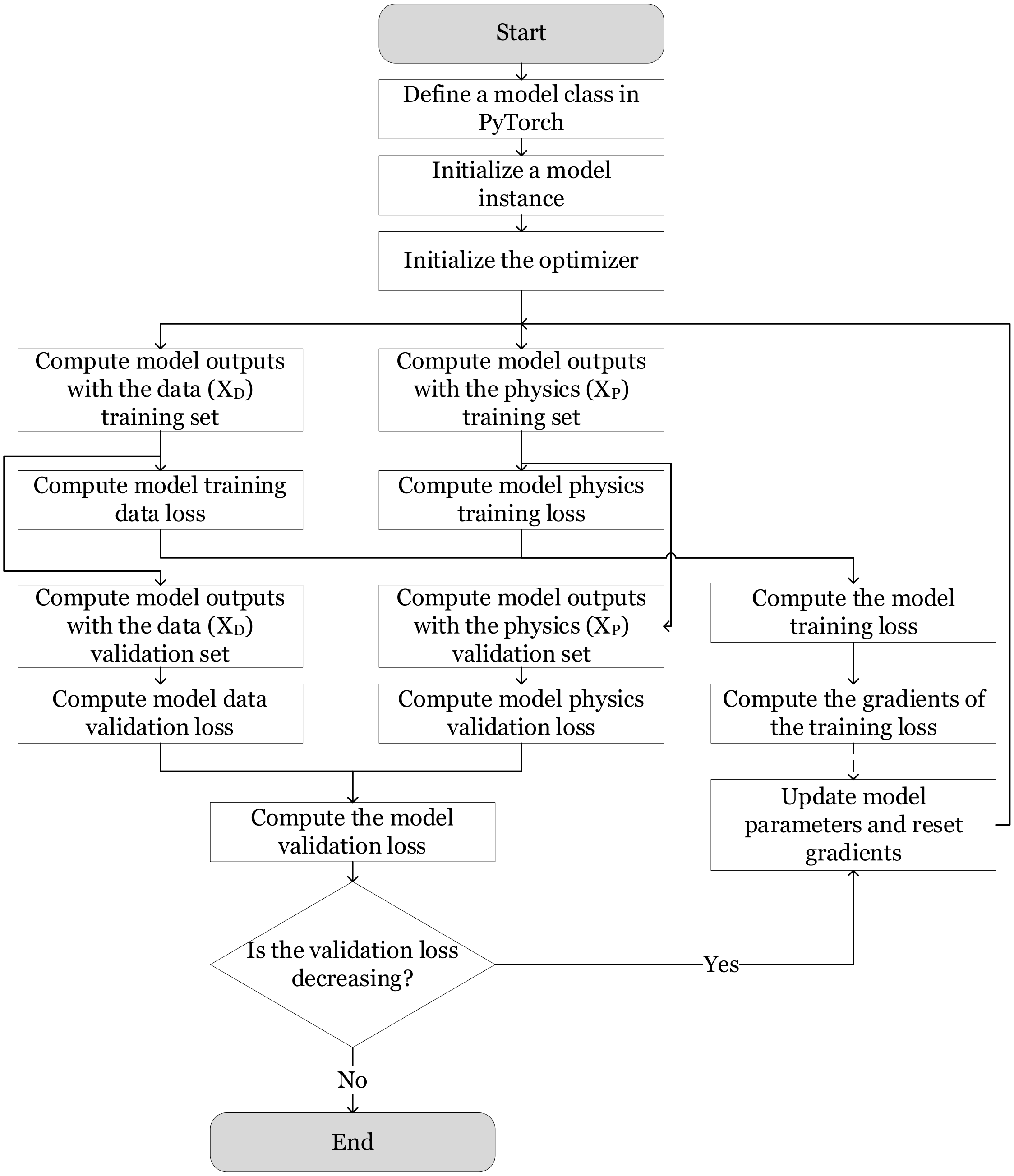

Start
Define a model class in PyTorch
Initialize a model instance
Initialize the optimizer
Compute model outputs with the data (XD) training set
Compute model outputs with the physics (XP) training set
Compute model training data loss
Compute model physics training loss
Compute model outputs with the data (XD) validation set
Compute model outputs with the physics (XP) validation set
Compute the model training loss
Compute model data validation loss
Compute model physics validation loss
Compute the gradients of the training loss
Compute the model validation loss
Update model parameters and reset gradients
Is the validation loss decreasing?
Yes
No
End

Figure 3-10. Training loop flow diagram.

The training process previously described does not allow for finding the optimal set of model parameters for different train-test-validation splits. To account for different train-validation-test splits during training, this research performed 10-fold cross-validation as shown in Figure 3-11. Cross-validation enables model training using a larger proportion of the data, aiming to reduce variance [26]. This method partitions the training and validation sets into *k (k = 10)* disjoint subsets or folds. For each fold, the model is trained with *k*-1 subsets and validated with the remaining subset [26]. The cross-validation error is the average validation error over all folds [20]. The researcher selected *k=10* as it has been empirically demonstrated to yield test error estimates without excessively high variance or high bias [20].

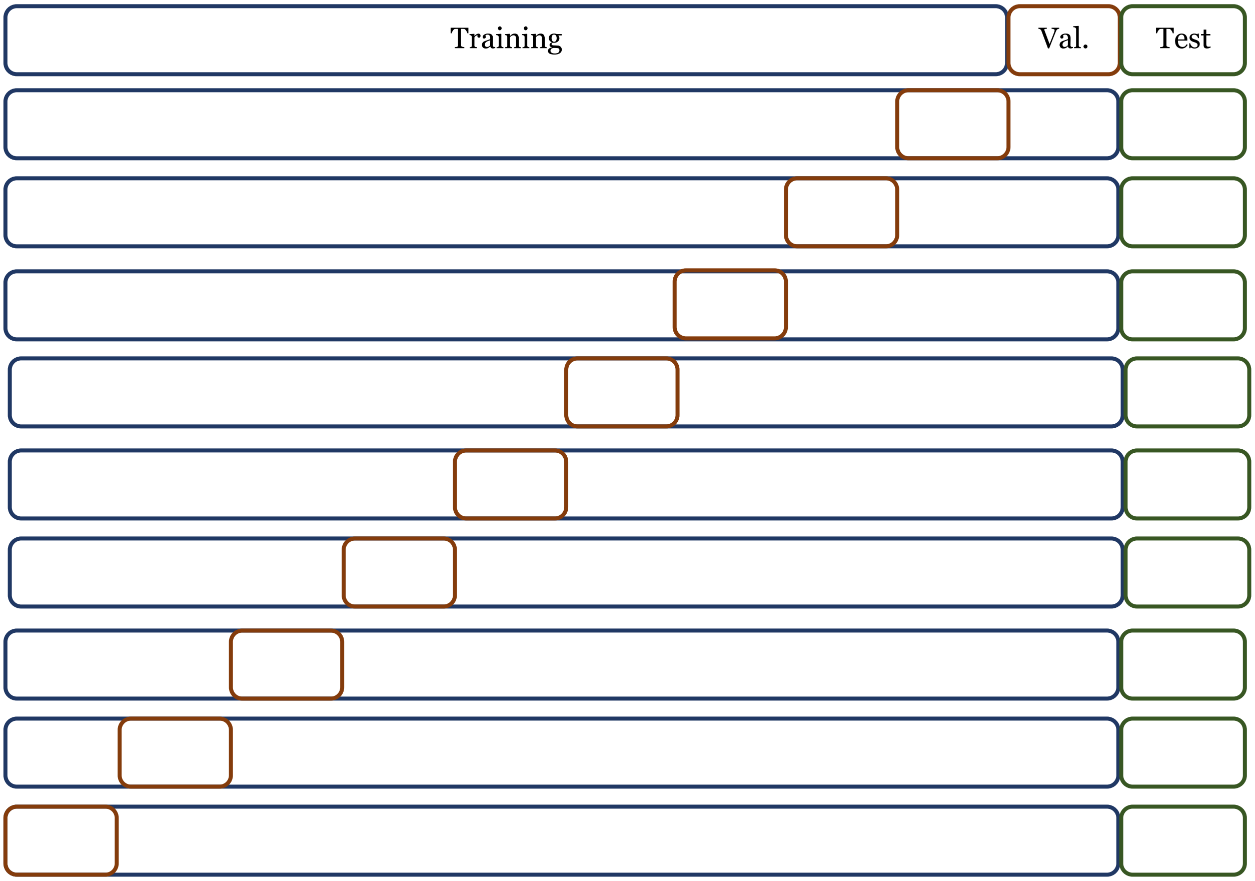


Figure 3-11. 10-fold cross validation.

Since model hyperparameters are fixed while executing 10-fold cross-validation, it is not possible to determine an optimal model architecture. This research leveraged hyperparameter optimization [93] to obtain the optimal model architecture. With hyperparameter optimization, it is possible to identify the set of hyperparameters that minimize the cross-validation loss [93]. The evaluated hyperparameters include the number of hidden layers and neurons per layer, the dropout rate, the learning rate [26] and the physics tuning hyperparameter (λ). Methods for hyperparameter optimization include Manual Search, Grid Search, Random Search, Gradient-based optimization, and Bayesian Optimization [93]. This research selected Bayesian Optimization as the hyperparameter tuning method. Bayesian Optimization leverages information from previous runs to hasten convergence to a local minimum, improving algorithm performance with respect to other hyperparameter tuning methods [93]. This research used the Optuna [94] package to perform Bayesian Optimization, for which Appendix E provides further details.

Table 3-8 presents the predefined range of variation for the model hyperparameters used in Bayesian Optimization. The ranges for the number of hidden layers, neurons per layer, dropout rate, and learning rate were selected based on common practice observed in the reviewed PIAI applications (see Table 2-3). The limits for the physics tuning hyperparameter (λ) vary by model and are based on the results from preliminary trials and subsequent graphical assessments of performance related to both data and physical constraints. The results from these trial runs are provided in Appendix F.

Table 3-8. Range of variation of model hyperparameters in Bayesian Optimization.

Hyperparameter	Model	Lower Limit	Upper Limit	Progression
No. of hidden layers	Both	2	4	In steps of 1
No. of neurons per layer	Both	4	128	By a factor of 2 (e.g., 4, 8, 16, …)
Dropout rate	Both	0	0.5	In steps of 0.1
Learning rate	Both	1x10^{-7}	1x10^{-1}	Logarithmic
Physics tuning hyperparameter (λ)	Stiffness	1x10^{-12}	1x10^{-6}	Logarithmic
	Fatigue	1x10^{-7}	1x10^{-1}	Logarithmic

This section concludes the chapter concerning the development of PIAI models for the NL-LAB stiffness and fatigue datasets. Through 10-fold cross-validation and Bayesian Optimization, the optimal set of model hyperparameters was found for each model. This optimal set included the value of the physics tuning hyperparameter that resulted in the lowest cross-validation loss. The following chapter presents and discusses the prediction results of the final PIAI fatigue and stiffness models.

4. Results and Discussion

This chapter presents the obtained PIAI models for stiffness and fatigue predictions based on the NL-LAB datasets. The results are presented in two sections corresponding with the last two stages of PIAI model development (see Figure 3-1). Section 4.1 introduces the results from the physics and data-driven feature selection analysis and the final features used for PIAI model training. Then, Section 4.2 introduces the results from model training and optimization for physics infusion. The multi-criteria assessment of these results provides an answer to the third research sub-question.

4.1. Feature Selection

The feature selection method introduced in this research comprises both physics and data-driven components. As delineated in Section 3.3, the physics component of feature selection automatically identified as important those features necessary for the computation of the physical model's output. Conversely, the important features from a data-driven perspective were those found to have a statistically significant impact in the model predictions. The data-driven component employed BorutaShap [74] as the feature importance analysis tool. The results of the BorutaShap feature importance analysis are presented through the normalized importance score (or Z-score) [74], obtained with SHAP values (see Appendix D). For each feature, a boxplot illustrates the median importance score, as well as the 25th percentile, the 75th percentile, and any outliers identified.

The importance score of each feature was compared with the maximum importance score of all shadow features to obtain the number of hits [74]. As mentioned in Section 3.3.1, an important feature is likely to have a higher number of hits than an unimportant feature. BorutaShap uses statistical testing with a binomial distribution to determine whether a feature is 'important', 'unimportant' or 'tentative' [74]. Important features have a statistically significant high number of hits [74]. Unimportant features scored significantly low hits and should not be included in predictive models [74]. Features labelled as 'tentative' are those for which a two-sided equality test yielded unconclusive results [74]. In BorutaShap boxplots, features are coloured in green, red, and blue. Features in green indicate features deemed as 'important', features in red were deemed as 'unimportant', and features in blue represent the importance score of the shadow features.

Figure 4-1 shows the importance scores obtained for the stiffness dataset. BorutaShap deemed 17 features as important and 5 features as unimportant. It is noted that two accepted features (“Volume_Target_Sand” and “Segment compactor”) reported similar median importance scores as four rejected features (“Volume_Target_Gravel”, “PMB”, “Shear box”, and “Field roller”). However, it is believed that the higher variability on the importance scores of the two accepted features yielded a higher number of hits that resulted in the acceptance verdict. Moreover, the important features from the physics component are also important features from a data-driven perspective (“Pa”, “Volume_Agg_Fraction”, “Volume_Target_bitumen”, “bit2_Gstar”).

The accepted features of the stiffness dataset were subjected to a correlation analysis using Spearman’s correlation rank. As shown in Figure 4-2, 7 features reported high (|ρ| > 0.8) correlation ranks. The features “Pa”, “Volume_Agg_Fraction”, and “bit2_Gstar” could not be removed due to their relevance to the physical model. Hence, the features “VA” and “densities” were removed as they were highly (|ρ| > 0.8) correlated with the features “Volume_Agg_Fraction”. Finally, the feature “bit2_TRenK” was removed as it showed high (|ρ| = 0.94) correlation with the feature “bit2_pen”. Appendix G presents further details on the results of the correlation analysis for the stiffness dataset.

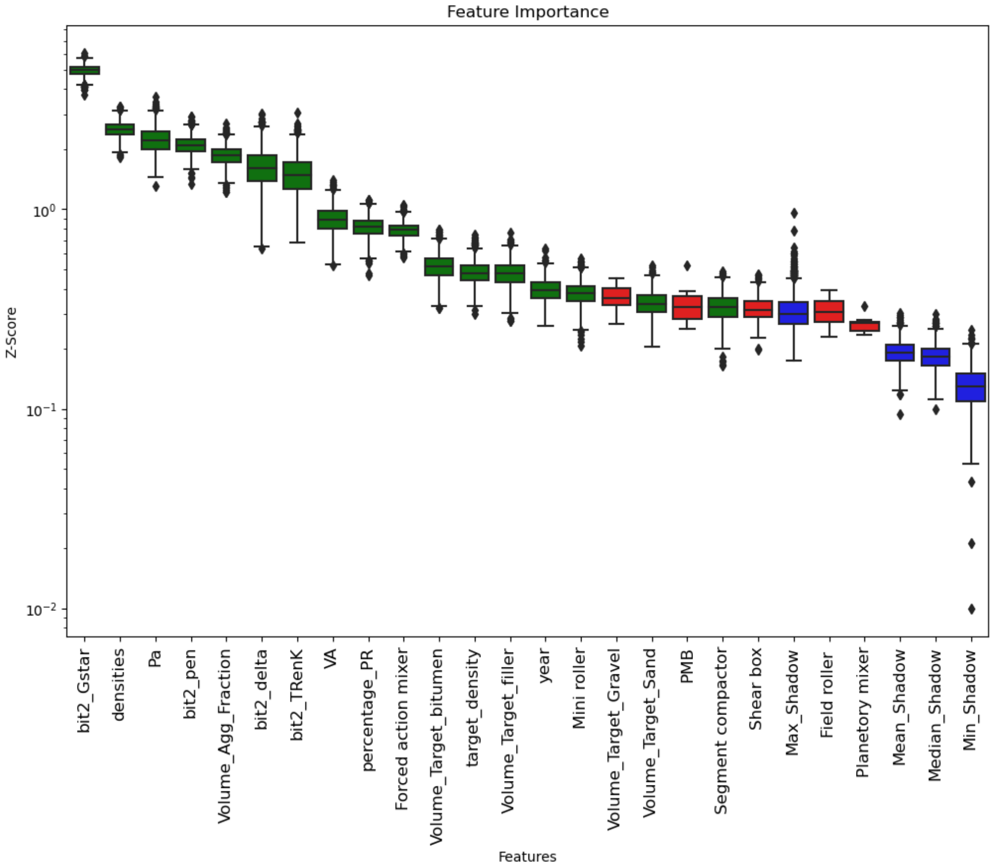


Figure 4-1. Data-driven feature importance results for the stiffness model.

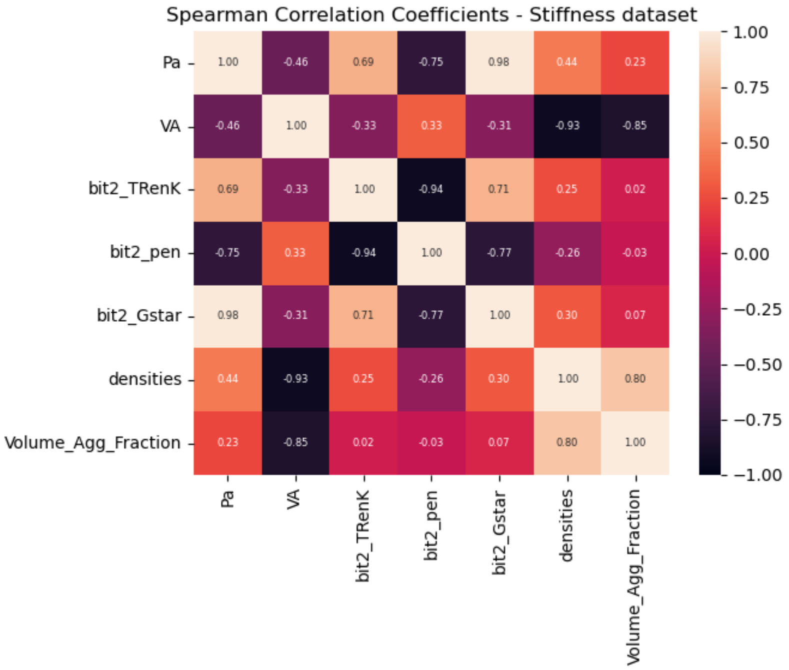


Figure 4-2. Correlation analysis for the stiffness model.

Figure 4-3 depicts the importance score for the fatigue dataset. BorutaShap deemed 12 features as important and 11 features as unimportant. Similar to the case of the stiffness model, two accepted features ("bit2_delta" and "percentage_PR") have similar median importance scores as two rejected features ("Planetory mixer" and "densities"). It is noted that two necessary features for computing the physical model output ("GP and "VP") were included in the set of accepted features. Although the feature "stiffness" was deemed as unimportant by BorutaShap, it was preserved given its importance for the physics component.

The accepted features of the fatigue dataset were subjected to a correlation analysis using Spearman's correlation rank. As shown in Figure 4-4, 2 features reported perfect monotonic increasing relationship (|ρ| = 1.00). Since the feature "VP" is relevant for computing the physical model output, the feature "VA" was removed from the dataset. Appendix G presents further details on the results of the correlation analysis for the fatigue dataset.

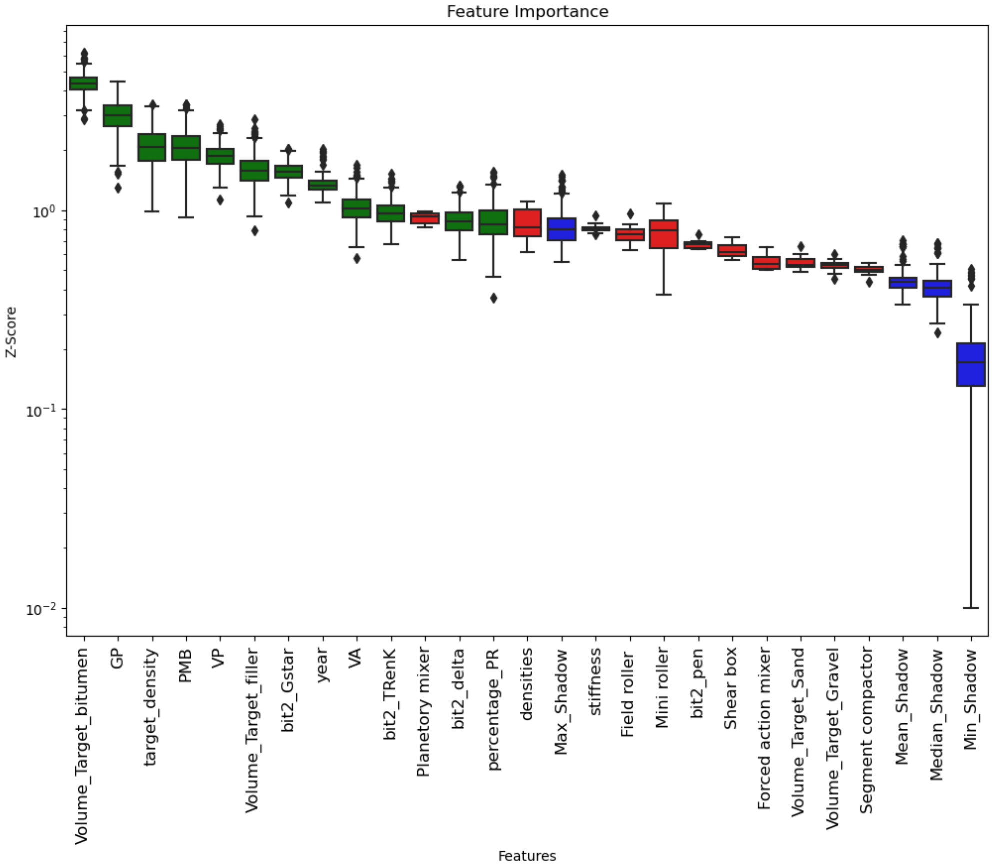


Figure 4-3. Data-driven feature importance results for the fatigue model.

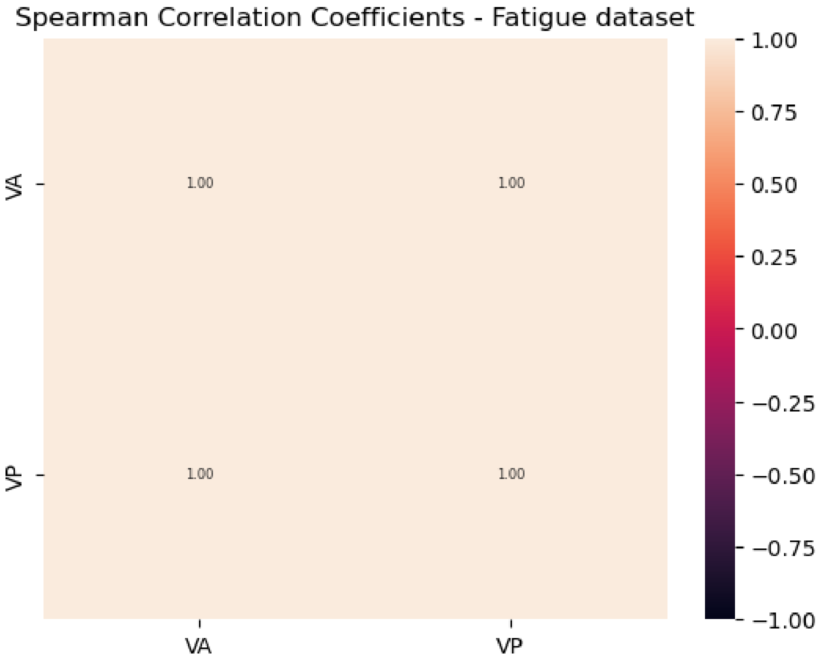


Figure 4-4. Correlation analysis for the fatigue model.

Table 4-1 outlines the final features incorporated into the PIAI stiffness and fatigue models. The inclusion of these features follows the results of the physics and data-driven components of feature selection. The PIAI stiffness model includes 14 features, of which 10 were identified as important by the data-driven component, and 4 were recognized as important by both the physics-based and data-driven components. Similarly, the PIAI fatigue model comprises 13 features, with 10 of them selected based on their importance for the data-driven component. Notably, 1 feature was deemed important exclusively by the physics component, while 2 features were considered important by both components.

Table 4-1. Relation of final features per model.

Model	Selected Features
Stiffness	Data component: "bit2_pen", "bit2_delta", "target_density", "percentage_PR", "Volume_Target_filler", "Volume_Target_Sand", "year", "Forced action mixer", "Mini roller", "Segment compactor". Data and physics component: "bit2_Gstar", "Volume_Target_bitumen", "Volume_Agg_Fraction", "Pa".
Fatigue	Data component: "bit2_TRenK", "bit2_Gstar", "bit2_delta", "target_density", "percentage_PR", "Volume_Target_filler", "Volume_Target_bitumen", "year", "PMB". Physics component: "stiffness". Data and physics component: "GP", "VP".

The final set of features was employed for the training and optimization of the PIAI stiffness and fatigue models. As outlined in Section 3.4, the optimal PIAI model architectures were determined through Bayesian Optimization. These optimized architectures were subsequently used to train the final PIAI stiffness and fatigue models. The following section presents the final PIAI stiffness and fatigue models along with a multi-criteria performance evaluation.

4.2. PIAI models for fatigue and stiffness predictions

This section presents the results obtained from applying the PIAI framework developed to generate prediction models for the NL-LAB fatigue and stiffness datasets. This research aimed to unveil the effects of physics infusion on PIAI model performance. The selected performance criteria were accuracy, interpretability, and consistency. The definitions and metrics used for each criterion are explained below.

Accuracy refers to the extent to which the PIAI model predictions ($\hat{y}_D$) align with the dataset targets (y_D) [20]. This research selected three accuracy metrics: the R^2 value, Root

Mean Square Error (RMSE), and Mean Average Percentage Error (MAPE). The R² value (see Equation 4-1) quantifies the proportion of the target variance explained by the model features [95]. It ranges between -∞ and +1, with the upper limit indicating a perfect fit between predictions and targets [95]. Negative R² values suggest the model performs worse than a baseline model represented by a horizontal line at the target mean (ȳ) [95].

The RMSE (see Equation 4-2) indicates the average distance between model predictions and targets [95]. An RMSE value of 0 indicates a perfect fit, while increasingly higher values indicate an increasing discrepancy between predictions and targets [95]. The MAPE (see Equation 4-3) also measures the distance between predictions and targets, but it does so in relative terms [95]. Like the RMSE, a MAPE of 0 indicates a perfect fit while higher values indicate a worse model fit [95]. Given that R² is generally more informative for evaluating the quality of a regression model [95], this research adopts the interpretation scale shown in

Table 4-2 [39]. It is also noted that the accuracy metrics were calculated by comparing PIAI model predictions ($\hat{y}_D$) with the targets (y_D) in the data-driven component (see Figure 3-8).

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\bar{y} - y_i)^2} \qquad \text{Equation 4-1}$$

where:

$\hat{y}_i$: Prediction of instance *i*.
y_i: Target of instance *i*.
$\bar{y}$: Target mean over all instances
n: Number of instances

$$RMSE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \qquad \text{Equation 4-2}$$

$\hat{y}_i$, y_i, and *n* were explained in Equation 4-1.

$$MAPE = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \qquad \text{Equation 4-3}$$

$\hat{y}_i$, y_i, and n were explained in Equation 4-1.

Table 4-2. Proposed interpretation scale for model accuracy and interpretability ([39]).

Accuracy interpretation	Lower R^2 value	Upper R^2 value
Poor	-∞	0.65
Moderate	0.65	0.75
Good	0.75	0.85
High	0.85	0.95
Very High	0.95	1.00

While accuracy metrics provide insights on the predictive power of a developed AI model, an additional analysis is required to assess interpretability. An interpretable AI model should follow a set of physical constraints to improve the understanding on the predictions generated [21]. The physical model infused in the PIAI models can be understood as a constraint on the effect of a subset of features in model predictions. Hence, a user familiar with the physical model infused can make better informed decisions on whether to trust the PIAI model predictions [21]. This research assessed interpretability as the extent to which model predictions ($\hat{y}_P$) in the physics component satisfy the soft constraint imposed by the physical model (y_P). It was opted to assess interpretability by comparing y_P and $\hat{y}_P$ graphically and without incorporating any metrics. Although the abovementioned accuracy metrics could also quantify interpretability, they were purposefully omitted to maintain clarity on the conceptual difference between accuracy and interpretability.

The accuracy criterion was extended to account for different model initializations with the consistency assessment. Since model parameters are initialized randomly using Xavier's Uniform distribution [91], the final parameters are expected to vary with different random seeds [22] after training. This research assessed consistency as the extent to which the average prediction over m ($m = 10$) different random seeds approximate the infused physical model. The effect of physics infusion on the variability among different predictions in the physics component was also assessed using the average coefficient of variation (see Equation 4-4). A lower average coefficient of variation is indicative of higher consistency on the predictions in the physics component [15]. Hence, the consistency assessment was performed with and without considering physics infusion in the model.

$$CV_{avg} = \frac{1}{m} \cdot \sum_{i=1}^{m} \frac{\sigma_i}{\mu_i} \qquad \text{Equation 4-4}$$

where:

CV_{avg}: Average coefficient of variation.

σ_i: Standard deviation of the prediction vector of seed *i*.

μ_i: Mean of the prediction vectors corresponding to seed *i*.

m: Number of random seeds evaluated.

This section provided detailed definitions, metrics and scales for three selected performance criteria: accuracy, interpretability, and consistency. The following subsections present the results from applying the PIAI framework to generate prediction models for the NL-LAB stiffness and fatigue datasets. The upcoming subsection bases on the preset criteria to assess the performance of the PIAI stiffness model.

4.2.1. PIAI model for stiffness prediction infused with micromechanics

Table 4-3 shows the architecture of the PIAI model for stiffness prediction infused with micromechanics. This architecture was obtained via Bayesian optimization by minimizing the cross-validation loss. As mentioned before (see Equation 3-14), the model loss is the sum of the data loss, and the physics loss weighted by the physics tuning hyperparameter (λ). The optimal λ value for the PIAI stiffness model was determined to be $3.80\text{x}10^{-10}$. However, the hyperparameter importance analysis performed in Optuna [94] deemed λ to be the least influent hyperparameter on the cross-validation loss, as shown in Figure 4-5. Therefore, it was decided to further analyse the effects of λ in prediction accuracy.

Table 4-3. Model architecture – PIAI stiffness model.

Hidden Layers	**Nodes per Layer**	**Activation Function**	**Dropout rate**	**Physics tuning parameter (λ)**	**Learning rate**
3	32	ReLU (linear activation in outer layer)	0	$3.80\text{x}10^{-10}$	$1.93\text{x}10^{-5}$

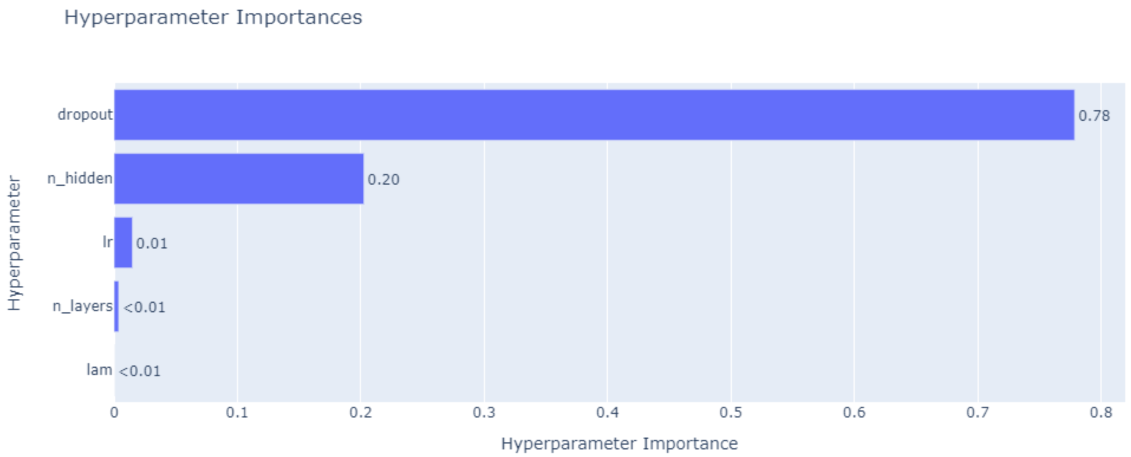


Figure 4-5. Hyperparameter Importances – PIAI stiffness model.

To further explore the low importance score of λ, this research conducted a sensitivity analysis. This analysis aimed to further understand the influence of λ in the model loss by training multiple models with a fixed optimal architecture and different λ values. For each model, the data and physics components of the model loss were calculated for the training, validation, and test sets. Although the losses of the data and physics component were calculated using the MSE, the RMSE was preferred for this analysis due to its consistency with the model target units. Figure 4-6 shows the variation of the RMSE with respect to λ for the data and physics components of the PIAI stiffness model. This figure unveiled the existence of a physics infusion region between λ = 1x10^{-11} and λ = 1x10^{-8}, encompassing the optimal λ value. In this region, the losses of the data and physics component are balanced. When λ is lower than 1x10^{-11} a low model loss is achieved in the data component at the expense of a physics agnostic model. Similarly, when λ is higher than 1x10^{-8}, the PIAI model adheres to the underlying physical model but fails to predict the model targets. Notably, within the physics infusion region, small changes in λ have a small effect on the model loss. Hence, the low λ influence is likely related to its low importance score, which may be attributed to an early narrowing of the search space in Bayesian Optimization.

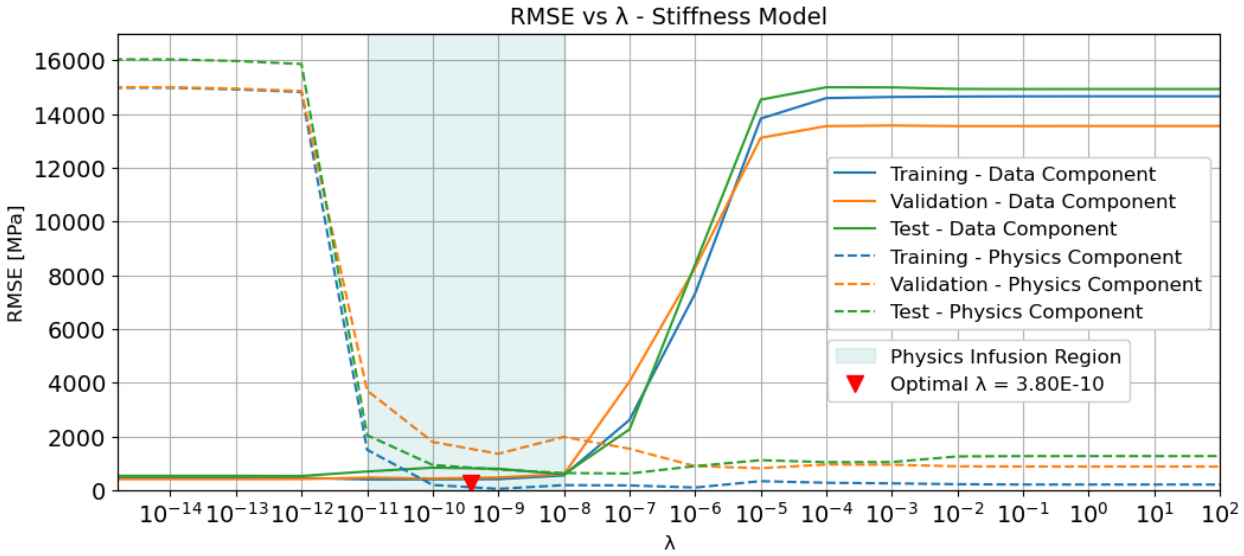


Figure 4-6. λ vs RMSE sensitivity analysis – PIAI stiffness model.

Table 4-4 and Figure 4-7 present the accuracy metrics and visualize the data fit of the PIAI stiffness model across different datasets. The model demonstrates a very high performance on the training set with an R² of 0.96 and a low RMSE of 418.18 MPa, effectively capturing the variance present in the target. These findings are further supported by the results on the validation set. The model maintains a high performance with an R² of 0.94 and an RMSE of 432.60 MPa, showcasing the model ability to fit unseen data during the validation phase. The model continues to exhibit good performance on the test set. In this set, the model reported an R² of 0.82 and a RMSE of 672.30 MPa. While there is some increase in the error metrics, the model captures more than 80% of the target variance with completely unseen data. Moreover, the MAPE exhibits a consistent pattern across the datasets, ranging from 3.21% to 5.17%. These results reflect the model stable predictive power, highlighting its reliability for stiffness prediction in asphalt mixes.

Table 4-4. Accuracy metrics of the PIAI stiffness model.

Set	R²	RMSE [MPa]	MSE	MAPE [%]
Training	0.96	418.18	174,873.16	3.21
Validation	0.94	432.60	187,144.06	3.78
Test	0.82	672.30	451,987.75	5.17

Interpretability of the PIAI stiffness model was assessed by the extent to which the model adhered to the physical constraint imposed by Zhang et al.'s [58] model. Figure 4-8 compares the predictions of the physics component ($\hat{y}_P$) with the targets given by the physical model (y_P). A visual assessment of this figure showcases the ability of the PIAI stiffness model to adhere to the infused physical model. The physical model is approximated in an impressive way for the training and test set, and to a slightly lower but still good extent for the validation set. These results validate the developed PIAI framework as a promising approach to infuse physics in AI predictive models for asphalt mix stiffness.

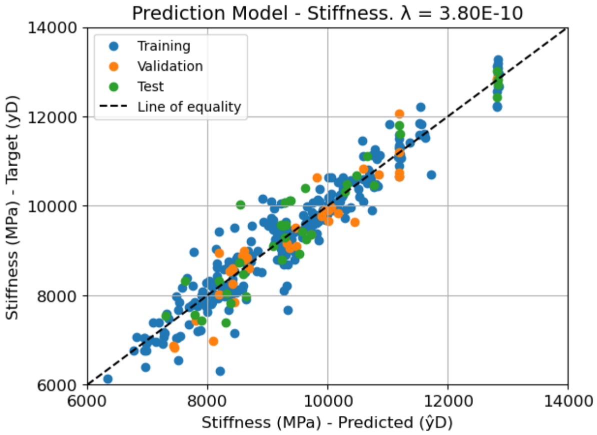


Figure 4-7. Predictions vs. Targets – PIAI stiffness model (Data component).

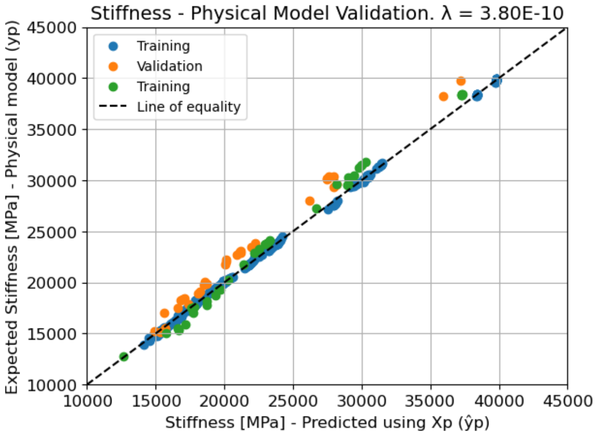


Figure 4-8. Predictions vs. Targets – PIAI stiffness model (Physics component).

Figure 4-9 and Figure 4-10 illustrate the results of the consistency assessment for the PIAI stiffness model. Figure 4-9 shows the mean and 95% confidence band of the predictions in the physics component ($\hat{y}_P$) when physics infusion is not considered in the PIAI model. As shown, the mean prediction in the physics component situates between 7,500 MPa and 12,500 MPa, without approximating the infused physical model. Moreover, the average coefficient of variation between predictions in the physics component was 6.49%. Figure 4-10 shows the results obtained when infusing physics in the PIAI model. The mean prediction closely approximates the infused physics for values lower than 25,000 MPa. Although model predictions over 25,000 MPa slightly underestimate the physical model targets, the physics approximation is also remarkable in this region. Moreover, the average coefficient of variation was reduced by an impressive 56% when physics was infused in the prediction model.

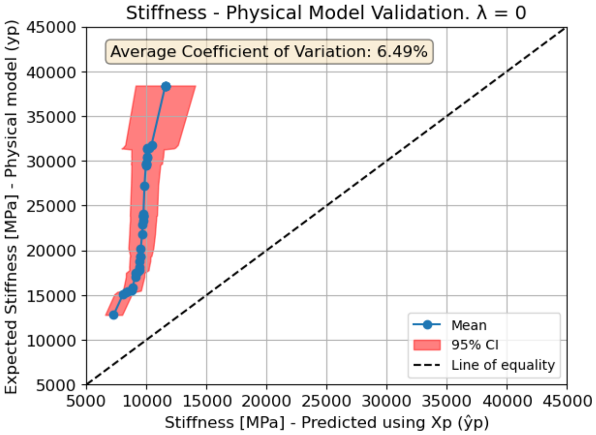


Figure 4-9. Consistency assessment – PIAI stiffness model (without physics).

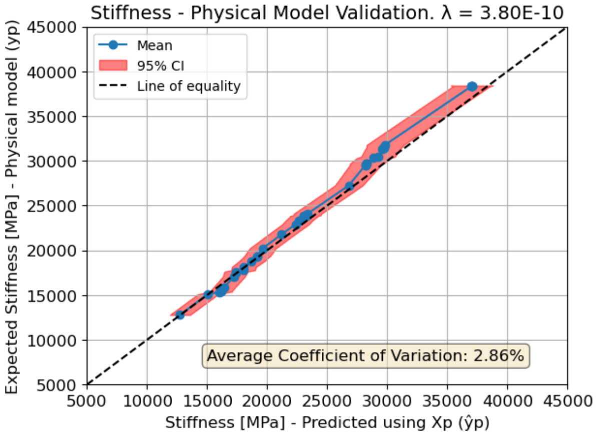


Figure 4-10. Consistency assessment – PIAI stiffness model (with physics).

The results obtained for the PIAI stiffness prediction model were satisfactory and promising regarding accuracy, interpretability and consistency. These results validate the effectiveness of the PIAI framework to infuse Zhang et al.'s [58] model in AI predictive models for asphalt mix stiffness. The following subsection introduces the results obtained after applying this framework to generate a PIAI model for asphalt fatigue prediction infused with energy dissipation theory.

4.2.2. PIAI model for fatigue prediction infused with energy dissipation theory

Table 4-5 shows the hyperparameters of the PIAI fatigue prediction model infused with energy dissipation theory. The hyperparameters correspond to the model architecture that achieved the lowest cross-validation loss in Bayesian Optimization. The optimal λ value for the PIAI fatigue model is 8.50x10^{-6}. Like in the PIAI stiffness model, λ was found by Optuna [94] to be the least influent hyperparameter on the cross-validation loss, as shown in Figure 4-11.

Table 4-5. Model architecture – PIAI fatigue model.

Hidden Layers	Nodes per Layer	Activation Function	Dropout rate	Physics tuning parameter (λ)	Learning rate
3	128	ReLU (linear activation in outer layer)	0	8.50x10^{-6}	1.21x10^{-5}

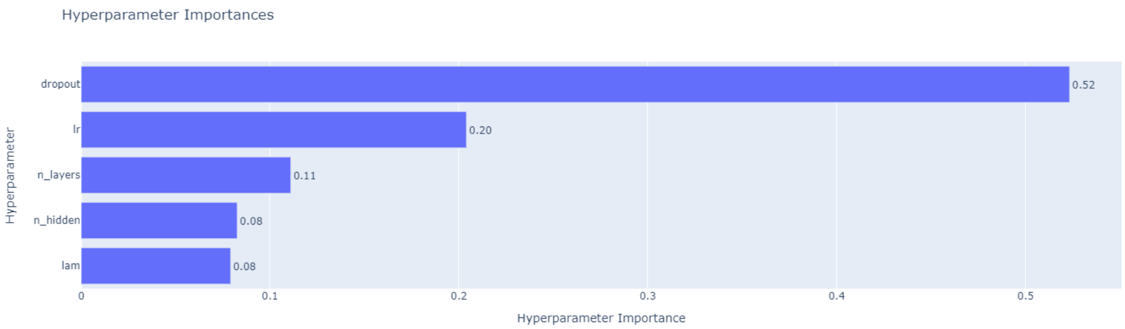


Figure 4-11. Hyperparameter Importances – PIAI fatigue model.

Given the low importance score of λ, a sensitivity analysis was conducted to further investigate its impact on the PIAI fatigue model's loss. The sensitivity analysis followed the same procedure described for the PIAI stiffness model. The results of the sensitivity analysis for the fatigue model are shown in Figure 4-12. The variation in the RMSE with respect to λ for the data and physics components unveiled a physics infusion region for the PIAI fatigue model. This region lies between λ = 1x10^{-6} and λ = 1x10^{-3} and comprises the optimal λ value. Within this range, both the data and physics component losses remain low. When λ is lower than 1x10^{-6}, the data component loss is small, but the model does not approximate the infused physics. Similarly, when λ is higher than 1x10^{-3}, the PIAI model adheres to the underlying physical model sacrificing prediction accuracy. Within the physics infusion region, small changes in λ led to minimal changes in the model data and physical losses. Hence, as concluded for the PIAI stiffness model, the low importance score of λ is likely related to an early narrowing of the search space in Bayesian Optimization.

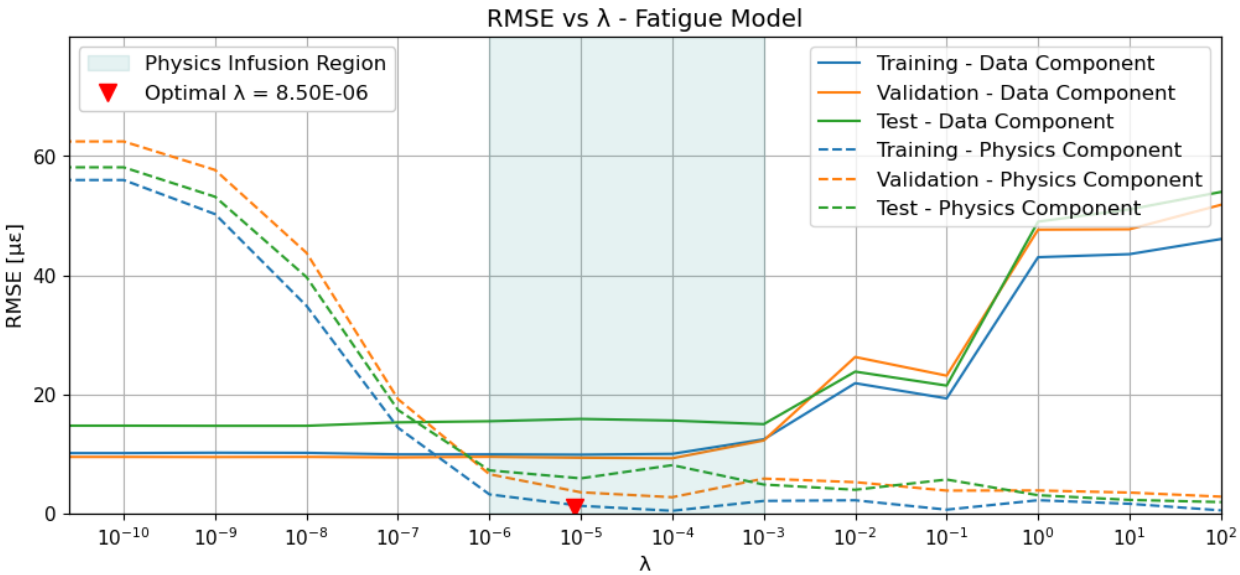


Figure 4-12. λ vs RMSE sensitivity analysis – PIAI fatigue model.

Table 4-6 and Figure 4-13 present a detailed overview of the accuracy metrics and visualize the data fit of the PIAI fatigue model across different datasets. The model demonstrated good performance on the training set, with an R² of 0.81 and an RMSE of 10.06 με, capturing a significant amount of the variance present in the target. These findings are further supported by the model performance on the validation set. On this set, the model exhibited an enhanced prediction accuracy with a reported R² of 0.82 and an RMSE of 9.53 με. However, when tested in unseen data, the prediction accuracy of the PIAI fatigue model decreases with an R² of 0.44 and an RMSE of 15.53 με. It is believed that this accuracy decrease may be related to the limited number of data instances and to the model architecture. The employed architecture is a neural network, which is simpler and more prone to overfitting compared to other alternatives like ensemble methods [96].

Table 4-6. Accuracy metrics of the PIAI fatigue model.

Set	R²	RMSE [με]	MSE	MAPE [%]
Training	0.81	10.06	101.22	6.26
Validation	0.82	9.53	90.82	7.53
Test	0.44	15.53	241.33	10.56

The interpretability of the PIAI fatigue model was assessed by the extent to which the model adhered to the physical constraint imposed by Shen & Carpenter's [66] model. Figure 4-14 compares the predictions of the physics component ($\hat{y}_P$) with the targets given by the physical model (y_P). A visual assessment of this figure showcases the ability of the PIAI fatigue model to adhere to the infused physical model. Despite the differences in performance reported for the data component, the physical model is approximated to a very good extent in all subsets. These findings reinforce the effectiveness of the developed PIAI framework in infusing physics in prediction models. The framework offers the potential to incorporate interpretability prediction models, even when an inferior accuracy is reported.

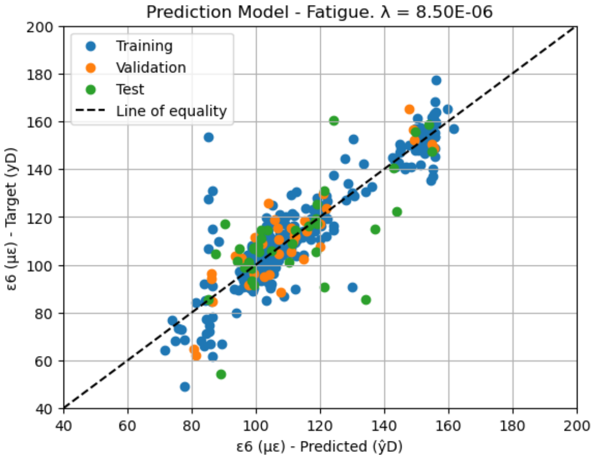


Figure 4-13. Predictions vs. Targets – PIAI fatigue model (Data component).

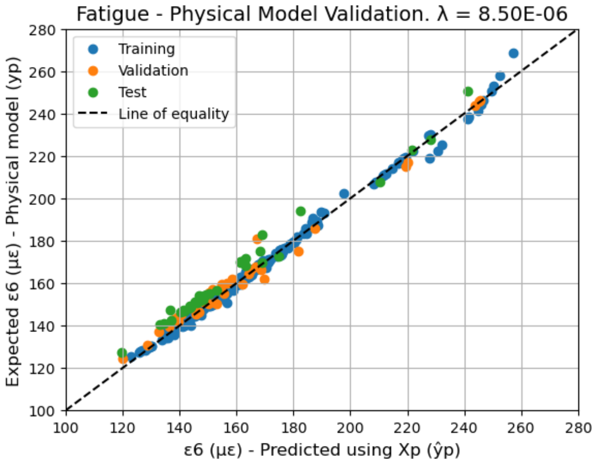


Figure 4-14. Predictions vs. Targets – PIAI fatigue model (Physics component).

Figure 4-15 and Figure 4-16 illustrate the results of the consistency assessment for the PIAI fatigue model. Figure 4-15 shows the mean and 95% confidence band of the predictions in the physics component (ŷP) when physics infusion is not considered in the PIAI model. The figure shows that the mean of the predictions in the physics component does not approximate the physical model in any region. Additionally, the average coefficient of variation between predictions in the physics component was 29.94%, with higher predictions exhibiting wider confidence bands. Figure 4-16 shows the results obtained when infusing physics in the PIAI model. The mean prediction closely approximates the infused physics for all the considered values. Moreover, the average coefficient of variation was reduced by a remarkable 88%, demonstrating the substantial impact of physics infusion in model consistency.

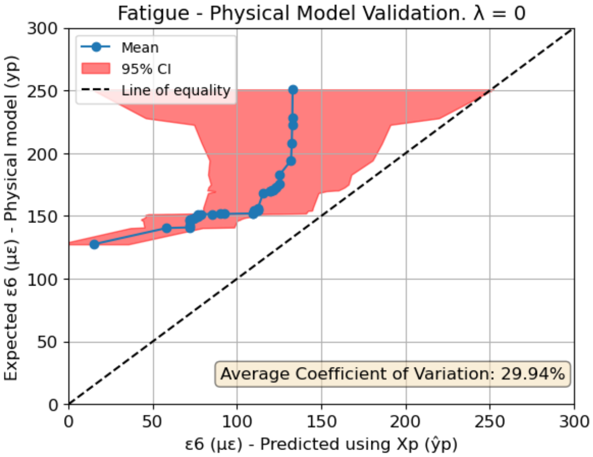


Figure 4-15. Consistency assessment – PIAI fatigue model (without physics).

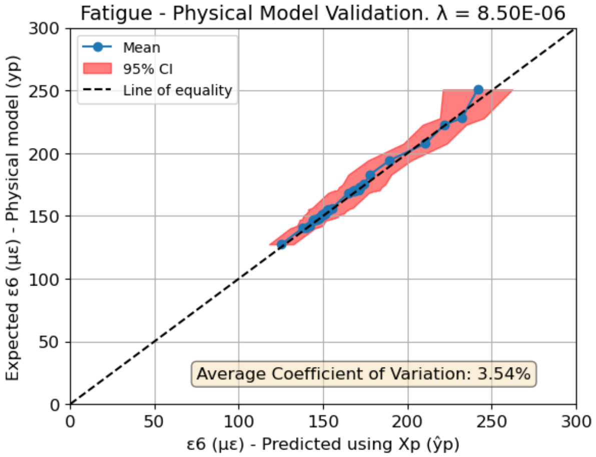


Figure 4-16. Consistency assessment – PIAI fatigue model (with physics).

The results obtained for the PIAI fatigue prediction model were satisfactory and promising regarding interpretability and consistency. These results validate the effectiveness and the PIAI framework to infuse Shen & Carpenter's [66] model in AI asphalt fatigue predictions. However, future research is needed to improve the accuracy of the PIAI fatigue model [66].

This Chapter concludes with the presentation and discussion of the prediction results of PIAI models developed for the NL-LAB fatigue and stiffness datasets. Based on the outcomes obtained, and the insights gained during framework development, the following Chapter presents the conclusions, answers to research questions, and recommendations for future research.

5. Conclusions and future recommendations

This Chapter provides the conclusions of this research and recommendations for future research work in three distinct sections. Section 5.1 provides general conclusions and introduce the main findings and societal relevance of this research. Then, Section 5.2 presents the answers to the research questions posed in Section 1.3. Finally, Section 5.3 provides future recommendations regarding data collection, testing, and PIAI model development.

5.1. General Conclusions

This research successfully developed a Physics-Informed Artificial Intelligence (PIAI) framework to infuse physical models in AI predictive models for pavement engineering. The PIAI framework offers a systematic approach to enhancing AI prediction models with physics for pavement performance predictions. Two prediction models were developed for the NL-LAB [6] fatigue and stiffness datasets. The PIAI fatigue model infused energy dissipation theory through Shen & Carpenter's [66] model. Similarly, the PIAI stiffness model infused micromechanics and homogenization theory through Zhang et al.'s [58] model.

An important finding of this research is that physical models seldom encompass all features and targets of the NL-LAB datasets. This challenge was addressed in the proposed PIAI framework with the implementation of several alignment controls. This research introduced, based on the dataset characteristics, a set of compatibility requirements for a candidate physical model to infuse (see Section 3.1). Additionally, the physics component introduced during feature selection (see Section 3.3) ensured preserving the necessary features for computing the physical model regardless of the data-driven component output. Furthermore, to ensure adequate physics infusion during PIAI model training, the features required to evaluate the physical models were isolated while preserving model dimensionality (see Section 3.4).

The developed PIAI framework effectively integrates physics into tabular data without spatial or temporal correlations, infusing physical models based on linear combinations of functions of features in the datasets. Hence, the application scope is limited to predictive tasks matching these conditions. When data has spatial or temporal correlations, it is recommended to modify the model architecture and loss function accordingly. Moreover, when the physical model to infuse is in differential form, it is recommended to modify the model loss to match that of a PINN (see Section 2.1.1).

The societal relevance of this research is underscored by its potential to foster greater acceptance and trust in AI within the pavement engineering community. By infusing well-researched physical models, PIAI can help demystify AI's "black box" perception while exploiting its ability to infuse new models. Moreover, PIAI can potentially make pavement performance assessments more expedite by reducing the need for extensive material testing. Remarkably, the PIAI framework can infuse new developments in pavement engineering concerning recycled and biobased materials, driving sustainability and innovation in the field.

5.2. Answers to Research Questions

This section provides an answer to the research questions posed in Section 1.3. The main research question is answered as follows:

Q: Can physics be infused to enhance AI prediction models for stiffness and fatigue of asphalt mixtures?

A: Yes, physics was successfully infused into AI models for stiffness and fatigue prediction in asphalt mixtures. The developed PIAI models satisfactorily approximate the infused physical model to enhance interpretability and consistency without compromising prediction accuracy. An analysis on the effects of physics infusion on prediction accuracy revealed the existence of a "physics infusion region". In this region the model losses of the data and physics component are both low, with small changes in the physics tuning hyperparameter (λ) resulting in small changes on both losses. For the PIAI stiffness model the physics infusion region lies between $\lambda = 1x10^{-11}$ and $\lambda = 1x10^{-8}$ (see Figure 4-6. λ vs RMSE sensitivity analysis – PIAI stiffness model.) whereas in the PIAI fatigue model this region lies between $\lambda = 1x10^{-6}$ and $\lambda = 1x10^{-3}$ (see Figure 4-12). Notably, the optimal value of the physics tuning hyperparameter was found in the physics infusion region in both models.

Q: Which method for infusing physics in AI models better suits the characteristics of the NL-LAB fatigue and stiffness datasets?

A: The selected method for physics infusion in prediction models for the NL-LAB fatigue and stiffness datasets was a Physics-Informed Loss Function. The main reason behind this selection is twofold. Candidate models to infuse often contain a subset of all features in the datasets. Hence, physics infusion aimed to constrain the effect of this subset of features without removing the impacts of additional features in the predictions. Previous PIAI applications [15] in PIAI pavement engineering successfully constrained the effects of a subset of features via a Physics-Informed Loss Function. Furthermore, instances in NL-LAB datasets were uncorrelated in time or space, limiting the applicability of a Physics-Informed Neural Network or a Physics-Informed Architecture.

Q: What conditions are required to infuse a physical model into a PIAI prediction model for the NL-LAB fatigue and stiffness datasets?

A: This research established criteria to ensure the infusion of a given physical model in AI predictive models. A physical model can only be infused in AI predictive models if its dependent and independent variables are part of the dataset features or were calculated from them. Besides, the dataset target must be included in the physical model, either explicitly or through a traceable calculation. Moreover, the testing conditions of the dataset features should match those used to obtain the physical model. It is possible to infuse physics without meeting this requirement. However, doing so without accounting for differences in testing conditions can lead to inconsistencies. Finally, the researcher recommends incorporating previously validated developments as physical models to minimize the risk of infusing flawed physical behaviour.

Q: How does infusing physics impact model accuracy, interpretability, and consistency?

A: Table 5-1 presents the accuracy metrics obtained by the PIAI models developed in this research compared to previous research in the same datasets [40], [39]. It is noted that Martini [39] did not develop a predictive model for fatigue prediction. The developed PIAI fatigue and stiffness models underperformed compared to previous research efforts. The researcher believes that the PIAI model architecture explains the lower accuracy of the obtained model. Prior research used gradient-boosting decision trees, which leverage a more efficient strategy for overfitting than the train-validation split [86]. In addition, this research used 10% of the data for testing, which may have been insufficient to ensure similar distributions in the training, validation, and test sets.

Table 5-1. Comparison of model accuracy with previous research.

Dataset	Author	R^2 - Test set
Fatigue	Mota Lontra [40]	0.80
	This research	0.44

Dataset	Author	R^2 - Test set
Stiffness	Martini [39]	0.96
	Mota Lontra [40]	0.95
	This research	0.82

This research produced novel insights into the interpretability and consistency of predictive models on the NL-LAB fatigue and stiffness datasets. Interpretability was assessed as the extent to which model predictions in the physics component satisfy the soft constraint imposed by the physical model. The results obtained suggest that the developed PIAI stiffness and fatigue models have an enhanced interpretability, as both models approximated remarkably the infused physical behaviour (see Figure 4-8 and Figure 4-14). Prediction consistency was assessed considering the predictions in the physics component of 10 different model initializations. The average prediction over all initializations approximated the infused physical behaviour in a remarkable way for both PIAI models (see Figure 4-10 and Figure 4-16). Moreover, physics infusion allowed for a reduction in the coefficient of variation in the physics component by 56% and 88% for the PIAI stiffness and fatigue models respectively. These results highlight the PIAI framework's capability to produce predictive models that reliably approximate the infused physical model across various initializations.

5.3. Future recommendations

Building upon the findings and insights gained from this research, a series of recommendations are proposed to guide future work in both data collection and PIAI model development. The recommendations presented in Section 5.3.1 focus on data collection and material testing within the NL-LAB project, whereas the recommendations presented in Section 5.3.2 address important considerations for advancing PIAI model development.

5.3.1. Recommendations on data collection and material testing

During the development of the PIAI framework, it became apparent that the NL-LAB datasets exhibit limited variability in feature values, with some features having fewer than 10 unique values among 371 entries. To address this issue, it is recommended to expand the NL-LAB datasets with additional data instances encompassing a broader range of mix compositions, sample ages, and bitumen functional properties. This increased variability is expected to uncover more intricate relationships between features and targets, thus potentially increasing accuracy in PIAI models.

The need for expanding the datasets is also relevant for addressing the differences in the test frequencies in the DSR test for bitumen shear modulus and the 4PBB test for asphalt mix stiffness. Considering these discrepancies is fundamental when developing future prediction models. Hence, the NL-LAB datasets should be expanded with master curve coefficients for the shear modulus/mix stiffness and the phase angle of bitumen and mix samples. Master curves incorporate the time-temperature superposition principle to enable calculations of visco-elastic material properties at different temperatures and frequency ranges. By accounting for the differences between test frequencies, stronger relationships between bitumen and asphalt mix functional properties could be obtained.

A related challenge lies in the generation of the NL-LAB fatigue dataset targets via extrapolation based on a single fatigue line spanning multiple samples. It is recommended to modify the target of the NL-LAB fatigue dataset to the number of cycles to failure (Nf_{50}). Since Nf_{50} was measured for each sample, a prediction model can potentially find enhanced relationships with this new target variable. Moreover, by incorporating the Ratio of Dissipated Energy Change (RDEC) per load cycle, it is feasible to directly calculate the plateau value (PV). This direct PV calculation could provide a basis for the infusion of new physical models in PIAI fatigue prediction.

5.3.2. Recommendations on advancing PIAI model development

To advance the PIAI framework developed in this research, several improvements are recommended. First, the robustness of data-driven component of feature selection method should be enhanced. Although this research performed a detailed analysis to select an adequate base regression model in BorutaShap [74], this analysis did not include hyperparameter tuning. Thus, it is recommended that future research incorporates hyperparameter tuning in the feature selection procedure to enhance the alignment between the selected features and their effect on model performance.

Beyond feature selection, the exploration of different model architectures can enhance the PIAI models accuracy, interpretability and consistency. The current PIAI framework relies on a neural network to infuse physics in model predictions. It is recommended to explore the feasibility and effectiveness of physics infusion considering other model architectures, such as tree-based learners or support vector machines. Infusing physics in different model architectures can provide new insights into the effects of physics infusion in AI models.

It is also recommended to assess the PIAI models interpretability and consistency on different datasets. Expanding these assessments to a broader range of data can provide a further understanding on the adaptability of the framework and the validity of the infused physical model in different scenarios.

The final recommendation of this research is related to exploring infusion of multiple physical models into a single PIAI framework. Incorporating a wider range of physical constraints can potentially enhance the understanding of the predictions generated. This incorporation can also provide new insights on the interaction of different research areas in pavement engineering, broadening the scope of the PIAI models.

References

[1] C. P. Ng, Law, T H, Jakarni, F M, and Kulanthayan, S, “Road infrastructure development and economic growth,” presented at the 10th Malaysian Road Conference & Exhibition 2018, IOP Publishing, 2019. doi: 10.1088/1757-899X/512/1/012045.

[2] D. Nawir, M. D. Bakri, and I. A. Syarif, “Central government role in road infrastructure development and economic growth in the form of future study: the case of Indonesia,” *City Territ. Archit.*, vol. 10, no. 1, p. 12, Apr. 2023, doi: 10.1186/s40410-022-00188-9.

[3] European Comission, “Highest motorway densities in German & Dutch regions.” Accessed: Aug. 02, 2024. [Online]. Available: https://ec.europa.eu/eurostat/web/products-eurostat-news/-/ddn-20220914-1

[4] OECD, “Asset Management for the Roads Sector,” ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT, Paris, France, 2001.

[5] OECD, “OECD Data Explorer - Transport infrastructure investment and maintenance spending.” Accessed: Aug. 02, 2024. [Online]. Available: https://data-explorer.oecd.org

[6] S. Erkens, J. Stigter, B. Sluer, and R. Khedoe, “NL-LAB: onderzoek naar de voorspellende waarde van proef 62,” 2014.

[7] NEN-EN, “Bituminous mixtures - Test methods - Part 26: Stiffness”.

[8] NEN-EN, “Bituminous mixtures - Test methods - Part 24: Resistance to fatigue”.

[9] J. E. S. L. Teixeira, C. M. Amaecing Junior, L. R. D. Rezende, V. T. F. Castelo Branco, and Y.-R. Kim, “Evaluation of asphalt concrete’s fatigue behavior using cyclic semi-circular bending test,” *Constr. Build. Mater.*, vol. 400, p. 132772, Oct. 2023, doi: 10.1016/j.conbuildmat.2023.132772.

[10] R. L. Carvalho and C. W. Schwartz, “Comparisons of Flexible Pavement Designs: AASHTO Empirical Versus NCHRP Project 1–37A Mechanistic–Empirical,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1947, no. 1, pp. 167–174, Jan. 2006, doi: 10.1177/0361198106194700116.

[11] AASHTO, “Mechanistic-Empirical Pavement Design Guide,” American Association of State Highway and Transportation Officials, 2008.

[12] Y. Xu and Z. Zhang, “Review of Applications of Artificial Intelligence Algorithms in Pavement Management,” *J. Transp. Eng. Part B Pavements*, vol. 148, no. 3, p. 03122001, Sep. 2022, doi: 10.1061/JPEODX.0000373.

[13] S. A. Faroughi *et al.*, “Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks in Scientific Computing,” Feb. 04, 2023, *arXiv*: arXiv:2211.07377. Accessed: Nov. 22, 2023. [Online]. Available: http://arxiv.org/abs/2211.07377

[14] A. Karpatne *et al.*, “Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2318–2331, Oct. 2017, doi: 10.1109/TKDE.2017.2720168.

[15] Y. Deng, H. Wang, and X. Shi, “Physics-guided neural network for predicting asphalt mixture rutting with balanced accuracy, stability and rationality,” *Neural Netw.*, vol. 172, p. 106085, Apr. 2024, doi: 10.1016/j.neunet.2023.12.039.

[16] N. Kargah-Ostadi, K. Vasylevskyi, A. Ablets, and A. Drach, “Reconciling Pavement Condition Data from Connected Vehicles with the International Roughness Index from Standard Monitoring Equipment Using Physics-Integrated Machine Learning,”

Transp. Res. Rec. J. Transp. Res. Board, p. 03611981231174406, Jun. 2023, doi: 10.1177/03611981231174406.

[17] N. Kargah-Ostadi, K. Vasylevskyi, A. Ablets, and A. Drach, “Physics-informed neural networks to advance pavement engineering and management,” *Road Mater. Pavement Des.*, pp. 1–22, Feb. 2024, doi: 10.1080/14680629.2024.2315073.

[18] C. Han, J. Zhang, Z. Tu, and T. Ma, “PINN-AFP: A novel C-S curve estimation method for asphalt mixtures fatigue prediction based on physics-informed neural network,” *Constr. Build. Mater.*, vol. 415, p. 135070, Feb. 2024, doi: 10.1016/j.conbuildmat.2024.135070.

[19] Y. Liu, P. Su, M. Li, Z. You, and M. Zhao, “Review on evolution and evaluation of asphalt pavement structures and materials,” *J. Traffic Transp. Eng. Engl. Ed.*, vol. 7, no. 5, pp. 573–599, Oct. 2020, doi: 10.1016/j.jtte.2020.05.003.

[20] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An Introduction to Statistical Learning with Applications in Python*. 2023.

[21] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, “Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges,” Jul. 09, 2021, *arXiv*: arXiv:2103.11251. Accessed: Aug. 16, 2024. [Online]. Available: http://arxiv.org/abs/2103.11251

[22] D. Picard, “Torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision,” May 11, 2023, *arXiv*: arXiv:2109.08203. Accessed: Aug. 13, 2024. [Online]. Available: http://arxiv.org/abs/2109.08203

[23] L. Liu and M. Tamer Özsu, *Encyclopedia of Database Systems*, 1st ed., vol. 1. Springer, 2009.

[24] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, May 2021, doi: 10.1038/s42254-021-00314-5.

[25] S.-P. Zhu *et al.*, “Physics-informed machine learning and its structural integrity applications: state of the art,” *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.*, vol. 381, no. 2260, p. 20220406, Nov. 2023, doi: 10.1098/rsta.2022.0406.

[26] S. J. D. Prince, *Understanding Deep Learning*. The MIT Press, 2024.

[27] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/j.jcp.2018.10.045.

[28] N. Rahaman *et al.*, “On the Spectral Bias of Neural Networks,” May 31, 2019, *arXiv*: arXiv:1806.08734. Accessed: Jun. 22, 2024. [Online]. Available: http://arxiv.org/abs/1806.08734

[29] T. Kapoor, H. Wang, A. Núñez, and R. Dollevoet, “Physics-Informed Neural Networks for Solving Forward and Inverse Problems in Complex Beam Systems,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2023, doi: 10.1109/TNNLS.2023.3310585.

[30] T. Kapoor, H. Wang, A. Núñez, and R. Dollevoet, “Physics-informed machine learning for moving load problems,” Apr. 01, 2023, *arXiv*: arXiv:2304.00369. Accessed: Nov. 23, 2023. [Online]. Available: http://arxiv.org/abs/2304.00369

[31] A. Sivaakumar, “”Creating Your Own Model Architecture in Machine Learning: A Step-by-Step Guide”,” Medium. Accessed: Jul. 21, 2024. [Online]. Available:

https://aiswaryasivakumar8.medium.com/creating-your-own-model-architecture-in-machine-learning-a-step-by-step-guide-522236c340d2

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. Accessed: Jul. 21, 2024. [Online]. Available: https://www.deeplearningbook.org/

[33] T. Kapoor, A. Chandra, D. M. Tartakovsky, H. Wang, A. Nunez, and R. Dollevoet, "Neural oscillators for generalization of physics-informed machine learning," Dec. 18, 2023, *arXiv*: arXiv:2308.08989. Accessed: Jul. 21, 2024. [Online]. Available: http://arxiv.org/abs/2308.08989

[34] "LTPP InfoPave - Home." Accessed: Mar. 15, 2024. [Online]. Available: https://infopave.fhwa.dot.gov/

[35] T. Kapoor, H. Wang, A. Núñez, and R. Dollevoet, "Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations," *Eng. Appl. Artif. Intell.*, vol. 133, p. 108085, Feb. 2024, doi: 10.1016/j.engappai.2024.108085.

[36] A. Daw, A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling," Sep. 28, 2021, *arXiv*: arXiv:1710.11431. Accessed: Mar. 08, 2024. [Online]. Available: http://arxiv.org/abs/1710.11431

[37] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, "Machine learning in materials informatics: recent applications and prospects," *Npj Comput. Mater.*, vol. 3, no. 1, p. 54, Dec. 2017, doi: 10.1038/s41524-017-0056-5.

[38] Rijkswaterstaat, "Nederlands Langjarig Asfalt Bemonstering (NL-LAB)," Duurzame - Infra. Accessed: Feb. 28, 2024. [Online]. Available: https://www.duurzame-infra.nl/kennis-informatie/knowlegde-based-pavement-engineering/nl-lab

[39] G. Martini, "Predictive Modelling of Asphalt Concrete Functional Properties Using Multiple Linear Regression and Gradient Boosting," Delft University of Technology, Delft, Oct. 2019.

[40] B. Mota Lontra, "Machine learning study on performance prediction of asphalt mixtures under Dutch conditions," Delft University of Technology, Delft, Dec. 2022.

[41] G. Seleridis, "Evaluation of the current test methods of water senstivity and permanent deformation," Delft University of Technology, Delft, Sep. 2017.

[42] NEN-EN, "Bitumen and bituminous binders - Determination of needle penetration".

[43] NEN-EN, "Bitumen and bituminous binders - Determination of the softening point - Ring and Ball method".

[44] NEN-EN, "Bitumen and bituminous binders - Determination of complex shear modulus and phase angle - Dynamic Shear Rheometer (DSR)".

[45] NEN-EN, "Bituminous mixtures - Test methods for hot mix asphalt - Part 6: Determination of bulk density of bituminous specimens," 2012.

[46] NEN-EN, "Bituminous mixtures - Test methods - Part 25: Cyclic compression test".

[47] NEN-EN, "Bituminous mixtures - Test methods - Part 12: Determination of the water sensitivity of bituminous specimens".

[48] F. L. Roberts, P. S. Kandhal, E. R. Brown, D.-Y. Lee, and T. W. Kennedy, *Hot Mix Asphalt Materials, Mixture Design and Construction*, 2nd ed. National Asphalt Pavement Association Research and Education Foundation, 1991.

[49] J. Read and D. Whiteoak, *The Shell Bitumen Handbook*, 5th ed. Thomas Telford Publishing, 2003.

[50] A. T. Papagiannakis and E. A. Masad, *Pavement Design and Materials*, 1st ed. Wiley, 2012. doi: 10.1002/9780470259924.

[51] Y. H. Huang, *Pavement Analysis and Design*, 2nd ed. Upple Saddle River, NJ: Pearson Education Inc., 2004.

[52] W. Zeiada *et al.*, “Review of the Superpave performance grading system and recent developments in the performance-based test methods for asphalt binder characterization,” *Constr. Build. Mater.*, vol. 319, p. 126063, Feb. 2022, doi: 10.1016/j.conbuildmat.2021.126063.

[53] NEN-EN, “Bituminous mixtures - Test methods - Part 8: Determination of void characteristics of bituminous specimens,” Jan. 2019.

[54] NEN-EN, “Bituminous Mixtures - Test Methods - Part 5: Determination of the Maximum Density,” 2019.

[55] R. N. Linden, J. P. Mahoney, and N. C. Jackson, “Effect of Compaction on Asphalt Concrete Performance,” *Transportation Research Record*, 1989.

[56] W. S. Mogawer, A. J. Austerman, J. S. Daniel, F. Zhou, and T. Bennert, “Evaluation of the effects of hot mix asphalt density on mixture fatigue performance, rutting performance and MEPDG distress predictions,” *Int. J. Pavement Eng.*, vol. 12, no. 2, pp. 161–175, Apr. 2011, doi: 10.1080/10298436.2010.546857.

[57] J. Droogers, “Asphalt concrete stiffness prediction based on composition and binder properties,” Delft University of Technology, 2018.

[58] H. Zhang, K. Anupam, A. Scarpas, and C. Kasbergen, “Issues in the Prediction of the Mechanical Properties of Open Graded Mixes,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2672, no. 40, pp. 32–40, Dec. 2018, doi: 10.1177/0361198118792117.

[59] D. W. Christensen, T. Pellinen, and R. F. Bonaquist, “Hirsch Model for Estimating the Modulus of Asphalt Concrete,” *J. Assoc. Asph. Paving Technol.*, vol. 72, pp. 97–121, 2003.

[60] D. W. Christensen and R. Bonaquist, “Improved Hirsch model for estimating the modulus of hot-mix asphalt,” *Road Mater. Pavement Des.*, vol. 16, no. sup2, pp. 254–274, Aug. 2015, doi: 10.1080/14680629.2015.1077635.

[61] F. S. Brown and J. M. Brunton, *An introduction to the analytical design of bituminous pavements*, 3rd ed. Nottingham: University of Nottingham.

[62] D. Hwang and M. W. Witzcak, *Program DAMA (Chevron), User’s Manual*. Department of Civil Engineering, University of Maryland, 1979.

[63] M. W. Witczak and O. A. Fonseca, “Revised Predictive Model for Dynamic (Complex) Modulus of Asphalt Mixtures,” *Transp. Res. Rec.*, no. 1540, pp. 15–23, 1996.

[64] H. U. Bahia, D. I. Hanson, M. Zeng, H. Zhai, M. A. Khatri, and R. M. Anderson, “Characterization of modified asphalt binders in Superpave mix design,” National Cooperative Highway Research Program, Washington, D.C., 459, 2001.

[65] M. A. Ishaq and F. Giustozzi, “Correlation between Rheological Fatigue Tests on Bitumen and Various Cracking Tests on Asphalt Mixtures,” *Materials*, vol. 14, no. 24, p. 7839, Dec. 2021, doi: 10.3390/ma14247839.

[66] S. Shen and S. H. Carpenter, “Dissipated Energy Concepts for HMA Performance: Fatigue and Healing,” University of Illinios at Urbana-Champaign, COE Report No. 29, Mar. 2007.

[67] AASHTO, *Standard Method of Test for Determining Dynamic Modulus of Hot Mix Asphalt (HMA)*, T 342-11, United States of America., 2015.

[68] S. H. Carpenter, K. A. Ghuzlan, and S. Shen, “Fatigue Endurance Limit for Highway and Airport Pavements,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1832, no. 1, pp. 131–138, Jan. 2003, doi: 10.3141/1832-16.

[69] D. Sasidharan, U. Saravanan, and J. M. Krishnan, “A methodology for post-processing the four-point beam bending data and computing stiffness modulus using harmonic analysis,” *Constr. Build. Mater.*, vol. 396, p. 132164, Sep. 2023, doi: 10.1016/j.conbuildmat.2023.132164.

[70] J. Brownlee, *Data Preparation for Machine Learning Data Cleaning, Feature Selection, and Data Transforms in Python*, V1.1. in Machine Learning Mastery. 2020.

[71] “AASHTO Soil Terminology – Pavement Interactive.” Accessed: Jun. 15, 2024. [Online]. Available: https://pavementinteractive.org/aashto-soil-terminology/

[72] A. Zheng and A. Casari, *Feature Engineering for Machine Learning*, 1st ed. United States of America: O’Reilly, 2018.

[73] L. Martinez Molera, “Machine Learning Q&A: All About Model Validation.” Accessed: Jun. 05, 2024. [Online]. Available: https://nl.mathworks.com/campaigns/offers/next/all-about-model-validation.html

[74] E. Keany, *BorutaShap : A wrapper feature selection method which combines the Boruta feature selection algorithm with Shapley values.* (Nov. 05, 2020). Zenodo. doi: 10.5281/zenodo.4247618.

[75] M. B. Kursa and W. R. Rudnicki, “Feature Selection with the Boruta Package,” *J. Stat. Softw.*, vol. 36, no. 11, 2010, doi: 10.18637/jss.v036.i11.

[76] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” Nov. 24, 2017, *arXiv*: arXiv:1705.07874. Accessed: Mar. 01, 2024. [Online]. Available: http://arxiv.org/abs/1705.07874

[77] M. B. Kursa and W. R. Rudnicki, “Boruta: Wrapper Algorithm for All Relevant Feature Selection.” p. 8.0.0, Dec. 06, 2009. doi: 10.32614/CRAN.package.Boruta.

[78] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Accessed: Jun. 12, 2024. [Online]. Available: https://christophm.github.io/interpretable-ml-book/shap.html

[79] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-6849-3.

[80] M. Berangi, B. Mota Lontra, K. Anupam, S. Erkens, D. Van Vliet, and M. Moenielal, “Gradient boosting decision trees to study laboratory and field performance in pavement management,” *Comput Aided Civ Inf.*, Aug. 2024, doi: 10.1111/mice.13322.

[81] “Welcome to the SHAP documentation — SHAP latest documentation.” Accessed: Aug. 15, 2024. [Online]. Available: https://shap.readthedocs.io/en/latest/

[82] V. Trevisan, “Boruta SHAP: an amazing tool for feature selection every data scientist should know,” Medium. Accessed: Jun. 12, 2024. [Online]. Available: https://towardsdatascience.com/boruta-shap-an-amazing-tool-for-feature-selection-every-data-scientist-should-know-33a5f01285c0

[83] J. Loughrey and P. Cunningham, “Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets,” in *Research and Development in Intelligent Systems XXI*, M. Bramer, F. Coenen, and T. Allen, Eds., London: Springer London, 2005, pp. 33–43. doi: 10.1007/1-84628-102-4_3.

[84] “RandomForestRegressor,” scikit-learn. Accessed: Aug. 07, 2024. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[85] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[86] A. V. Dorogush, V. Ershov, and A. Gulin, “CatBoost: gradient boosting with categorical features support”.

[87] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006, doi: 10.1007/s10994-006-6226-1.

[88] C. Spearman, “The proof and measurement of association between two things,” vol. 15, Jan. 1904.

[89] C. M. Bishop, *Pattern Recognition and Machine Learning*. in Information Science and Statistics. Cambridge, U.K.: Springer, 2006.

[90] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” Dec. 03, 2019, *arXiv*: arXiv:1912.01703. Accessed: Mar. 08, 2024. [Online]. Available: http://arxiv.org/abs/1912.01703

[91] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks”.

[92] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Jan. 29, 2017, *arXiv*: arXiv:1412.6980. Accessed: Jun. 16, 2024. [Online]. Available: http://arxiv.org/abs/1412.6980

[93] L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.

[94] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” Jul. 25, 2019, *arXiv*: arXiv:1907.10902. Accessed: Jun. 24, 2024. [Online]. Available: http://arxiv.org/abs/1907.10902

[95] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Comput. Sci.*, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.

[96] A. Mohammed and R. Kora, “A comprehensive review on ensemble deep learning: Opportunities and challenges,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 2, pp. 757–774, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.014.

[97] L. S. Shapley, “A value for n-person games,” The RAND Corporation, Santa Monica, CA, USA, Mar. 1952.

[98] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for Hyper-Parameter Optimization”.

[99] F. Hutter, H. Hoos, and K. Leyton-Brown, “An Efficient Approach for Assessing Hyperparameter Importance”.

Appendix A. Features in the NL-LAB Datasets

This appendix details the features contained the NL-LAB datasets during different stages of PIAI model development. Table A 1 shows an overview of the raw features contained in the dataset, prior to the data preparation stage. Table A 2 provides and overview of the resulting features in the datasets after implementing the data preparation procedures discussed in Section 3.2. Finally, Table A 3 shows the final features included in the PIAI models for stiffness and fatigue prediction. These features were deemed important after applying the physics and data-driven feature selection method outlined in Section 3.3.

Table A 1. Raw features from the NL-LAB datasets.

Feature	Description	Category	Dataset
filename	Filename	Informational	All
phase_tot	Abbreviation Work-Phase-Lab-Year	Informational	All
work	Work	Categorical	All
phase	Phase	Categorical	All
lab	Laboratory	Informational	All
year	Measurement year	Mix property	All
monsternames	Sample name	Informational	All
densities	Specimen (bulk) density	Mix property	All
VA	Air voids	Mix property	All
bit2_TRenK	Softening point of recovered binder [°C]	Bitumen property	All
bit2_pen	Penetration of recovered binder [0.1 mm]	Bitumen property	All
bit2_Gstar	Complex modulus [G*] of recovered binder [Pa] @ 10 rad/s, 20°C	Bitumen property	All
bit2_delta	Phase angle [δ] of recovered binder [°] @ 10 rad/s, 20°C	Bitumen property	All
target_density	Target density [kg/m³]	Mix property	All
percentage_PR	Percentage of Reclaimed Asphalt in the mix	Mix property	All
percentage_bit	Percentage of bitumen in the mix	Mix property	All
Volume_Target_C22_4	Target volume fraction of C22/4 aggregate	Mix property	All
Volume_Target_C16	Target volume fraction of C16 aggregate	Mix property	All
Volume_Target_C11_2	Target volume fraction of C11/2 aggregate	Mix property	All
Volume_Target_C08	Target volume fraction of C08 aggregate	Mix property	All
Volume_Target_C05_6	Target volume fraction of C05/6 aggregate	Mix property	All

Feature	Description	Category	Dataset
Volume_Target_C002mm	Target volume fraction of C002mm aggregate	Mix property	All
Volume_Target_C00063mu	Target volume fraction of C00063mu aggregate	Mix property	All
Volume_Target_filler	Target volume fraction of filler	Mix property	All
Volume_Target_bitumen	Target volume fraction of bitumen	Mix property	All
Mass_Target_C22_4	Target mass fraction of C22/4 aggregate	Mix property	All
Mass_Target_C16	Target mass fraction of C16 aggregate	Mix property	All
Mass_Target_C11_2	Target mass fraction of C11/2 aggregate	Mix property	All
Mass_Target_C08	Target mass fraction of C08 aggregate	Mix property	All
Mass_Target_C05_6	Target mass fraction of C05/6 aggregate	Mix property	All
Mass_Target_C002mm	Target mass fraction of C002mm aggregate	Mix property	All
Mass_Target_C00063mu	Target mass fraction of C00063mu aggregate	Mix property	All
Mass_Target_filler	Target mass fraction of filler	Mix property	All
Mass_Target_bitumen	Target mass fraction of bitumen	Mix property	All
mix_setup	Type of mixer	Categorical	All
comp_setup	Type of compactor	Categorical	All
stiffness	Stiffness [MPa] @ 8 Hz, 20°C	Target	Stiffness
EPS6_individual	Individual calculated strain at 10^6 load cycles [με]	Target	Fatigue

Table A 2. Features of the NL-LAB datasets after data preparation.

Feature Description	Type	Unique Values	Min. Value	Max. Value	Mean	Standard deviation
Specimen (bulk) density [kg/m^3]	Continuous	371	2,307.81	2,453.60	2,391.09	31.71
Air voids [-] (fa)	Continuous	371	0.01	0.07	0.04	0.02
Softening point of recovered binder [°C]	Continuous	22	55.80	82.60	63.32	8.02
Penetration of recovered binder [0.1 mm]	Continuous	16	11.00	53.00	25.54	10.55
Complex modulus [G*] of recovered binder [Pa] @ 10 rad/s, 20°C	Continuous	23	194,139.51	6,632,724.39	1,140,421.61	1,650,702.97
Phase angle [δ] of recovered binder [°] @ 10 rad/s, 20°C	Continuous	23	39.96	66.57	58.35	8.16

Feature Description	Type	Unique Values	Min. Value	Max. Value	Mean	Standard deviation
Target density [kg/m³]	Continuous	6	2,360.00	2,399.00	2,380.85	10.71
Percentage of Reclaimed Asphalt in the mix [%]	Continuous	4	50.00	65.00	58.28	6.19
Target volume fraction of filler [%]	Continuous	6	5.76	7.65	6.30	0.60
Target volume fraction of bitumen [%] (fb)	Continuous	6	0.10	0.13	0.11	0.01
Target volume fraction of Gravel	Continuous	6	36.42	42.48	39.54	2.10
Target volume fraction of Sand	Continuous	6	38.94	46.63	43.24	2.19
Measurement year	Continuous	4	0.00	6.00	0.54	1.23
Binary indicator of Polymer Modified Bitumen	Categorical (binary)	2	0.00	1.00	N/A	N/A
Binary indicator of Forced action mixer	Categorical (binary)	2	0.00	1.00	N/A	N/A
Binary indicator of Planetory mixer	Categorical (binary)	2	0.00	1.00	N/A	N/A
Binary indicator of Field roller compactor	Categorical (binary)	2	0.00	1.00	N/A	N/A
Binary indicator of Mini roller compactor	Categorical (binary)	2	0.00	1.00	N/A	N/A
Binary indicator of Segment compactor	Categorical (binary)	2	0.00	1.00	N/A	N/A
Binary indicator of Shear box compactor	Categorical (binary)	2	0.00	1.00	N/A	N/A
Volume fraction of Aggregates [-] (fa)	Continuous	371	0.82	0.89	0.85	0.01
Aggregate organization factor [-]	Continuous	371	0.29	0.87	0.51	0.16
Aggregate Gradation Parameter [-]	Continuous	6	0.39	0.50	0.45	0.04
Volumetric Parameter [-]	Continuous	244	0.05	0.40	0.24	0.09
Stiffness [MPa]	Continuous	371	6,144.00	17,866.47	9,737.44	2,095.35

Feature Description	Type	Unique Values	Min. Value	Max. Value	Mean	Standard deviation
Initial strain yielding a fatigue life of 10^6 cycles [με]	Continuous	371	37.14	177.48	112.78	22.70

Table A 3. Final features included in the PIAI stiffness and fatigue prediction models.

Feature	Feature Description	Model	Component that deemed important
"bit2_pen"	Penetration of recovered binder [0.1 mm]	Stiffness	Data
"bit2_Gstar"	Complex modulus [G*] of recovered binder [Pa] @ 10 rad/s, 20°C'	Both	Data and Physics (Stiffness model) Data (Fatigue model)
"bit2_delta"	Phase angle [δ] of recovered binder [°] @ 10 rad/s, 20°C	Both	Data
"target_density"	Target density [kg/m³]	Both	Data
"percentage_PR"	Percentage of Reclaimed Asphalt in the mix [%]	Both	Data
"Volume_Target_filler"	Target volume fraction of filler [%]	Both	Data
"Volume_Target_bitumen"	Target volume fraction of bitumen [%] (fb)	Both	Data and Physics (Stiffness model) Data (Fatigue model)
"Volume_Target_Sand"	Target volume fraction of Sand	Stiffness	Data
"year"	Measurement year	Both	Data
"PMB"	Binary indicator of Polymer Modified Bitumen	Fatigue	Data
"Forced action mixer"	Binary indicator of Forced action mixer	Stiffness	Data
"Mini roller"	Binary indicator of Mini roller compactor	Stiffness	Data
"Segment compactor"	Binary indicator of Segment compactor	Stiffness	Data
"Volume_Agg_Fraction"	Volume fraction of Aggregates [-] (fa)	Stiffness	Data and Physics

Feature	Feature Description	Model	Component that deemed important
“Pa”	Aggregate organization factor [-]	Stiffness	Data and Physics
“GP”	Aggregate Gradation Parameter [-]	Fatigue	Data and Physics
“VP”	Volumetric Parameter [-]	Fatigue	Data and Physics
“stiffness”	Stiffness [MPa]	Fatigue	Physics

Appendix B. Gradation Curves

Figure B 1 illustrates the gradation curves of the NL-LAB stiffness and fatigue datasets. As both datasets utilize the same set of raw features, the gradation curves are identical. These curves depict the sieve sizes associated with features corresponding to gravel and sand content. Additionally, the plots highlight the passing percentages for the assumed Nominal Maximum Aggregate Size (NMAS) and Primary Control Sieve (PCS), as discussed in Section 3.1.2.

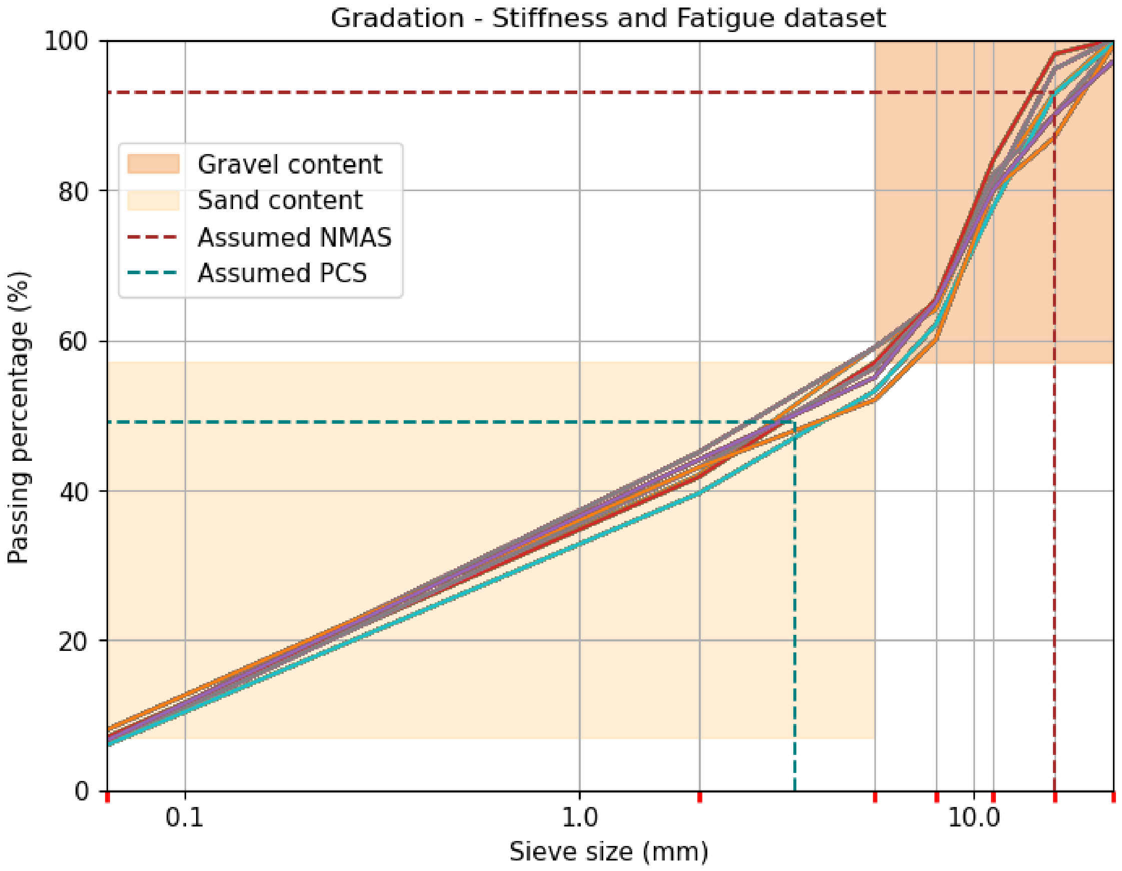


Figure B 1. Gradation curves for the stiffness dataset.

Appendix C. Mix and compaction setups on the NL-LAB datasets

This appendix provides further background behind the decision to remove the "work" and "phase" features from the datasets during the data preparation stage. Table C 1 presents the relationship between "work" and "phase" combinations and "mix setup" and "comp. setup" combinations. Since every possible "work-phase" combination is reflected in a "mix setup – comp setup" combination, the features "work" and "phase" could be removed from the dataset. Moreover, the "work" feature refers to the project number where each sample originates [6], which by itself is not expected to influence pavement performance predictions. However, since it is known that work 6 contains Polymer Modified Bitumen (PMB) [6], an additional feature was created to distinguish samples with PMB.

Table C 1. Phase-Work and Mix-Compaction setup relationship in the NL-LAB datasets.

Phase		1						2						3					
Work		1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
Mix Setup	Comp. Setup	Number of entries																	
Asphalt plant	Field roller	-	-	-	-	-	-	-	-	-	-	-	-	54	18	29	54	36	18
	Hand roller	-	-	-	-	-	-	-	6	-	-	-	-	-	-	-	-	-	-
	Mini roller	-	-	-	-	-	-	9	-	-	18	-	-	-	-	-	-	-	-
	Segment compactor	-	-	-	-	-	-	9	18	-	-	18	18	-	-	-	-	-	-
	Shear box	-	-	-	-	-	-	-	-	10	-	-	-	-	-	-	-	-	-
Forced action mixer	Hand roller	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Mini roller	-	-	-	18	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Segment compactor	8	18	-	-	18	18	-	-	-	-	-	-	-	-	-	-	-	-
	Shear box	-	-	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Planetory mixer	Mini roller	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Appendix D. Supplementary Material on SHAP Values

SHAP values are used in BorutaShap as the feature importance score, as mentioned in Section 3.3. SHAP stands for Shapley Additive Explanation [76] values. SHAP values serve the purpose of explaining a prediction of an instance **x** by computing the contribution of each feature to the prediction [78]. For doing so, a mathematical foundation on cooperative game theory is used, in which the features (players) form coalitions to make predictions (game) [78]. Lloyd Shapley introduced the Shapley value [97] as a method to understand the contribution of a specific member of a coalition in the value generation task [40].

Consider a model *f(*x) that is explained by an explanation model *g(z')* as shown in Equation D 1. Figure D 1 illustrates how the Shap values ϕ_j explain how to get from the model output when no features are known (*E[f(z)]*), to the current output *f(x)* are known [76].

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j \cdot z'_j$$

Equation D 1

Where:

$z' \in \{0,1\}^M$: Coalition vector.

M : Maximum coalition size.

ϕ_0 : Null output of the explanation model.

ϕ_j : Feature attribution (Shapley value) for a feature *j*.

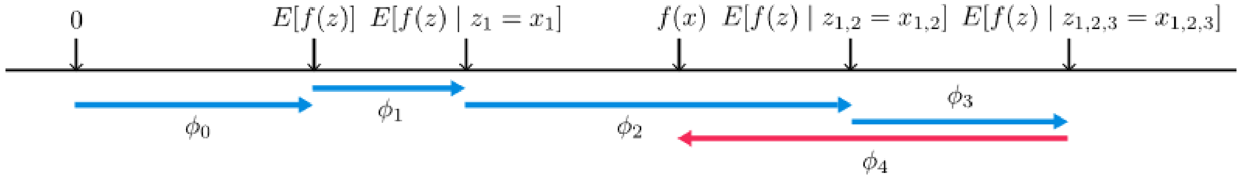


Figure D 1. SHAP values illustration ([76]).

It is worth noting that the order in which the coalition is formed is important, and therefore, the SHAP values for a feature *j* on a single prediction are calculated as the mean value of ϕ_j across all possible coalitions [76]. Furthermore, the global SHAP value is the one used in BorutaShap [74]. This is calculated by averaging the absolute Shapley values per features across the data [78], as shown in Equation D 2.

$$I_j = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| \phi_j^{(i)} \right|$$

Equation D 2

Where:

I_j : Global SHAP importance value for feature *j*.

$\phi_j^{(i)}$: SHAP importance value for feature *j*, in data instance *i*.

n: Number of instances (rows) in the data.

Appendix E. Supplementary Material on Bayesian Optimization

Different approaches can be followed to find an optimal set of hyperparameters, such as Manual Search, Grid Search, Random Search, Gradient-based optimization, and Bayesian Optimization [93]. Among these methods, Bayesian Optimization offers faster computational efficiency by leveraging previous results to decide whether narrowing down or expanding the search space [93]. Yang & Shami [93] provided an overview of the Bayesian Optimization procedure, as follows:

1. Generating a probabilistic surrogate model of the objective function. In this research the objective function is the cross-validation loss (see Section 3.4). The surrogate model is represented by the probability of a model performance score *y* given the hyperparameters *x* **p(*y*|*x*)** [40].

2. Determining the optimal hyperparameters on the surrogate model.

3. Evaluating the original objective function with the newly obtained hyperparameters.

4. Updating the surrogate model with the results obtained.

5. Repeating steps 2-4 for a given number of iterations.

This research employed the Optuna [94] package for Bayesian Optimization. Optuna uses a Tree-structured Parzen Estimator (TPE) [98] as probabilistic surrogate model [94]. Using a TPE is advantageous as it enables optimization for combinations of continuous and discrete hyperparameters [26]. With TPE, the probabilistic surrogate model is constructed by modelling *p(x|y)* and *p(y)* and using Bayes' theorem to get the posterior probability, as shown in Equation E 1 [98].

$$p(y \mid x) = \frac{p(x \mid y) \cdot p(y)}{p(x)}$$

Equation E 1

where:

x: Model hyperparameters.

y: Model performance score.

TPE approximates the likelihood function *p(x|y)* using two non-parametric density functions *l(x)* and *g(x)*, as shown in Equation E 2 [98].

$$p(x \mid y) = \begin{cases} l(x) & \text{if } y < y^{*} \\ g(x) & \text{if } y \geq y^{*} \end{cases}$$ Equation E 2

where:

l(x): Density formed by using the observations $x^{(i)}$ such that the loss $f(x^{(i)})$ was less than y^{*}.
g(x): Density formed by using the remaining observations.
y^{*}: Threshold of the objective function.

Furthermore, TPE uses the Expected Improvement (EI) [98] criterion to find the optimal set of model hyperparameters (x^{*}). Equation E 3 illustrates the EI calculation procedure. The EI is maximized by values of *x* that have high probability under l(x) and low probability under g(x) [98]. Finally, in every iteration, TPE will return the candidate value x* with the highest EI [98].

$$EI = \int_{-\infty}^{y^{*}} \left(y^{*} - y\right) p\left(y \mid x\right) dy \ \propto \ \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1}$$ Equation E 3

where:

EI: Expected Improvement.
y: Objective function value.
y^{*}: Threshold of the objective function.
γ: Quantile of the observed values so that $p(y<y^{*}) = \gamma$

Through Optina, it is also possible to obtain hyperparameter importances [94]. Optuna uses using the f-ANOVA [99] hyperparameter importance evaluation algorithm [94]. This algorithm fits a Random Forest Regressor that predicts the objective function value obtained in every trial from the set of hyperparameters used in said trial [94].

Appendix F. Results of trial model runs

This appendix presents the results of the trial runs of the PIAI models to determine the range of λ values to include in Bayesian Optimization, as mentioned in Section 3.4. The lower λ value is that in in which PIAI model predictions approximate the model targets but do not infuse the physical model. Conversely, the upper value of the range is a λ value that guarantees physics infusion at the expense of a decreased prediction accuracy.

To preliminarily evaluate the effects of λ on prediction accuracy and physics component of PIAI models, the architecture shown in Table F 1 was used for model training. This preliminary architecture was used for both PIAI stiffness and fatigue models and corresponds to common practice in AI predictive modelling [26]. It should be noted that the final model architecture differs per model and was obtained via Bayesian Optimization in a later stage.

Table F 1. Model architecture – PIAI stiffness model.

Hidden Layers	**Nodes per Layer**	**Activation Function**	**Dropout rate**	**Learning rate**
3	64	ReLU (linear activation in outer layer)	0	$1x10^{-4}$

For the preliminary PIAI stiffness model, the evaluation range for λ varies between λ = $1x10^{-12}$ and λ = $1x10^{-6}$. Figure F 1 and Figure F 2 show the results on the data and physics component of the lower value of λ ($1x10^{-12}$). The results obtained show an adequate model accuracy in the data-driven component but a limited capability of physics infusion. Conversely, Figure F 3 and Figure F 4 show the effects of the upper value of λ ($1x10^{-6}$) on the data and physics components of the preliminary model. These results show that the upper λ value approximates physics but sacrifices prediction accuracy. It is expected that the optimal λ value for the PIAI stiffness model lies between λ = $1x10^{-12}$ and λ = $1x10^{-6}$.

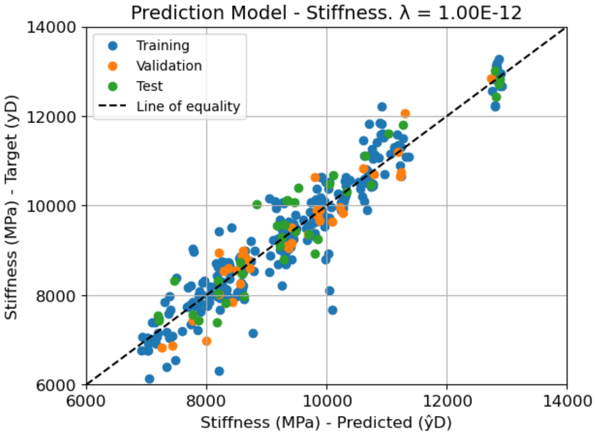


Figure F 1. Predictions vs. Targets – PIAI stiffness model (Data component). Trial with λ = 1x10^{-12}

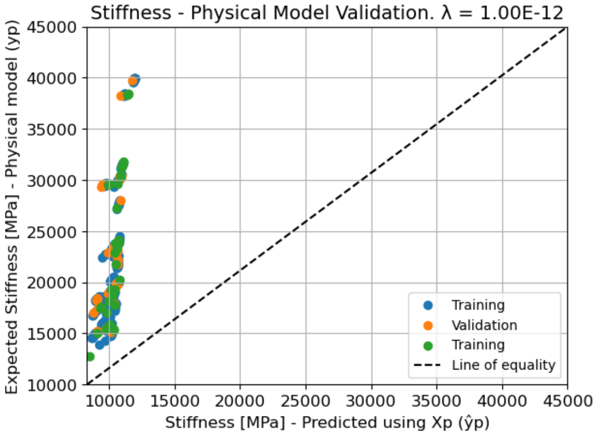


Figure F 2. Predictions vs. Targets – PIAI stiffness model (Physics component). Trial with λ = 1x10^{-12}

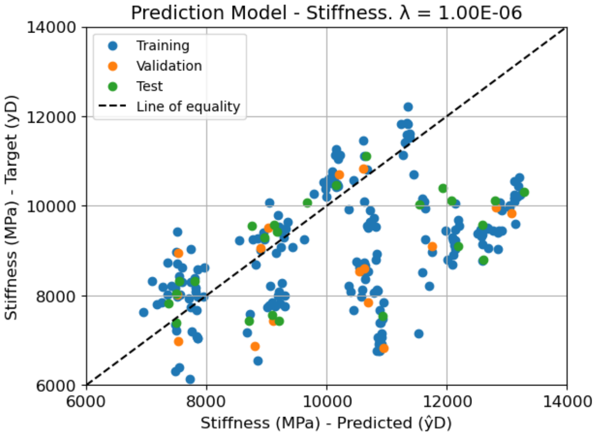


Figure F 3. Predictions vs. Targets – PIAI stiffness model (Data component). Trial with λ = 1x10^{-6}

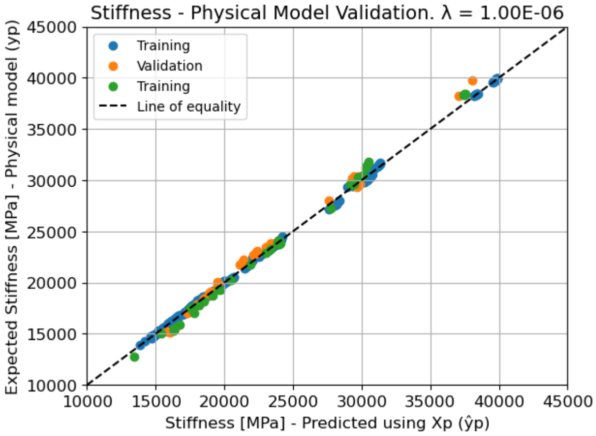


Figure F 4. Predictions vs. Targets – PIAI stiffness model (Physics component). Trial with λ = 1x10^{-6}

In the preliminary PIAI fatigue model, the selected evaluation range for λ varies between λ = 1x10^{-7} and λ = 1x10^{-1}. Figure F 5 and Figure F 6 show the results on the data and physics component of the lower value of λ (1x10^{-7}). The results obtained show an adequate model accuracy in the data-driven and no physics infusion in model predictions. Conversely, Figure F 7 and Figure F 8 show the effects of the upper value of λ (1x10^{-1}) on the data and physics components of the preliminary model. These results show that the upper λ value approximates physics at the expense of a reduce prediction sacrifices prediction accuracy. Hence is expected that the optimal λ value for the PIAI stiffness model lies between λ = 1x10^{-7} and λ = 1x10^{-1}.

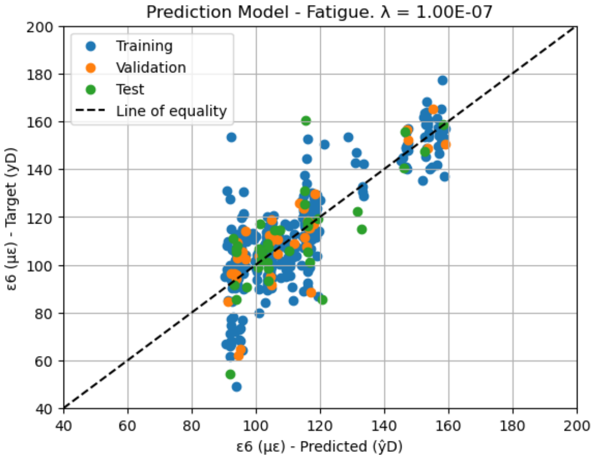


Figure F 5. Predictions vs. Targets – PIAI fatigue model (Data component). Trial with λ = 1x10^{-7}

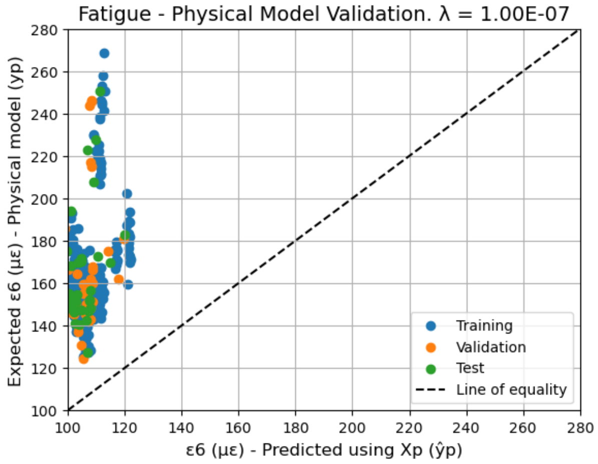


Figure F 6. Predictions vs. Targets – PIAI fatigue model (Physics component). Trial with λ = 1x10^{-7}

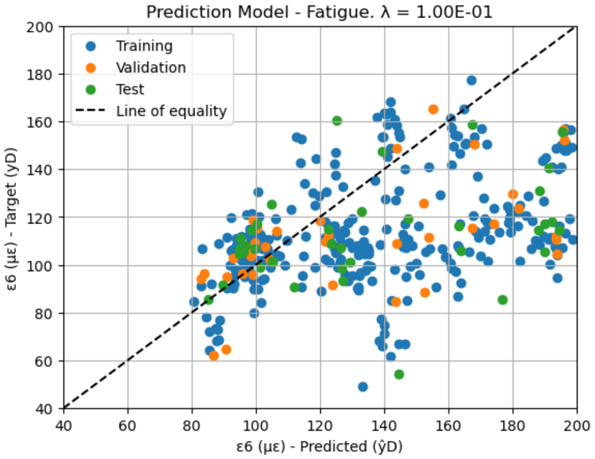


Figure F 7. Predictions vs. Targets – PIAI fatigue model (Data component). Trial with λ = 1x10^{-1}

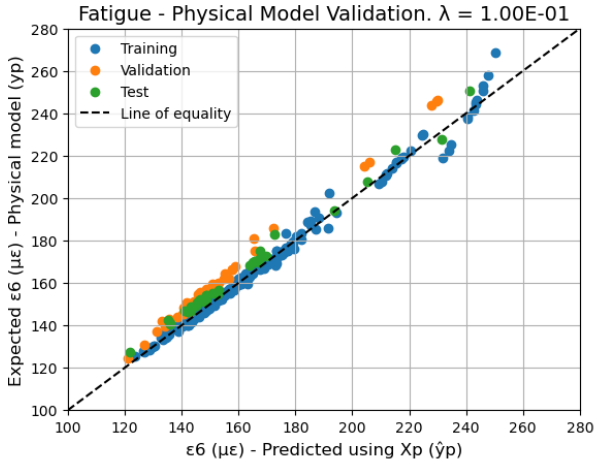


Figure F 8. Predictions vs. Targets – PIAI fatigue model (Physics component). Trial with λ = 1x10^{-1}

Appendix G. Correlation results

This appendix presents an extended version of the correlation analyses performed after feature selection in the stiffness and fatigue datasets. Three figures are presented for each dataset. The first figure corresponds to the correlation matrix comprising the entire set of accepted features. The second figure corresponds to the reduced correlation matrix with only the elements reporting a high (|ρ| > 0.8) Spearman correlation rank. The third figure presents the correlation matrix after removal of the highly correlated features in each dataset (see Section 4.1).

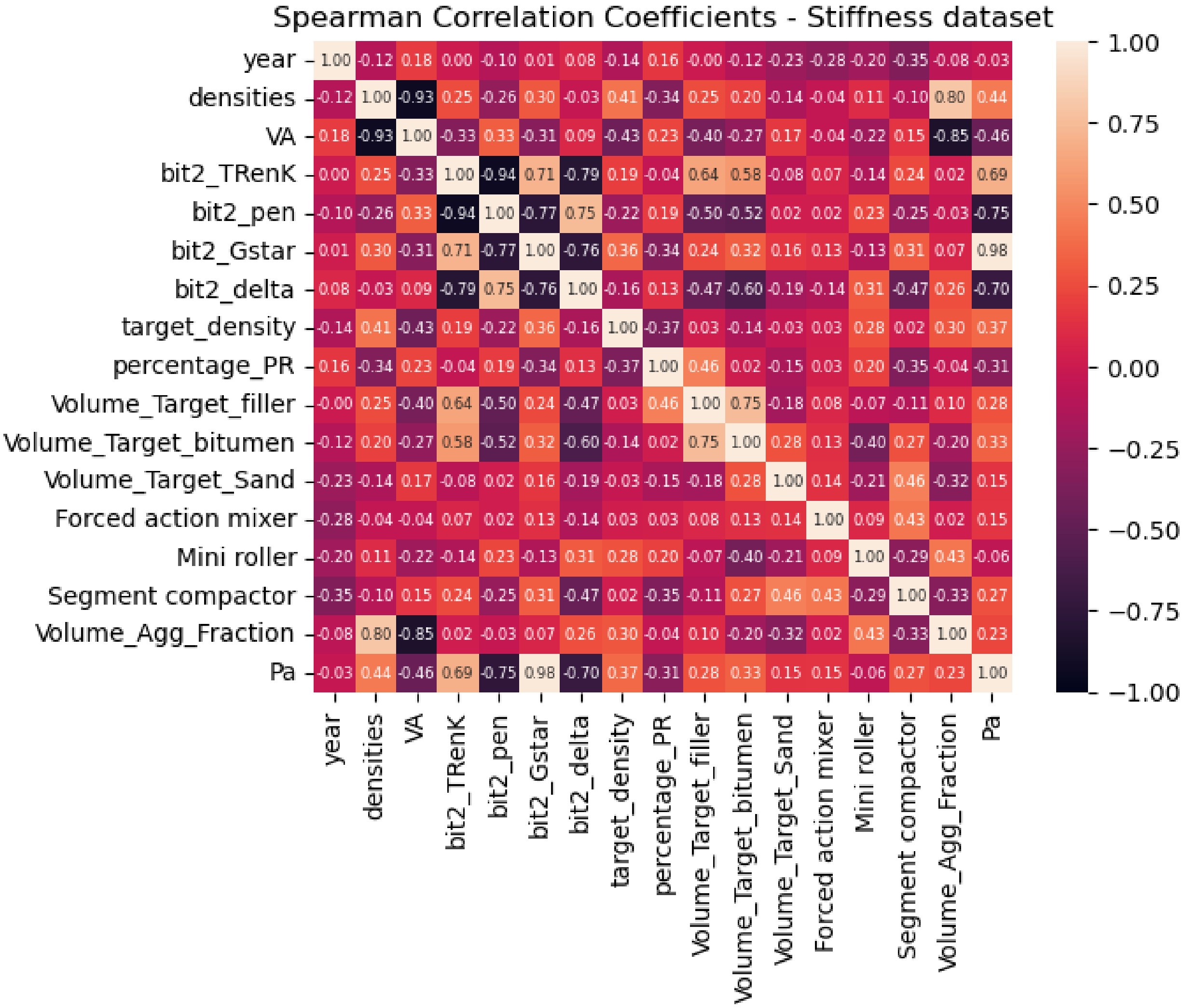


Figure G 1. Correlation analysis for the stiffness dataset considering all important features.

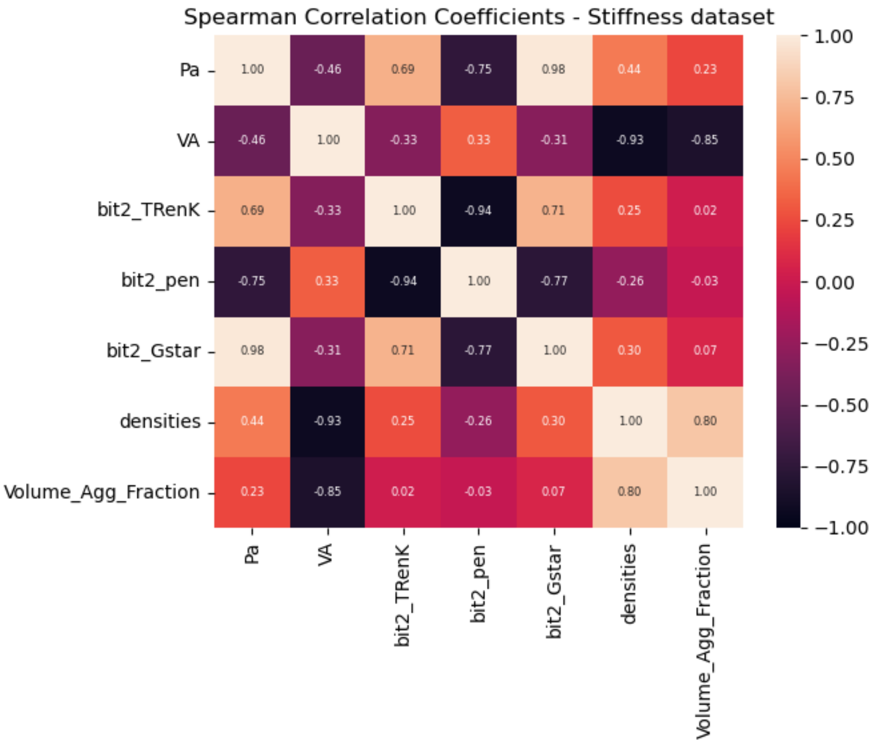


Figure G 2. Features of the stiffness dataset reporting high (|ρ| > 0.8) correlation.

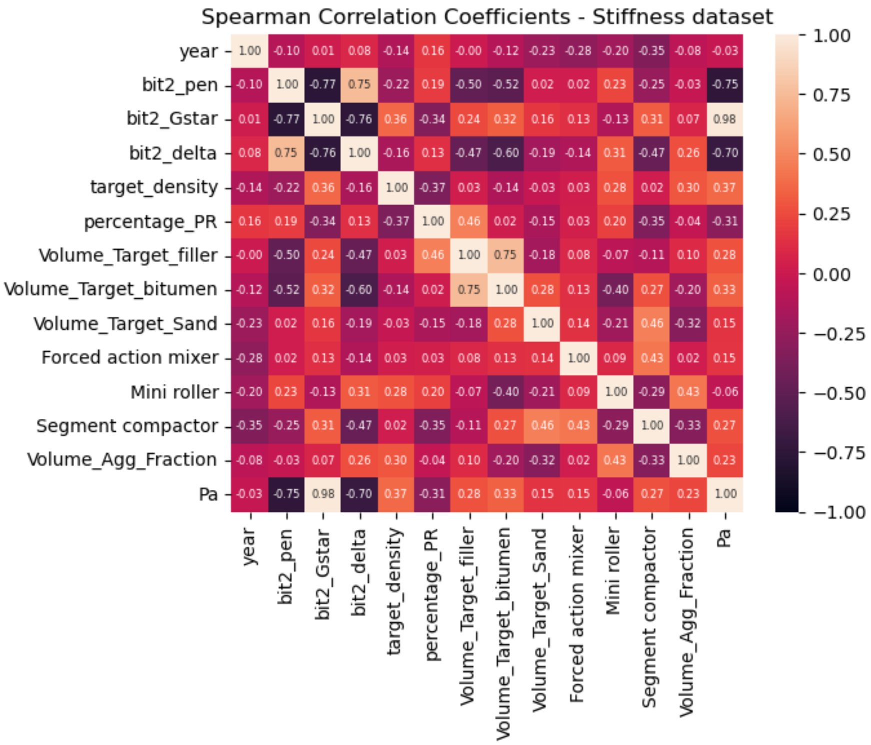


Figure G 3. Correlation analysis for the stiffness dataset after removing correlated features.

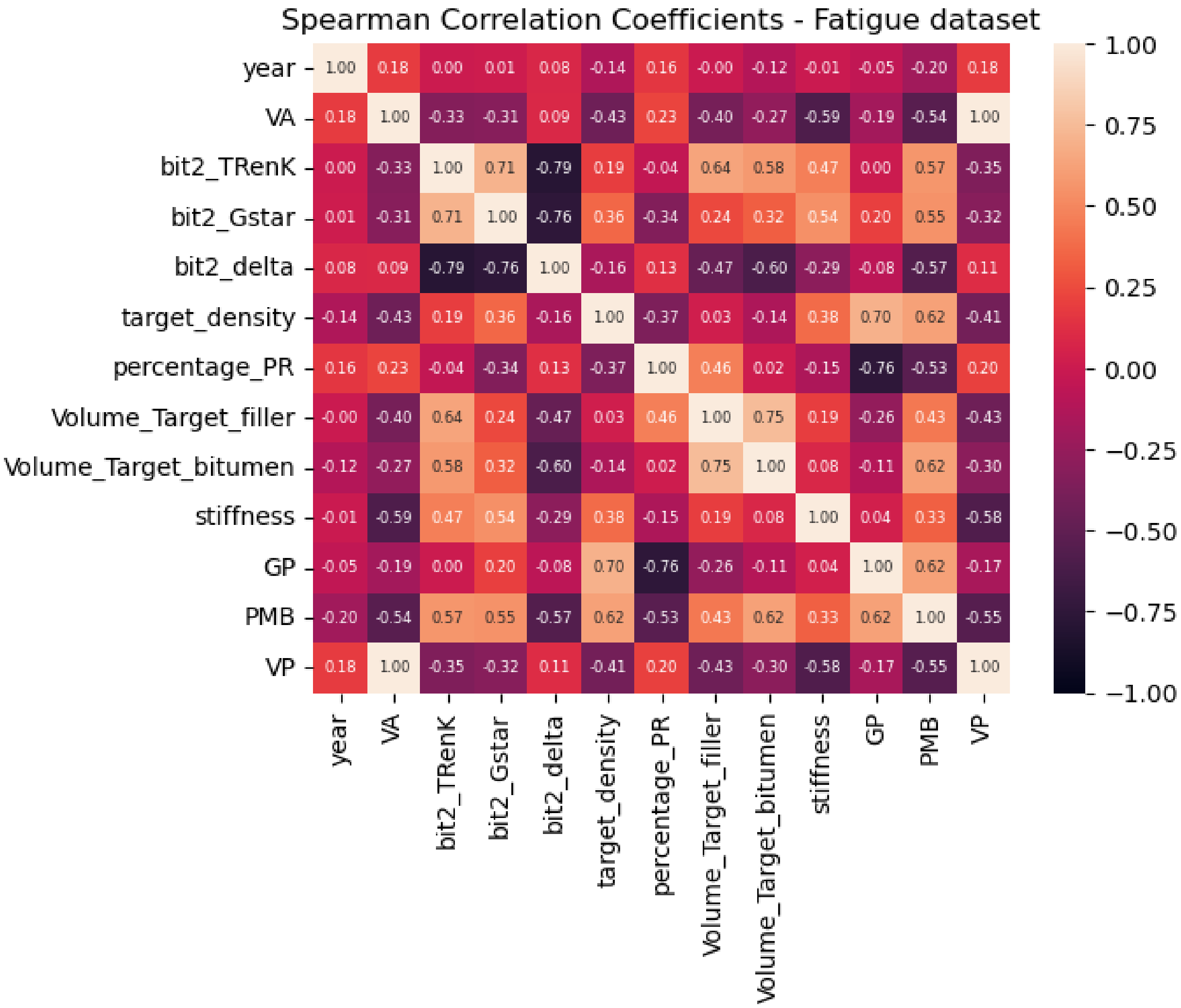


Figure G 4. Correlation analysis for the fatigue dataset considering all important features.

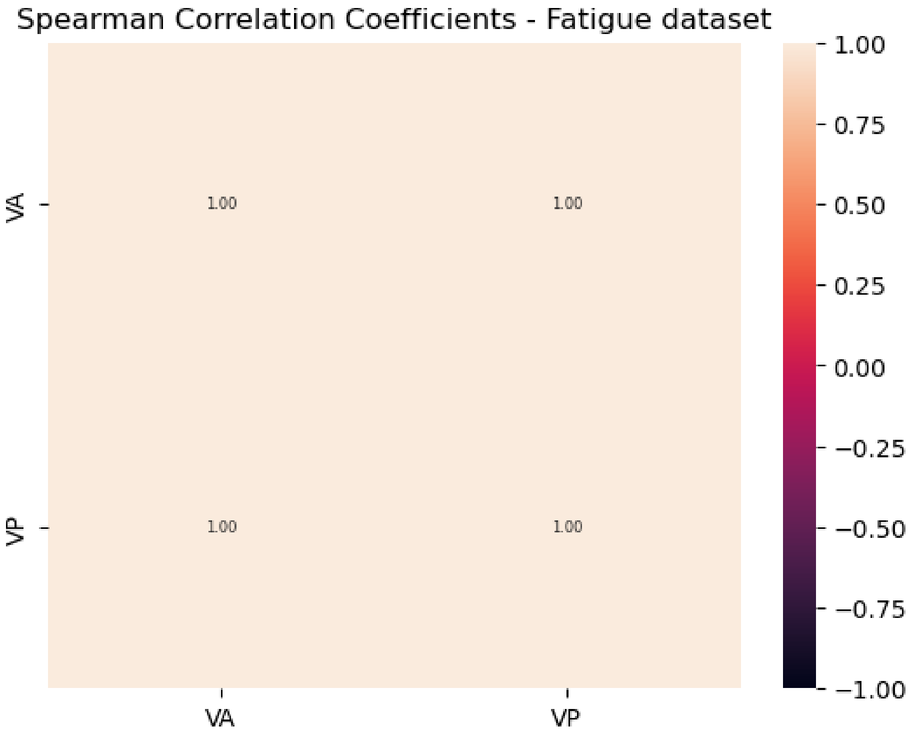


Figure G 5. Features of the fatigue dataset reporting high (|ρ| > 0.8) correlation.

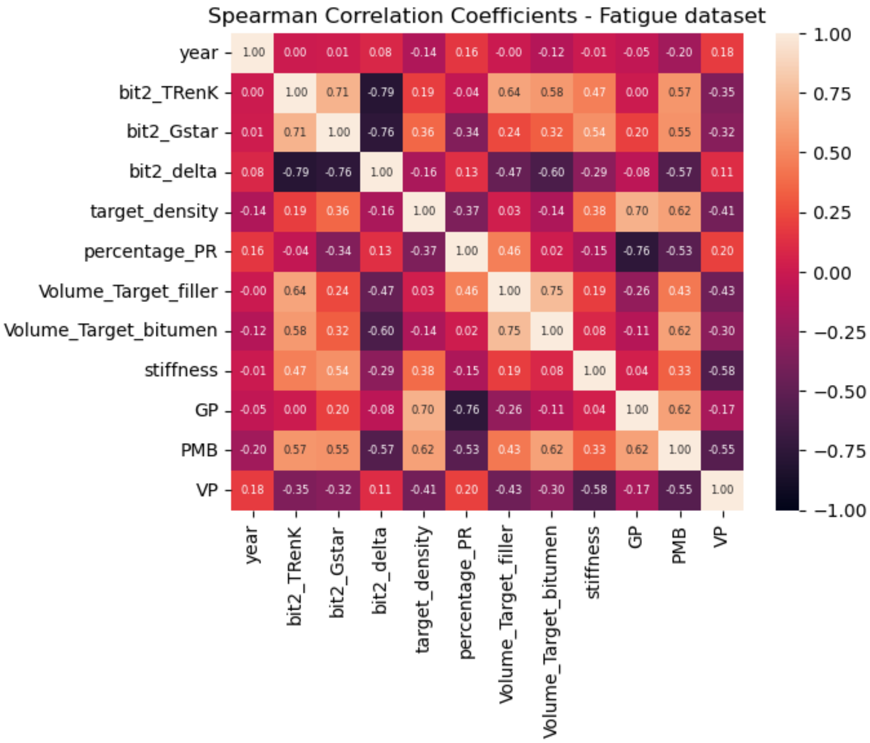


Figure G 6. Correlation analysis for the fatigue dataset after removing correlated features.