Application of Deep Learning Methodologies in Steel Wheel Design

Material Design & Processing Communications

WILEY

RESEARCH ARTICLE OPEN ACCESS

Application of Deep Learning Methodologies in Steel Wheel Design

Davide Ronco[1] | Giorgio Gallio[2] | Roberto Fontana[3] | Franco Pellerey[3]

[1]Department of Mathematical Sciences at Politecnico di Torino and Research and Innovation in MW Italia S.r.l., Rivoli, Italy | [2]Department of Research and Innovation in MW Italia S.r.l., Rivoli, Italy | [3]Department of Mathematical Sciences at Politecnico di Torino, Turin, Italy

Correspondence: Davide Ronco (davide.ronco@polito.it, d.ronco@it.mw-wheels.com)

Received: 4 October 2025 | **Revised:** 13 March 2026 | **Accepted:** 1 April 2026

Academic Editor: Pramita Mishra

Keywords: deep learning | fatigue | FEM simulation | neural networks | optimization | wheel design

ABSTRACT

Wheel design represents a complex engineering challenge, characterized by constraints related to safety and structural efficiency. This study explores the integration of deep learning (DL) techniques into the optimization process of steel wheel design, with the goal of accurately predicting stress distributions, wheel weight, and material usage based on geometric inputs. A feed-forward neural network was trained on 172 FEM-based (finite element method–based) simulations, achieving high predictive accuracy (R^2 values between 0.92 and 0.99 across all target variables). Once trained, the model is able to evaluate over 570,000 design configurations in less than 7 min, identifying optimal geometries that minimize critical stresses. Compared with conventional FEM-based workflows which typically require up to 5 days for equivalent optimization, the proposed approach drastically reduces computational time and enables rapid exploration of the design space, supporting more efficient and informed design decisions. Beyond the specific case study, the proposed workflow provides practical insights for the adoption of data-driven surrogate models in engineering applications traditionally relying on FEM-based design processes.

1 | Introduction

Structural optimization of safety–critical components remains a central challenge in modern mechanical engineering. Components such as automotive wheels are subjected to complex cyclic loading conditions and must satisfy stringent fatigue resistance requirements while meeting constraints related to weight, manufacturability, and cost.

In this context, fatigue-driven design represents a multiobjective optimization problem, where geometric parameters strongly influence stress distribution and structural durability. The accurate prediction of stress fields under service-like loading conditions is therefore essential during early design stages to ensure compliance with regulatory and performance requirements.

The finite element method (FEM) is widely adopted for fatigue assessment and structural verification due to its ability to provide detailed stress distributions. However, when embedded within iterative design loops, FEM-based optimization becomes computationally expensive. In practical engineering workflows, only one or two geometric parameters are typically varied at a time to isolate their individual effects and maintain interpretability. Although this strategy allows engineers to understand parameter influence, it significantly limits the exploration of the full multidimensional design space. Each geometric

modification requires remeshing and recalculation, leading to long development cycles when multiple design variables must be investigated simultaneously.

To overcome the computational burden associated with repeated FEM evaluations, surrogate modeling techniques have been increasingly adopted in structural optimization. Surrogate models are aimed at approximating the input–output relationship of high fidelity simulations, enabling rapid prediction of structural responses at a fraction of the computational cost.

Traditional surrogate approaches include polynomial response surfaces, radial basis functions [1] and Kriging [2, 3]. These methods have been successfully applied in various structural design problems to reduce the number of required FEM evaluations while maintaining acceptable prediction accuracy.

More recently, machine learning (ML) and deep learning (DL) techniques have gained attention as powerful data driven alternatives for surrogate modeling in computational mechanics. Neural networks, in particular, have demonstrated strong capability in capturing nonlinear relationships between geometric parameters and structural responses, especially in high-dimensional design spaces. Applications of ML-based surrogate models have been reported in structural optimization, fatigue life prediction, and materials modeling [4, 5]. In particular, Bessa et al. [4] proposed a data-driven framework for materials modeling under uncertainty, demonstrating the effectiveness of ML in approximating complex simulation based responses. Similar approaches have been explored in structural optimization and generative design contexts, where neural networks are integrated with CAD/CAE systems to accelerate evaluation and design exploration.

Compared with traditional metamodels, deep neural networks can scale more effectively with increasing dimensionality and large datasets, making them particularly attractive for multiparameter design problems where complex interactions among variables exist.

Nevertheless, most existing studies focus on generic benchmark problems or on highly flexible geometries, and limited attention has been devoted to fatigue-driven optimization of structurally constrained industrial components.

Despite the growing body of literature on ML based surrogate modeling in structural mechanics, its application to fatigue driven design of steel wheel discs remains limited. Most existing studies focus either on generic benchmark structures or on components with high geometric flexibility, such as alloy wheels, where neural networks are often employed for generative design and aesthetic structural co optimization [6].

However, steel wheels present a fundamentally different design context. Their geometry is strongly constrained by stamping process feasibility, limited geometric variability, and strict fatigue homologation requirements. Unlike alloy wheels, whose design often integrates stylistic considerations with structural optimization, steel wheel discs are predominantly governed by structural durability, manufacturability constraints, and cost efficiency.

Moreover, surrogate models developed in generic structural optimization contexts are not necessarily transferable to highly constrained industrial environments, where design parameters are limited and strongly coupled with production processes. This calls for domain specific surrogate modeling strategies that are aligned with real industrial design workflows.

To the best of the authors' knowledge, no previous work has proposed a DL-based surrogate model specifically tailored to fatigue-driven optimization of steel wheel discs using FEM-generated stress data as ground truth. This represents a relevant gap, particularly in industrial contexts where repeated FEM analyses significantly increase development time.

The present study addresses this gap by developing a multioutput feed-forward neural network capable of predicting stress levels in critical disc regions, as well as disc weight and material usage, directly from geometric parameters. The model is trained exclusively on high fidelity FEM simulations and is validated on an independent test set to assess generalization capability.

The main contributions of this work can be summarized as follows:

1. *Development of a domain-specific* DL *surrogate model* for fatigue-driven optimization of steel wheel discs, trained exclusively on high-fidelity FEM simulations.
2. *Multioutput prediction capability*, enabling the simultaneous estimation of stress levels in four critical disc regions, as well as disc weight and raw material surface, directly from geometric input parameters.
3. *Validation on an independent test set*, ensuring that the proposed model demonstrates reliable generalization within the defined design space.
4. *Industrial-scale design space exploration*, where the trained model evaluates over 570,000 geometric configurations in a few minutes, drastically reduces the computational effort compared with conventional iterative FEM workflows.
5. *Demonstration of practical integration within constrained manufacturing environments*, highlighting how data-driven surrogate models can accelerate fatigue-oriented design processes under real production constraints.

2 | Material and Methods

The wheel plays a crucial role in vehicle safety and performance, being responsible for supporting the vehicle's weight and maintaining road contact through the tire. As a mature automotive component, the wheel's design is constrained by well-established technologies and strict safety regulations, particularly concerning fatigue resistance. Fatigue failures can result from the complex combination of radial and axial loads acting cyclically during vehicle operation.

Steel wheels are subject to rigorous homologation and validation tests aimed at ensuring structural durability under real-life conditions. The most relevant fatigue tests are as follows:

- Bending fatigue test: mainly targets the evaluation of the wheel disc area; and
- Rolling fatigue test: focuses on the rim and the weld between the disc and the rim.

Figure 1 shows a scheme of wheel terminology.

In the bending fatigue test, a bending moment M_{Dc}, expressed in Nm, is applied to simulate wheel loading conditions during cornering, which typically causes failures in the disc area.

The dynamic bending moment M_{Dc} is generated by rotating an eccentric mass hinged to the main arm tip in synchronous forward whirl mode [8] (Figure 2a), and it is obtained by:

$$M_{Dc} = bm\varepsilon\omega^2 \quad (1)$$

where:

- b is the arm length (m)
- ε is the mass eccentricity (m)
- ω is the angular speed (rad/s)
- m is the value of the mass applied at the end of the arm (kg)

The arm length, (b) is calculated from the hub and the position of the mass, and the ET is the inset (in m) and it represents the distance between the hub and the middle of rim channel.

Another fatigue test is the rolling fatigue test (Figure 2b), in which the wheel is subjected to a radial force (F_{ZW} in N) and rotates freely, driven by a drum that rotates at an angular speed (ω in rad/s), simulating road conditions and often leading to failures in the rim or welds.

The key test parameters include the load to apply to the wheel (bending moment [M_{Dc}] for the bending fatigue test and radial

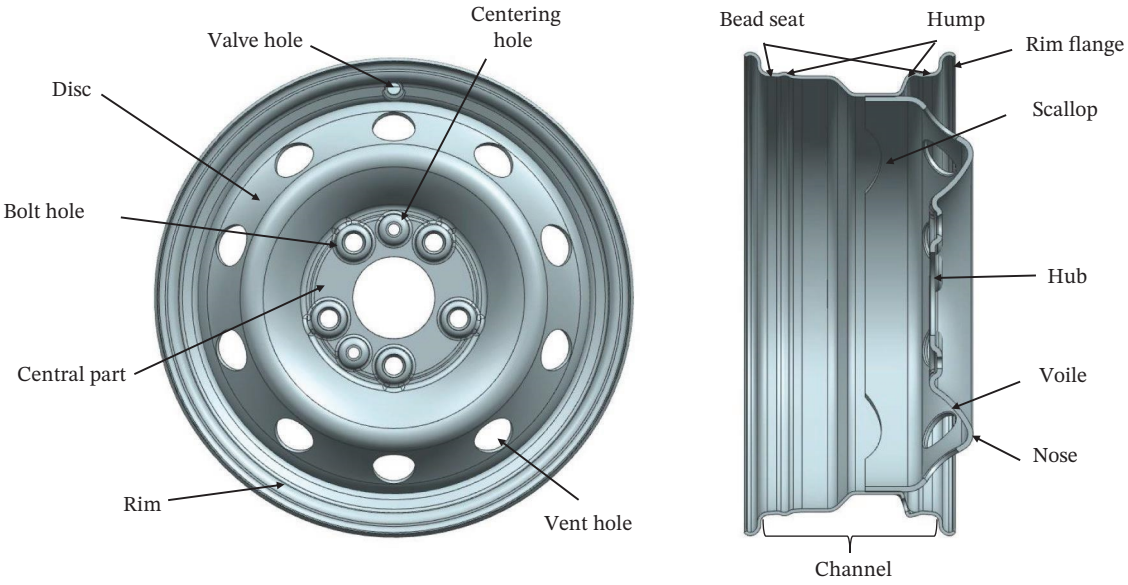

FIGURE 1 | Wheel terminology [7].

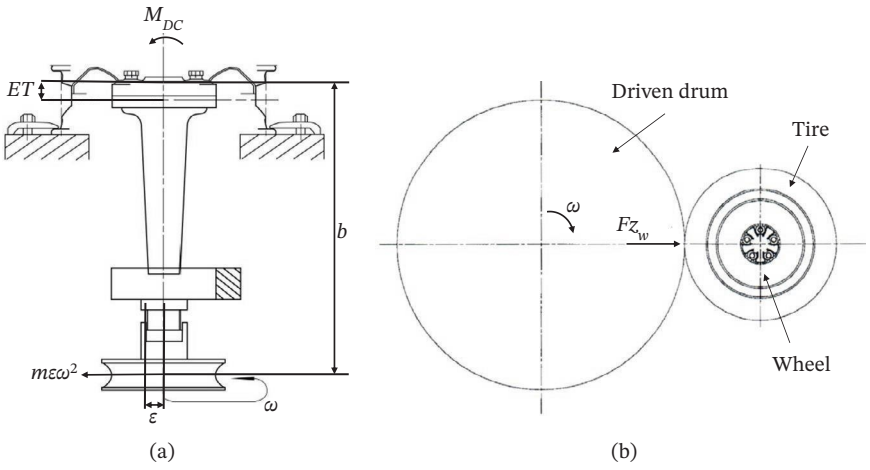

FIGURE 2 | Bending fatigue test scheme (a) and rolling fatigue test scheme (b) [7].

force [F_{ZW}] for rolling fatigue test), as well as the minimum number of cycles (Nc) required to successfully pass the test.

Figure 3, shows an example of bending fatigue test requirements.

The car manufacturers define the requirements for wheel dimensioning (based on their internal standards) by providing two reference points (blue squares), obtained from the combination of two bending moment levels ($M_{Dc}1$, high load, and $M_{Dc}2$, low load) and the corresponding number of cycles (N_C1 and N_C2).

During steel wheel validation, experimental tests are carried out for each load–cycle combination (the number of tests required for each combination also depends on the car makers' standard). The number of cycles reached at the end of the test must be higher (positioned to the right) than the requirement points (orange stars).

The test may be concluded manually, with the operator stopping the bending machine and inspecting the wheel for cracks using dye penetrant, or automatically, when the machine stops due to excessive arm oscillation caused by the loss of rigidity associated with crack formation.

Similarly, the rolling fatigue test includes one or two load–cycle combinations, and the corresponding fatigue test results must be higher than the requirement points in order for the test to be considered passed.

Failures due to fatigue can be localized depending on the type of test performed. Basing on experimental data collected by MW Italia S.r.l. during the years, the bending fatigue test predominantly stresses the disc portion of the wheel. This is the region where cracking typically initiates under the imposed bending moment, particularly near the hub interface and ventilation holes. On the other hand, the rolling fatigue test tends to induce failures in the rim and in the weld between the disc and the rim.

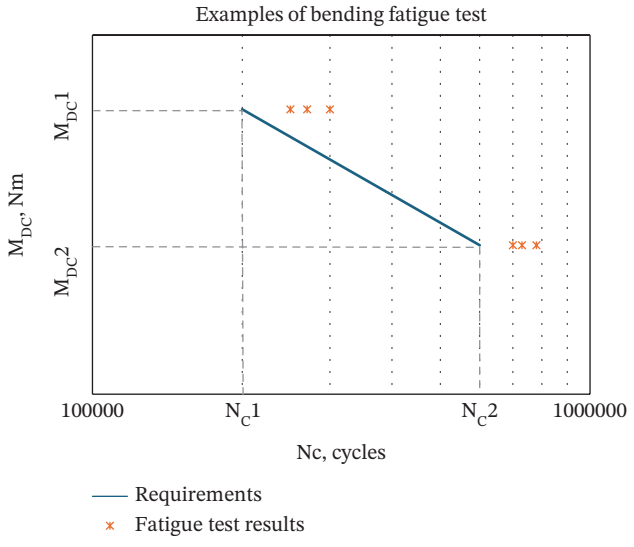

FIGURE 3 | Example of fatigue requirements.

Figure 4 shows real case examples of failure for the two types of tests.

Steel wheel design is governed by strict fatigue requirements, as well as additional constraints related to aesthetics, dynamic behavior when mounted on the vehicle, and dimensional compatibility with other components such as tires, valves, and brake systems [9].

Fatigue resistance remains the most critical requirement, as wheels are safety critical components that must endure cyclic loading over extended periods without failure.

Steel wheels are typically manufactured through stamping and welding processes, both of which require expensive, custom-made tooling. Since this tooling must be defined early in the development process and is difficult and costly to modify, design iterations are limited. As a result, steel wheels are not well suited for fast prototyping or exploratory design cycles. This limitation underscores the importance of accurate simulation and predictive modeling tools during the early stages of design.

The most commonly adopted tool for fatigue life prediction is the FEM, which provides detailed stress and strain distributions under simulated loading conditions. FEM analyses are essential for verifying compliance with fatigue requirements and identifying critical zones that could lead to failure for the respective tests [10, 11].

In Figure 5, two examples of FEM results and the relative physical test during the steel wheel homologation are shown.

The design process of a steel wheel is a fairly complex iterative process.

At the end of this iterative cycle, the wheel design is defined as an optimization of several factors.

The elements driving this iterative process are the various requirements that must be met.

In Figure 6, the optimization process scheme, described in detail in this section, is represented.

The starting point *a* (Figure 6) is the list of customer requirements, such as:

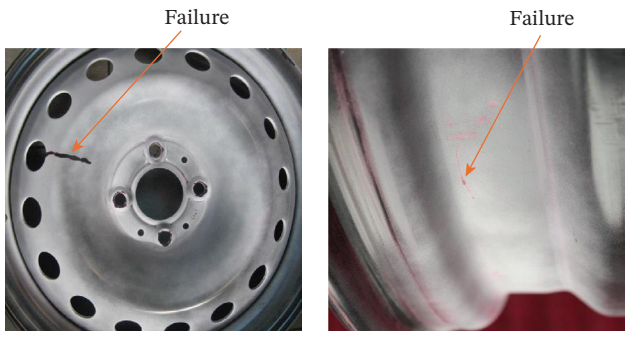

FIGURE 4 | Failure mode on bending fatigue test (a) and on rolling fatigue test (b).

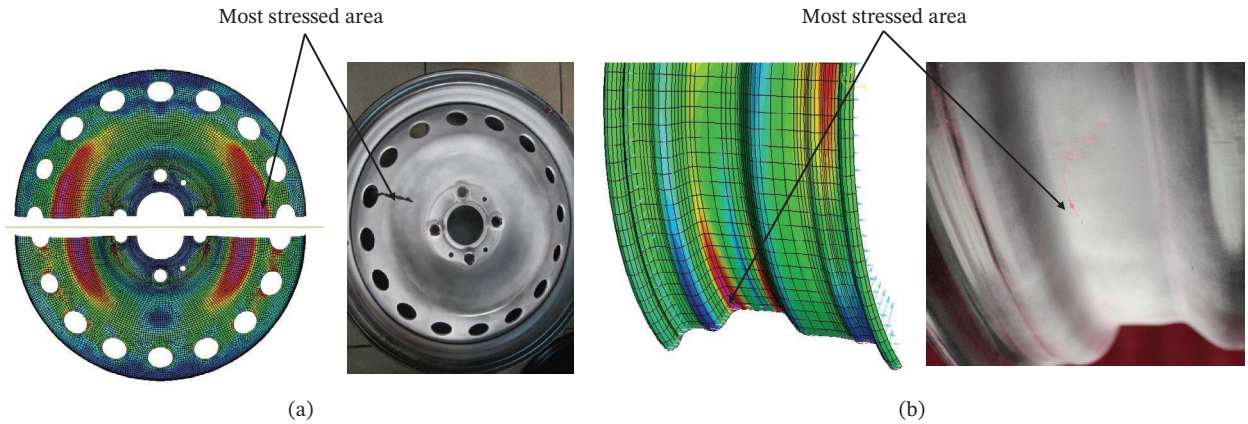

FIGURE 5 | FEM simulation and failure in bending fatigue test (a) and on rolling fatigue test (b).

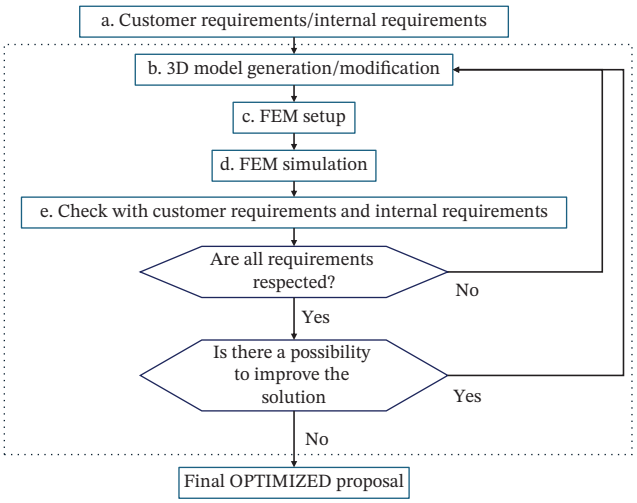

FIGURE 6 | Optimization process scheme.

- Dimensions
- Target weight
- Fatigue requirements
- Interfaces (for brakes, valves, and hubcaps)
- Dynamic requirements: NVH (noise, vibration, and harshness), dynamic stiffness, system resonance frequencies
- Impact resistance requirements
- Aesthetics
- Ventilation surface area for the braking system

In addition to these requirements, internal constraints must also be considered, such as:

- Limitations imposed by the stamping process (which restricts the possibility of creating certain shapes or very small radii)
- Constraints related to the materials used
- Production cost limits, in order to create a competitive product

Design is therefore an optimization process that is structured into five phases, repeated in a loop until the optimal design is achieved:

A: Starting point (list of customer requirements)

B: Creation of the 3D model and clearance verification (brake calipers, hubcaps, valves, etc.) in a computer-aided design (CAD) software

C: Mesh generation in the CAD environment and completion in the FEM environment of the CAD software

D: FEM calculations to verify all required specifications in the FEM software

E: Check with the customer requirements and internal design constraints

At the end of Phase *E*, if all the requirements are respected, there is the research for possible improvements. If potential improvements are identified, or if the result does not meet one or more requirements, the model is modified, and the process restarts from Phase *B*.

This design loop involves several iterative steps. Usually, only a few parameters, such as dimensions, radii, and heights are modified at a time, to avoid interference between parameter effects. Each new configuration is updated in the 3D model, considering the interfaces (e.g., brakes, hubcaps, and valves), followed by meshing and FEM setup. In industrial practice, and based on the authors' direct experience, the iterative FEM based optimization process for steel wheels (in accordance with MW Italia internal procedures) typically requires several iterations (in some cases above 50) and may take from one to several days to converge.

Pros of this process include a full understanding of how each parameter affects fatigue behavior (stress) and clear visibility of the optimization direction.

Cons include long computation time (1–5 days), limited number of parameters analyzed per iteration, difficulty achieving the best optimal solution possible, and potential human error in parameter selection.

The need for faster fatigue behavior prediction tools in the early design stages, while maintaining the existing level of accuracy, is addressed in this study, which was conducted within the framework of an Executive PhD, carried out at Politecnico di Torino (Polito), in collaboration with the manufacturer MW Italia S.r.l.

Several studies have shown that ML-based surrogate models can replace computationally expensive finite element analyses, reducing simulation time by orders of magnitude while maintaining high accuracy [12, 13].

Although several studies have successfully applied DL techniques to structural analysis and fatigue prediction in general engineering contexts [4, 5], to the best of the authors' knowledge their application to the fatigue driven design of steel wheels has remained largely unexplored, as also suggested by recent reviews on ML-based structural optimization.

Although many recent studies on alloy wheels [6] apply neural networks to improve both structural performance and aesthetic design, this approach is not directly transferable to steel wheels. Alloy wheels typically have a strong stylistic component, and AI-based optimization is often used to explore innovative shapes that balance aesthetics and strength.

In contrast, steel wheels have very limited aesthetic relevance, as they are usually covered by plastic hubcaps, and their geometry is strongly constrained by stamping process feasibility. For this reason, the present study focuses exclusively on the technical aspect of fatigue resistance rather than on style driven shape optimization. This distinction highlights the need for dedicated research tailored specifically to steel wheels, whose design challenges differ from those of alloy wheels.

The novelty of this work lies in developing a neural network capable of predicting stress, weight, and material usage directly from geometric parameters, effectively acting as a surrogate model for FEM simulations. Unlike the traditional FEM-based workflow, which requires multiple remeshing and computation cycles, the proposed DL model enables instantaneous evaluation of new geometries after training, thereby drastically reducing design time and computational cost while maintaining FEM level accuracy ($R^2 > 0.97$).

This integration demonstrates how data-driven methods can complement physics-based simulations by enabling rapid design space exploration, early-stage optimization, and decision-making support in industrial environments.

In this approach, the designer's role shifts fundamentally. Rather than manually modifying a few parameters at each iteration, the designer defines the design space boundaries at the beginning of the process. These boundaries include the minimum and maximum values for each key dimension of the wheel, considering all necessary interface constraints (e.g., brake systems, valves, and hubcaps).

Once these bounds are established, the DL model autonomously explores the full design space, evaluating hundreds of thousands of possible configurations in a short time and returning the optimal or most optimized solution according to fatigue performance criteria. The designer's task is then reduced to the final verification and validation of the proposed configuration.

Thus, unlike the standard approach where the engineer is the active agent of the optimization loop, in the DL-assisted process the engineer becomes the strategic supervisor, defining constraints and approving the solution. This paradigm shift significantly reduces lead time and increases efficiency in the design workflow.

In this work, a DL approach was integrated into the structural optimization process, with the goal of improving efficiency while maintaining the same prediction accuracy by rapidly identifying the geometries that minimize fatigue critical stresses, disc weight, and raw material usage. These predictions, which would normally require several days of iterative FEM simulations, can be obtained within seconds once the model is trained, thereby significantly accelerating the overall design workflow while maintaining high prediction accuracy.

The neural network model was developed using R [14] for data preparation and handling, and Keras [15], an open-source high-level neural network application programming interface (API)—as described by Keras and Chollet—for model definition, training, and validation.

The workflow included several key steps:

- Definition of input and output variables (e.g., geometric parameters and structural responses)
- Data normalization
- Dataset splitting into training and validation sets
- Hyperparameter tuning (e.g., number of layers, neurons per layer, activation functions)
- Model training and evaluation of training results

Once trained, the model was used for prediction of structural performance, enabling:

- Stress optimization based on the predicted responses, and
- Comparison of neural network results with numerical simulations or experimental data.

2.1 | Definition of Input and Output

The dataset used in this study is entirely simulation-based and was generated through finite element analyses performed on a parametric steel wheel model. Each dataset sample corresponds to a unique geometric configuration defined by a set of input parameters.

The input variables used in the neural network consist of the main geometric dimensions of the wheel model, such as radii, heights, and center radius positions. At this stage of the project, only a subset of eight dimensions (H10, R20, R50, D40, D50,

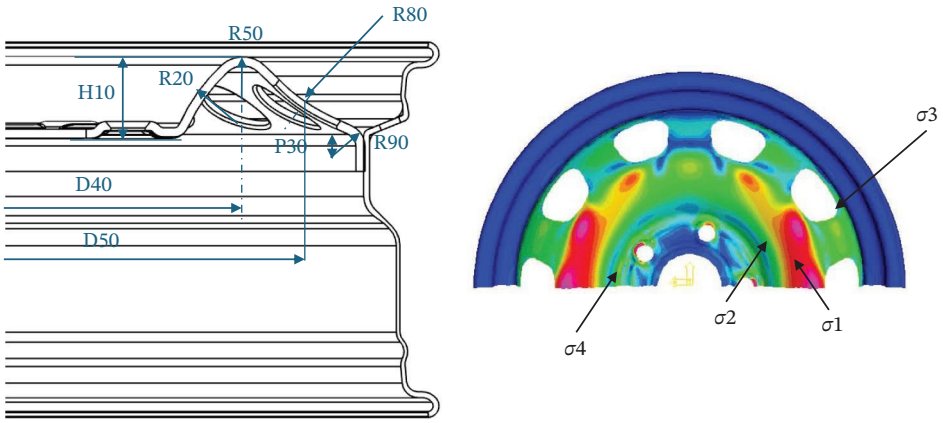

FIGURE 7 | Input parameters (a) and output parameters (b), stress in critical areas.

TABLE 1 | List of input variables.

Input variable	Name	Description	Units	Min value	Max value
H10	Nose height	Vertical distance between the central part of disc and higher part of profile	mm	30	50
R20	Voile radius	Radius of the profile horizontally interposed between the central part of the disc and the disc nose	mm	50	110
R50	Nose radius	Radius of the highest part of the disc profile (called nose)	mm	10	20
D40	Nose position	Diametral position of the center radius of the disc nose	mm	245	265
D50	Vent hole position	Diametral position of the axis of ventilation hole	mm	318	335
R80	Vent hole area radius	Radius of the profile where the ventilation holes are located	mm	−500 ("−"if convex)	+500 ("+" if concave)
R90	Disc flange radius	Radius between the vent hole area and the more external part of the disc in horizontal direction	mm	10	20
P30	Disc flange radius position	Vertical position of the center radius of the disc flange radius	mm	−4 ("−" if the center is below the central part)	+4 ("+" if the center is above the central part)

P30, R80, and R90) described in Figure 7a and in Table 1, were varied, whereas the remaining parameters were kept constant, but these variables can significantly modify the wheel shape.

The output variables predicted by the model and described in Figure 7b and in Table 2 include:

- Stress values in critical zones of the wheel: $\sigma 1$: nose (S_Nose), $\sigma 2$: voile (S_Voile), $\sigma 3$: ventilation hole area (S_VH), $\sigma 4$: central part (S_CP)
- Disc weight
- Surface area of the raw material

TABLE 2 | List of output variables.

Output variable	Name	Description	Units
$\sigma 1$	S_Nose	Stress obtained by FEM on the nose area, defined as the region vertically more external of the wheel disc	MPa
$\sigma 2$	S_Voile	Stress obtained by FEM on the voile area, defined as the region horizontally interposed between the nose area and the central part area of the wheel disc	MPa
$\sigma 3$	S_VH	Stress obtained by FEM on the vent hole area, defined as the region where al located the vent holes adopted for the braking cooling	MPa
$\sigma 4$	S_CP	Stress obtained by FEM on the central part area, defined as the central region of the wheel disc, where are located the holes to fix the wheel to the hub	MPa
Weight	Disc weight	Weight of the wheel disc calculate by CAD software	kg
Raw surface	Disc raw surface	Surface of raw material od wheel disc, without considering holes calculated by CAD software	mm^2

All data used for training were generated through high fidelity finite element simulations performed on the parametric wheel model, in accordance with the internal procedures of MW Italia S.r.l. No experimental measurements were used at this stage. For each geometric configuration, a dedicated FEM simulation was carried out to compute the stress distribution in critical regions of the wheel, as well as the disc weight and the raw material surface.

The finite element simulations used to generate the dataset were performed to reproduce a bending fatigue test, which is known to represent one of the most critical loading conditions for steel wheel discs.

The simulations were conducted using a linear elastic formulation, assuming isotropic material behavior. A reference bending moment (M_{Dc}) of 1000 Nm was applied consistently to all geometric configurations. Owing to the linear nature of the analysis, stress results corresponding to different load levels can be obtained through proportional scaling.

The procedure adopted for applying boundary conditions and loads follows an internal industrial methodology developed and routinely used by MW Italia S.r.l. Due to confidentiality constraints, detailed implementation aspects cannot be disclosed; however, the same modeling approach was consistently applied across the entire dataset to ensure coherence and comparability of results.

A two-dimensional shell model was employed, under the assumption of constant sheet thickness across the wheel disc, and the mesh was generated using shell elements with a characteristic element size of approximately 3 mm. This modeling choice represents a standard and computationally efficient approach for thin-walled steel wheel structures and is widely adopted in industrial practice.

The mechanical properties assigned to the material correspond to typical values for structural steel commonly used in wheel design at MW Italia S.r.l. The adopted FEM modeling strategy is part of the company's established design workflow and has been internally validated for structural assessment of wheel discs.

The FEM dataset provides fully deterministic and noise-free structural responses, which is particularly suitable for training supervised learning models when the dataset size is limited.

Due to industrial confidentiality constraints, the dataset cannot be made publicly available.

The nomenclature adopted in Tables 1 and 2 follows the internal design conventions used at MW Italia S.r.l. For clarity and to facilitate comprehension for readers unfamiliar with this terminology, a concise description of each geometric variable has been included. The model was trained using a supervised learning approach, where known geometric inputs were mapped to corresponding output variables (stress, weight, and surface area). This allowed the neural network to learn from a labeled dataset and generalize predictions on new geometries [16].

This input–output structure allows the neural network to learn the relationship between design parameters and structural performance, facilitating prediction and optimization processes.

2.2 | Data Normalization

All input and output variables were normalized to the [0, 1] range using min–max normalization [17], according to Equation (2):

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

This approach helps to improve numerical stability and accelerates convergence of the model during the training phase by reducing scale disparities among variables. In gradient-based optimization, large differences in feature magnitudes can lead to poorly conditioned loss landscapes and inefficient weight updates. By mapping all variables to a common scale, the optimization process becomes more stable and allows for more effective gradient descent steps. Due to the large-scale differences among variables (e.g., input ranges such as R50: 10–30 mm vs. R80: 130–1000 mm), normalization ensures a consistent scale, which is crucial for stable and efficient model training.

Although standardization is more commonly discussed in the literature [18], min–max normalization serves a similar purpose by mapping all features to a common scale.

2.3 | Training and Validation Dataset Split

The full dataset currently includes 172 samples and this number looks adequate for the present case study. The analysis focuses on a single 17″ steel wheel, whose geometric variability is limited and fully described by a small number of input parameters. These parameters are sufficient to define the entire disc geometry, which results in a low dimensional and highly constrained design space.

In addition, the dataset is generated entirely through deterministic FEM simulations, ensuring high repeatability and an absence of measurement noise. For each input parameter, the sampling strategy was defined so as to uniformly cover its feasible dimensional range, avoiding clustering near a single extreme and ensuring good representation of the design space.

Furthermore, the model exhibits excellent predictive performance, with R^2 values above 0.97 and low MSE (mean square error) across all normalized targets, confirming that the available dataset is sufficient for learning the underlying geometric–structural relationships of this case study.

The dataset was randomly split into 80% for training and 20% for independent testing [19]. During training, a validation subset corresponding to 10% of the training portion (i.e., 8% of the full dataset) was automatically generated using the "validation_split" option in Keras. This validation set was used exclusively to monitor convergence, whereas the 20% test set was never used during training or hyperparameter tuning and was employed only for the final assessment of model generalization.

This fixed ratio was chosen due to the relatively small size of the dataset. The random selection ensures that both subsets are representative of the overall data distribution, allowing for a reliable performance evaluation during the model development process.

It should be noted that the objective of the present study is not to benchmark different ML algorithms, but to assess the feasibility and effectiveness of a DL-based surrogate model for FEM-driven steel wheel design. For this reason, model performance is evaluated in terms of agreement with FEM results and generalization within the defined design space.

To address generalization, the dataset was consistently divided into training, validation, and independent test subsets. As already said, the test set was not used during model training or hyperparameter tuning and serves to evaluate predictive performance on unseen geometries.

In this context, the main advantage of the proposed approach over traditional FEM-based workflows lies in computational efficiency. Once trained, the surrogate model enables near instantaneous prediction of stresses, weight, and material usage for new geometries, whereas a full FEM simulation must be repeated for each design iteration.

2.4 | Hyperparameter Optimization

A full factorial design (FFD) was implemented directly in the code to identify the optimal configuration of hyperparameters [20].

Each configuration was evaluated over 100 epochs, and the one with the lowest MSE was selected as optimal.

The hyperparameters considered in this study are as follows:

2.4.1 | Number of Neurons per Layer

This defines the number of neurons for each hidden layer (i.e., the neural network layers placed between the input and output).

For each layer, a minimum and maximum number of neurons was defined, and for the first layer also an intermediate value was included.

Considering that the neural networks have four hidden layers, the number of neurons in each layer was selected from a discrete set of candidate values:

- 1st hidden layer: 16, 32, and 64 neurons
- 2nd hidden layer: 16 and32 neurons
- 3rd hidden layer: 8 and 16 neurons
- 4th hidden layer: 4 and 8 neurons

2.4.2 | Batch Size

The batch size determines the number of data samples used to update the weights during a single training iteration.

Three approaches are distinguished:

- Full batch gradient descent: more stable but converges slowly (weights updated once per epoch).
- Stochastic gradient descent (SGD): faster but less stable (weights updated at every sample).
- Minibatch gradient descent: balance between stability and speed. Three minibatch sizes, based on powers of 2, were tested: 4, 8, and 16.

2.4.3 | Learning Rate

A key parameter that determines how quickly or slowly the model updates the weights in response to the error during training.

The update rule is defined as follows:

$$\theta := \theta - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L\left(x^{(i)}, y^{(i)}; \theta\right) \tag{3}$$

where:

- θ: vector of model parameters (weights and biases)
- η: learning rate
- m: number of samples in the batch
- $L\left(x^{(i)}, y^{(i)}; \theta\right)$: loss function computed on input $x^{(i)}$ and target $y^{(i)}$
- $\nabla_\theta L$: gradient of the loss function with respect to θ

This equation describes the fact that at each training step, the model parameters θ are updated by moving in the direction opposite to the gradient (to minimize the loss), scaled by the learning rate η.

If η is too large, the model may overshoot the minimum and fail to converge (or even diverge).

If η is too small, training will be very slow and may get stuck in a local minimum.

Two levels of learning rate were selected: 0.001 and 0.005.

2.4.4 | Activation Function

Among the evaluated architectural choices, two types of activation functions were considered for the hidden layers: ReLU (rectified linear unit) and Tanh [21].

- ReLU (Equation (4) a) is widely used due to its simplicity and efficiency. It introduces nonlinearity by zeroing out negative inputs while keeping positive values unchanged. It allows for sparse representations and faster convergence.
- Tanh (Equation (4) b) outputs values between −1 and 1 and is symmetric around the origin, making it useful for data normalized around zero. However, it may suffer from vanishing gradients in deep networks.

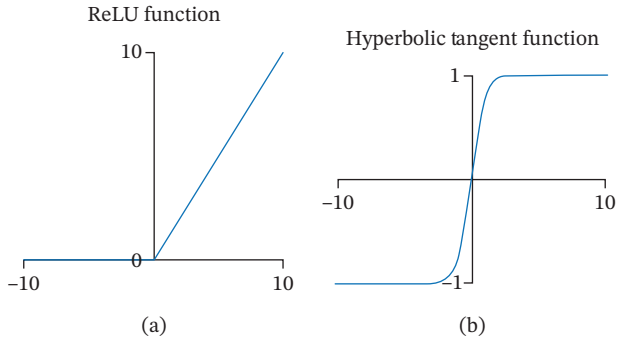

FIGURE 8 | Activation function tested ReLU (a) and Tanh (b).

TABLE 3 | Hyperparameter search space explored in full factorial design.

Hyperparameter	Values tested
Neurons in 1st hidden layer	[16, 32, 64]
Neurons in 2nd hidden layer	[16, 32]
Neurons in 3rd hidden layer	[8, 16]
Neurons in 4th hidden layer	[4, 8]
Batch size	[4, 8, 16]
Learning rate	[0.001, 0.005]
Activation function	[ReLU. Tanh]

Figure 8 shows the mathematical definitions and visual comparison of the two functions.

Table 3 reports the complete set of hyperparameters analyzed through the FFD, whereas the optimal configuration identified is reported later in the article.

2.4.5 | Hidden Layer

Another key architectural hyperparameter that was investigated is the depth of the neural network. The number of hidden layers directly influences the model's capacity to capture nonlinear relationships between geometric parameters and fatigue performance indicators.

Although increasing the number of layers can theoretically improve model accuracy, in practice it significantly increases computational cost and the time required for hyperparameter tuning.

TABLE 4 | Network architectures evaluated through full factorial design.

Test	Hidden layer	Number of neurons
1	Neurons in 1st hidden layer	[16, 32, 64]
	Neurons in 2nd hidden layer	[16, 32]
	Neurons in 3rd Hidden layer	[8, 16]
	Neurons in 4th hidden layer	[4, 8]
2	Neurons in 1st hidden layer	[16, 32]
	Neurons in 2nd hidden layer	[16, 32, 64]
	Neurons in 3rd hidden layer	[16, 32]
	Neurons in 4th hidden layer	[8, 16]
	Neurons in 5th hidden layer	[4, 8]
3	Neurons in 1st hidden layer	[16, 32]
	Neurons in 2nd hidden layer	[16, 32, 64]
	Neurons in 3rd hidden layer	[16, 32, 64]
	Neurons in 4th hidden layer	[16, 32]
	Neurons in 5th hidden layer	[8, 16]
	Neurons in 6th hidden layer	[4, 8]

To assess the effect of model depth, a FFD [22] was performed not only for architectures with four hidden layers but also extended to configurations with five and six layers. For each depth level, the same hyperparameter search space (neurons per layer, activation function, learning rate, and batch size) was systematically explored.

In Table 4 the three architectures are shown evaluated through FFD.

The results of this comparative analysis are summarized in Table 5. Increasing the number of hidden layers beyond four did not yield improvements in accuracy: Both the MSE and mean absolute error (MAE) values did not decrease with increasing the number of layers, and the R^2 values decreased across all output variables. Moreover, the computational cost increased significantly, with the FFD for six layers requiring up to 180 min of optimization.

2.4.6 | Epochs

Epochs indicate how many times the algorithm goes through the entire training set. The number must be properly chosen to avoid:

- Overfitting: The model learns the training data too well but fails to generalize to new data. In this case, the validation loss increases, whereas the training loss decreases.
- Underfitting: the model does not have enough time to learn, resulting in high errors for both training and validation sets.

Figure 9 [23] shows typical loss curves for underfitting, overfitting, and ideal training.

In this study, the number of epochs was selected based on preliminary experiments performed across multiple network configurations (four, five, and six hidden layers). These tests showed that the loss curves tended to stabilize well before 500 epochs, suggesting that the model had already learned the main data patterns.

Figure 10 illustrates the evolution of the training and validation MSE for the final model. The training loss decreases rapidly during the initial epochs, whereas the validation loss follows a similar trend, confirming that learning is effective. After approximately 80–100 epochs, the validation curve reaches a plateau and remains close to the training curve, indicating that the model has converged without showing signs of overfitting.

The absence of divergence between the two curves throughout training confirms that the model generalizes well to unseen data.

Based on these observations, 500 epochs were selected to ensure complete convergence while allowing continuous monitoring of validation behavior. This value provided a robust margin beyond the stabilization point without introducing unnecessary computational overhead.

2.4.7 | Optimizer

An optimizer is an algorithm that adjusts the weights and biases of a neural network during training to minimize the loss function, that is, the error between the predicted values and the true values.

In this work, the Adam (adaptive moment estimation) optimizer [24] was used. Adam is an advanced optimizer based on gradient descent. It has the advantage of automatically adapting the learning rate, resulting in faster and more stable training, even in the presence of noisy or complex data.

2.5 | Neural Network Definition

To justify the selection of the final network architecture, several configurations with four, five, and six hidden layers (described in Table 4) were evaluated through FFD. Table 5 reports the

TABLE 5 | Summary of performances and training time for different numbers of hidden layers.

Hyperparameters	Test 1	Test 2	Title 3
Number of hidden layer	4	5	6
MSE	0.0028	0.0043	0.0051
MAE	0.0312	0.0436	0.0524
	Training–validation	Training–validation	Training–validation
R^2 nose	0.9789–0.9733	0.9759–0.9317	0.9452–0.8867
R^2 voile	0.9760–0.9217	0.9670–0.8741	0.8839–0.7807
R^2 vent hole	0.9816–0.9605	0.9769–0.9091	0.9305–0.9177
R^2 central part	0.9932–0.8358	0.9832–0.8281	0.9088–0.8618
R^2 weight	0.9986–0.9976	0.9969–0.9872	0.9921–0.9801
R^2 surface	0.9981–0.9972	0.9969–0.9879	0.9865–0.9761
Time	30 min	40 min	180 min

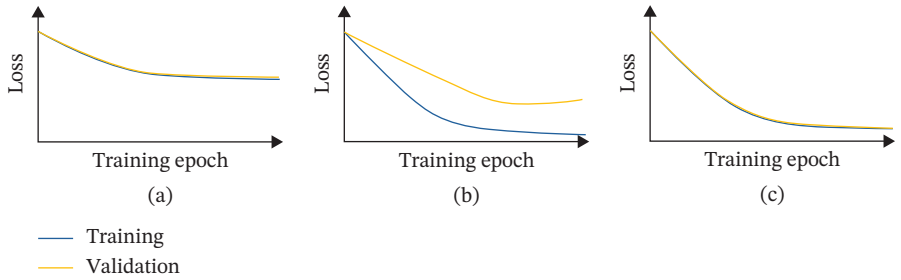

FIGURE 9 | A schematic diagram showing the loss curves from simplified cases of underfitting (a), overfitting (b), and just-right fitting (c) in model training.

TABLE 6 | Summary of neural network architecture.

Hyperparameter	Value
Input layer	Eight neurons ➔ input values
1st hidden layer	64 neurons
2nd hidden layer	32 neurons
3rd hidden layer	16 neurons
4th hidden layer	8 neurons
Output layer	Six neurons ➔ output values
Learning rate	0.005
Batch size	4
Activation function	ReLU
Epochs	500
Optimizer	Adam

performance metrics of the best model obtained for each architectural depth. These results clearly show that the four-layer model achieves the best trade-off between accuracy and computational efficiency, and therefore it was selected as the optimal architecture.

The final neural network is described in Table 6 and it is referred to as the Test 1 in Table 5.

This architecture was chosen as the best performing configuration based on the lowest validation MSE. It reflects a progressive dimensionality reduction from input to output layers, which helps the network learn complex nonlinear mappings efficiently.

A schematic representation of the final neural network architecture is reported in Figure 10. The model is a fully connected feed-forward network, with one input layer receiving the geometric parameters of the wheel disc, a sequence of hidden layers (defined according to the optimal hyperparameters summarized in Table 6) using nonlinear activation functions, and an output layer providing the predicted fatigue critical stresses, disc weight, and raw material surface area. This diagram summarizes the number of layers, neurons per layer, and activation functions, and is intended to facilitate reproducibility of the proposed approach.

2.6 | Training Process

Once the architecture of the neural network was defined, the training phase was conducted using supervised learning. In this phase, weights and biases are iteratively updated to minimize the MSE loss function (L), defined as follows:

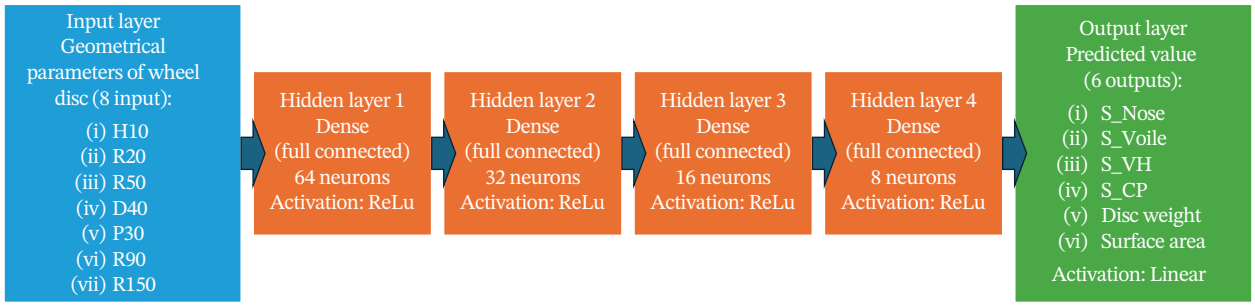

FIGURE 10 | Schematic architecture of the neural network used in this study, showing the input layer (geometric parameters), hidden layers (fully connected with nonlinear activation functions), and output layer (fatigue-critical stresses, disc weight, and raw-material surface area).

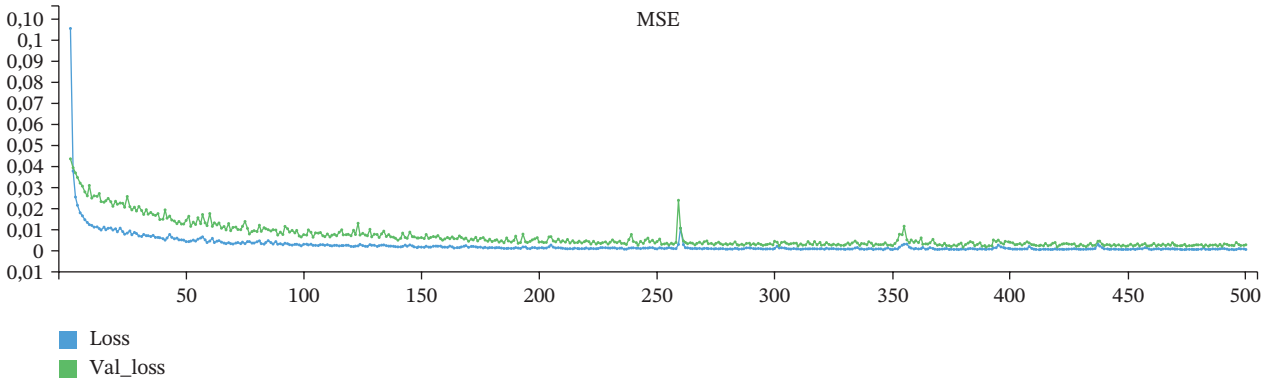

FIGURE 11 | MSE loss function: train data (blue) and validation data (green).

TABLE 7 | Coefficient of determination R^2 for each output (stress in critical area, disc weight, and area of raw material).

Output variable	R^2training data	R^2test data
Stress on nose ($\sigma 1$)	0.9789	0.9733
Stress on voile ($\sigma 2$)	0.9760	0.9217
Stress on vent hole ($\sigma 3$)	0.9816	0.9605
Stress on central part ($\sigma 4$)	0.9932	0.8358
Disc weight	0.9986	0.9976
Raw surface	0.9981	0.9972

$$L := \frac{1}{n}\sum_{i=1}^{n}\left(y_{i,real} - y_{i,pred}\right)^2 \tag{4}$$

During each training epoch, the minibatch gradient descent algorithm computes the error gradient with respect to the network weights using small data subsets (minibatches). The weights are then updated through backpropagation, propagating the error backward across the network layers.

The weight update rule is given by the formula in Equation (3).

This iterative process continues for the defined number of epochs until convergence criteria are met, or the loss function stabilizes as it is visible in Figure 11.

3 | Results and Discussion

To evaluate the performance of the trained DL model, the predicted stress values were compared with those obtained through traditional FEM simulations for each critical area of the disc wheel, the disc weight, and the surface area of the raw material, as described in section 2.

For each region, the coefficient of determination R^2 was computed to quantify the model's accuracy in predicting stress values. An R^2 value closer to 1.0 represents a better fit. Comparison between predicted and actual values is visible in Table 7.

In addition to the training results (80% of dataset), the generalization capability of the model was evaluated on the independent 20% test set, which was never used during training or hyperparameter tuning.

The scatter plots in Figure 12 illustrate the relationship between predicted and actual values of stress for the four analyzed areas (nose, voile, vent hole, central part), disc weight and the raw surface using the test set. For each graph, the ideal 45° reference line (perfect prediction) is visible.

The results confirm that the trained model can reliably approximate FEM stress outputs across the relevant design space, supporting its application as a surrogate model in preliminary design optimization.

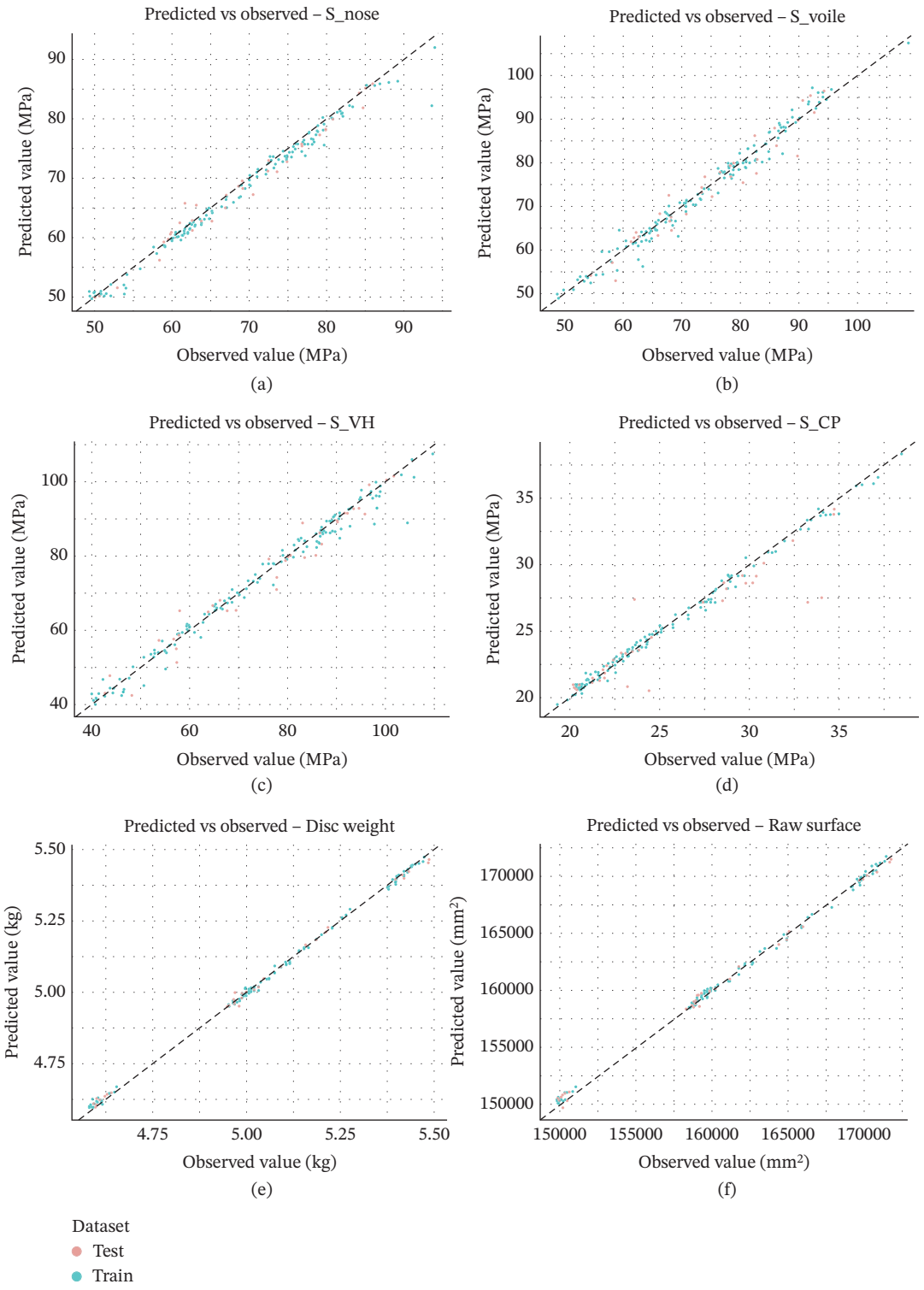

FIGURE 12 | Scatter plots with relationship between observed values and predicted values for the six different outputs: stress on nose (a), stress on voile (b), stress on vent holes area (c), stress on central part (d), disc weight (e), and raw surface (f).

3.1 | Prevision and Wheel Optimization

Once the neural network model was trained, it is possible to use it to predict the stress state of the wheel disc, its weight, and the required material quantity for manufacturing. A parameter sweep was performed over eight geometric input variables (Figure 7a), each varied between its minimum and maximum allowable values using a predefined step size, defined as the incremental interval applied to each parameter (see Table 8). This procedure generated a total of 571,725 geometric combinations.

For each configuration, the trained model predicted the corresponding stress values, disc weight, and material surface. This approach enabled an exhaustive exploration of the design space

at negligible computational cost. In just over 6 min, the algorithm identified the Top 4 optimal configurations (Figure 13), which minimized the stress values in the four critical areas of the disc (S_Nose, S_Voile, S_VH, and S_CP).

To validate the model, the selected configurations were simulated using FEM with the corresponding geometric parameters. A strong similarity was observed between the predicted and simulated values.

In Table 9, it shows the four optimized configurations with the value of stress obtained by FEM, the value of weight and raw surface obtained by CAD, and the relative values predicted by neural network.

TABLE 8 | List of input parameters and their range, for a total of 571,725 combinations.

Parameter	Min value (mm)	Max value (mm)	Step (mm)
R20	85	110	5
R50	10	20	1
D40	245	265	5
H10	30	50	1
R80	150	500	50
D50	325	335	1
R90	10	20	2
P30	−4	4	2

These results confirm the high accuracy and reliability of the DL model and highlight its potential for integration into the design optimization workflow, significantly reducing the number of FEM simulations required during early-stage design exploration.

The coefficient of determination R^2 was calculated for the value of stress, together with the Pearson coefficient [25], and the values are very high: R^2 stress is 0.983 and Pearson coefficient is 0.992. It was also calculated the Pearson coefficient for disc weight (0.992) and for the raw surface (0.994).

In the scatter plot in Figure 14, it shows the correlation between the stress value predicted by neural network and the stress obtained by FEM.

3.2 | Impact of Training Dataset Size on Model Performance

To assess the model's robustness with respect to data availability, an additional analysis was conducted by training the network with decreasing portions of the original dataset. Importantly, in all cases the same data partitioning strategy was maintained: 80% of each reduced dataset was used for training (with 10% internal validation), whereas the remaining 20% was kept as an independent test set and was never used during training or hyperparameter tuning. Specifically, the R^2 values were compared for three scenarios:

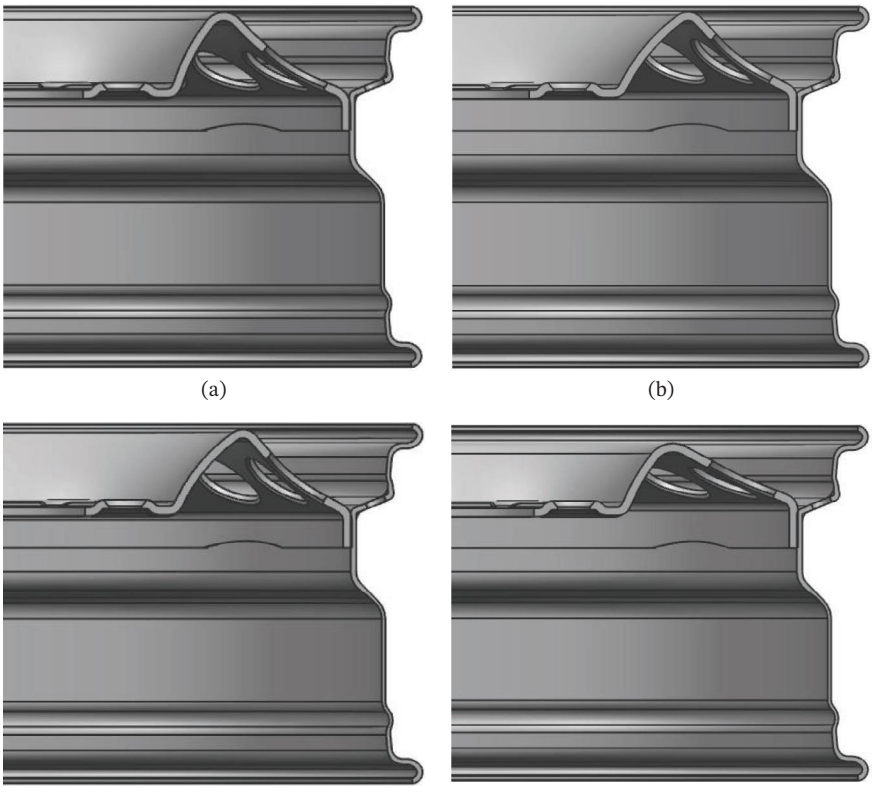

FIGURE 13 | The optimal configurations that minimize the stress on nose (a), the stress on voile (b), the stress on ventilation hole area (c), and the stress on central part (d).

TABLE 9 | Comparison between the value obtained by neural network and by FEM/CAD.

R20 (mm)	R50 (mm)	D40 (mm)	H10 (mm)	R80 (mm)	D50 (mm)	R90 (mm)	P30 9mm)	Method	S_Nose (MPa)	S_Voile (MPa)	S_VH (MPa)	S_CP (MPa)	Weight (kg)	Raw surface (mm^2)
109	12	245	50	150	328	12	−4	NN	48.66	53.23	43.37	24.35	5.37	169,020.0
								FEM/CAD	50.39	55.38	45.39	26.12	5.39	169,513.2
109	10	265	50	150	328	10	0	NN	48.96	48.25	42.67	24.83	5.39	169,584.3
								FEM/CAD	51.01	48.87	40.67	24.68	5.46	171,361.9
99	10	265	50	150	328	10	−2	NN	49.47	48.88	40.63	25.75	5.39	169,561.9
								FEM/CAD	50.31	47.95	39.84	25.17	5.42	170,206.6
109	16	245	43	500	328	10	4	NN	67.12	74.39	71.66	19.28	5.17	164,065.2
								FEM/CAD	70.32	82.27	73.89	24.53	5.09	161,723.4

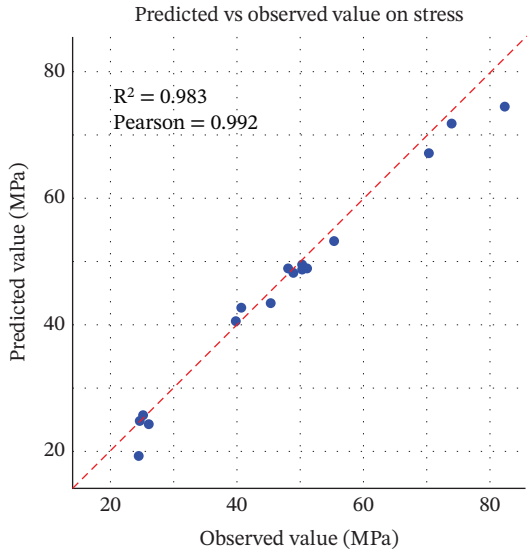

FIGURE 14 | Predicted stress by neural network versus observed value obtained by FEM.

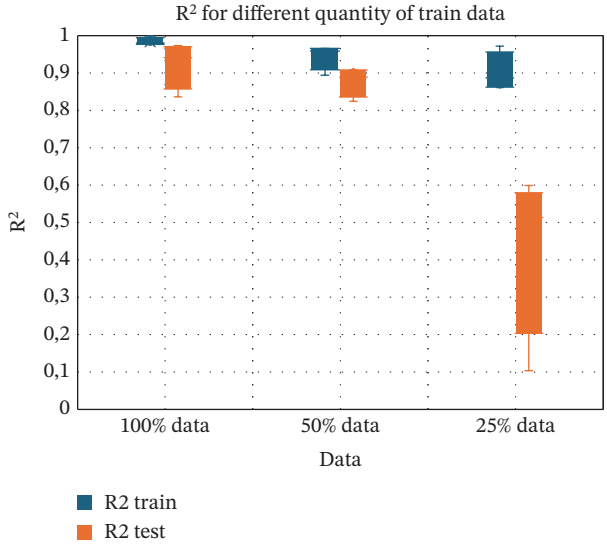

FIGURE 15 | Performance trend variating the quantity of training dataset size.

- 100% of the dataset (172 observations)
- 50% of the dataset (86 observations)
- 25% of the dataset (43 observations)

As shown in Figure 15, reducing the size of the training dataset results in a clear performance degradation, particularly on the test set. Although the model retains good generalization ability with 50% of the data, its predictive power drops significantly when trained on only 25% of the original dataset. In this latter case, test R^2 values show large variability and a marked reduction in accuracy.

This analysis highlights the importance of having a sufficiently large and representative dataset to ensure reliable and stable predictions when applying DL to engineering design problems.

3.3 | Comparison With Existing Surrogate Modeling Approaches

The performance of the proposed DL surrogate model can be contextualized within the broader literature on surrogate modeling in structural optimization. Surrogate models are widely employed to reduce the computational burden associated with high-fidelity numerical simulations while preserving acceptable prediction accuracy. Comprehensive reviews highlight the increasing adoption of surrogate-based approaches in engineering design and optimization problems, particularly where repeated FEM evaluations are computationally expensive [2].

Classical metamodeling techniques including response surface methods, radial basis functions, and Kriging models have long been used to approximate structural responses in design optimization problems. These approaches have demonstrated reliable performance in moderate dimensional parametric studies, especially when the number of simulations is limited and computational cost is a critical factor.

More recently, ML-based surrogate models, particularly artificial neural networks (ANN), have gained increasing attention in structural mechanics applications. Neural network surrogates trained on finite element simulation data have been successfully applied to fatigue related prediction and structural response estimation problems, often reporting coefficients of determination (R^2) above 0.90 depending on problem complexity and dataset size [26].

Within this context, the R^2 values achieved in the present study (0.92–0.99 across all predicted outputs) fall within, and in several cases approach, the range of predictive accuracy reported in surrogate modeling studies for structural applications where R^2 values above 0.90 are typically considered indicative of reliable surrogate performance [27]. The dataset size adopted here (172 FEM simulations) is consistent with industrial parametric optimization scenarios, where each high-fidelity simulation entails significant computational cost and data generation must remain practically feasible.

In terms of computational efficiency, the trained neural network evaluated over 570,000 geometric configurations in less than 7 min. In contrast, an equivalent iterative FEM-based workflow would require several days of computation. This order of magnitude reduction in evaluation time aligns with the primary objective of surrogate modeling strategies and demonstrates the practical effectiveness of the proposed approach.

It is also important to note that many existing surrogate modeling studies focus on academic benchmark structures or geometrically flexible components. In contrast, the present work addresses a structurally constrained industrial component governed by fatigue requirements and manufacturing limitations. This highlights the applicability of DL-based surrogate models not only in exploratory design contexts but also in tightly constrained, production-driven optimization environments.

4 | Conclusion and Next Steps

This initial study demonstrates that a DL-based approach can be a valuable tool in the early stages of wheel disc design optimization. The model not only learns the relationship between geometry and performance (in terms of stress, weight, and material usage), but it also significantly reduces the computational cost by rapidly exploring hundreds of thousands of design combinations. Once trained, the model enables the rapid exploration of hundreds of thousands of design combinations at negligible computational cost and can autonomously identify optimal configurations within minutes. This capability has the potential to significantly accelerate the design workflow and assist engineers in evaluating alternative geometries while maintaining the accuracy typically achieved through FEM simulations.

Possible developments include expanding the training dataset to cover a wider range of geometries, incorporating additional design parameters, and evaluating other loading conditions. The integration of physics-informed neural networks (PINNs) [28, 29] may also allow the model to better capture relationships governed by physical laws. Finally, the implementation of a user-friendly graphical interface may facilitate the practical deployment of the methodology in industrial environments.

Acknowledgments

This work was supported by MW within an Executivel PhD program.

Funding

This work was supported by MW Italai S.r.l. within an Executive PhD progam. Open access publishing facilitated by Politecnico di Torino, as part of the Wiley - CRUI-CARE agreement.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. T. W. Simpson, J. D. Peplinski, P. N. Koch, and J. K. Allen, "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering With Computers* 17, no. 2 (2001): 129–150, https://doi.org/10.1007/PL00007198.

2. J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science* 4 (1989): 409–423.

3. H. Tianchen, W. Qingshan, C. Liming, and Z. Rui, "Kriging-Based Uncertainty Optimization of Vibration Characteristics for Laminated Elliptical Shells Considering Material and Load Uncertainties," *European Journal of Mechanics - A/Solid* 111 (2025): 105587, https://doi.org/10.1016/j.euromechsol.2025.105587.

4. M. A. Bessa, R. Bostanabad, Z. Liu, et al., "A Framework for Data-Driven Analysis of Materials Under Uncertainty: Countering the Curse of Dimensionality," *Computer Methods in Applied Mechanics and Engineering* 320 (2017): 633–667, https://doi.org/10.1016/j.cma.2017.03.037.

5. F. E. Bock, R. C. Aydin, C. J. Cyron, N. Huber, S. R. Kalidindi, and B. Klusemann, "A Review of the Application of Machine Learning and Data Mining Approaches in Continuum Materials Mechanics," *Frontiers in Materials* 6 (2019): 110, https://doi.org/10.3389/fmats.2019.00110.

6. S. Yoo, S. Lee, S. Kim, K. H. Hwang, J. H. Park, and N. Kang, "Integrating Deep Learning Into CAD/CAE System: Generative Design and Evaluation of 3D Conceptual Wheel," *Structural and Multidisciplinary Optimization* 64, no. 4 (2021): 2725–2747, https://doi.org/10.1007/s00158-021-02953-9.

7. S. Venturini, C. Rosso, and M. Velardocchia, "An Automotive Steel Wheel Digital Twin for Failure Identification Under Accelerated Fatigue Tests," *Engineering Failure Analysis* 158 (2024): 107979, https://doi.org/10.1016/j.engfailanal.2024.107979.

8. Wheel Standards Committee, *SAE J328 Wheels — Passenger Car and Light Truck Performance Requirements and Test Procedures* (SAE International, 2021), https://doi.org/10.4271/J328_202107.

9. International Organization for Standardization, "ETRTO European Tyre and Rim Technical Organisation," in *Standards Manual* (ISO, 2025).

10. X. Wang, and X. Zhang, "Simulation of Dynamic Cornering Fatigue Test of a Steel Passenger Car Wheel," *International Journal of Fatigue* 32, no. 2 (2010): 434–442, https://doi.org/10.1016/j.ijfatigue.2009.09.006.

11. M. Firat, R. Kozan, M. Ozsoy, and O. H. Mete, "Numerical Modeling and Simulation of Wheel Radial Fatigue Tests," *Engineering Failure Analysis* 16, no. 5 (2009): 1533–1541, https://doi.org/10.1016/j.engfailanal.2008.10.005.

12. S. Santoni, "Modeling the Stiffness of a Compliant Mechanism Using Neural Networks, Mech," *Engineering Master's Thesis, Università politecnica delle March e, A. A* (2021): -2022.

13. C. Wang, L. Song, and J. F. Fan, "End-to-End Structural Analysis in Civil Engineering Based on Deep Learning," *Automation in Construction* 138 (2022): 104255, https://doi.org/10.1016/j.autcon.2022.104255.

14. R Core Team, *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, 2025)https://www.R-project.org/.

15. F. Chollet, *Keras: The Python Deep Learning Library* (SciX, 2015)Available at: https://keras.io.

16. C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).

17. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. (Morgan Kaufmann, 2012).

18. I. Godfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).

19. V. R. Joseph, "Optimal Ratio for Data Splitting," *Statistical Analysis and Data Mining The ASA Data Science Journal* 15, no. 4 (2022): 531–538, https://doi.org/10.1002/sam.11583.

20. B. Bischl, M. Binder, M. Lang, et al., "Hyperparameter Optimization: Foundations, Algorithms, Best practices, and Open challenges," *Data Mining and Knowledge Discovery* 13, no. 2 (2023): e1484, https://doi.org/10.1002/widm.1484.

21. T. Szandala, *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks* (Wroclaw University of Science and Technology, 2020).

22. D. C. Montgomery, *Design and Analysis of Experiments*, 9th ed. (Wiley, 2017).

23. S. Cai, S. Bileschi, E. D. Nielsen, and F. Chollet, *Deep Learning With Java Script – Neural Networks in TensorFlow.js* (Manning Publication, 2020).

24. D. P. Kingma, and J. Ba, "Adam: A Method for Stochastic Optimization," (2014)https://arxiv.org/abs/1412.6980.

25. G. Vicario, and R. Levi, *Metodi statistici per la sperimentazione* (Esculapio, 2008).

26. T. Gandía-Sastre, and M. Prados-Privado, "Integrating Finite Element Data With Neural Networks for Fatigue Prediction in Titanium Dental Implants: A Proof-of-Concept Study," *Applied Sciences* 15, no. 19 (2025): 10362, https://doi.org/10.3390/app151910362.

27. D. Lu, and D. M. Ricciuto, "Efficient Surrogate Modeling Methods for Large-Scale Earth System Models Based on Machine-Learning Techniques," *Geoscientific Model Development* 12, no. 5 (2019): 1791–1807, https://doi.org/10.5194/gmd-12-1791-2019.

28. 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," *Journal of Computational Physics* 378 (2019): 686–707, https://doi.org/10.1016/j.jcp.2018.10.045.

29. J. Kong, D. Zhou, X. Wei, and H. Wang, "A Physics-Informed Neural Network Method for LC Parameter Estimation in Three-Phase Inverter," in *Proceedings of the 2024 IEEE 10th International Power Electronics and Motion Control Conference (IPEMC2024-ECCE Asia)* (IEEE, 2024).