

A Deep Learning-Based Approach to the Estimation of Jominy Profile of Medium-Carbon Quench Hardenable Steels

Valentina Colla,* Marco Vannucci, Ismael Matino, and Renzo Valentini

The possibility to estimate the Jominy profile of steel based on its chemical composition is of utmost importance and high practical relevance for industries, which enables a preliminary assessment of the suitability of a specific steel grade to a particular application or to the requirements of a customer, by saving time and resources as the Jominy end-quench test is costly and time-consuming. More importantly, an estimator can be used in steel grade design, by supporting the investigation of the most suitable chemistry to meet some given specifications. The article proposes a novel approach to estimate the hardenability profile of medium-carbon quench hardenable steels, which exploits the potential of deep learning to correlate the steel metallurgy to the entire shape of the curve rather than to its single points, by thus being adaptable to a wide range of steel grades while providing very accurate estimates. Moreover, the proposed approach is suitable to implement a transfer learning paradigm, as it can exploit the knowledge acquired by training on a specific dataset to adapt the model to different steel grades for which less data or data holding different features are available.

machine learning (ML) approaches^[1,2] and mechanical behavior of metals and alloys, and connected technological developments are sensibly impacted by this category of approaches.^[3] They can, indeed, help metallurgical industry in sustaining the ever-increasing pressure toward the production of materials facing extreme conditions and/or providing optimal trade-off among weight, strength, ductility, corrosion resistance, toughness, as well as production costs and environmental compatibility. For instance, AI can support microstructural alloy tailoring and characterization, by providing reliable estimates of fundamental physical and mechanical properties,^[4] as well as monitoring of qualitative performance of each production step.^[5,6]

In the steel field, the application of ML for mechanical properties prediction is per se not a novelty, and a paradigmatic example is provided by the estimate of

the steel hardenability. Hardenability is the capability of steel to improve its hardness by quenching into martensite at a certain cooling rate after austenitizing at high temperature.^[7,8] For a specific cooling rate, hardenability is known to depend mostly on the steel metallurgy as well as on the austenite grain size.^[9] The hardenability of a specific steel grade is usually characterized through the so-called Jominy end-quench test,^[10] which provides a curve commonly known as Jominy profile. Being this test expensive and time-consuming, since many decades researchers applied numerical methods to estimate steel hardenability based on its chemistry.


The first models exploited standard statistical approaches, such as the multiplier-based solution proposed by Grossman in the 1940s,^[11] or a parametric linear model introduced in the 1960s, which is used as input steel metallurgy, austenite grain size, and quench end distance.^[12] In the 1970s, several authors worked on improving and generalizing the seminal model of Grossmann.^[13–16] Regression analysis was also applied in this domain with a focus on the effects of microalloying elements.^[17] In particular, the effect of the different chemical elements on each point of the Jominy curve was investigated in ref. [18] for a narrow range of Boron steels, and it was shown that C, Si, Mn, P, and Cr influence the whole shape of the curve, while Ni, Cd, B, and N mainly impact the sixth hardness value. More recently, nonlinear regression was applied to estimate

1. Introduction

In the last decade, an increasing interest is observed toward an interdisciplinary research field named “Materials Informatics”, which lies between materials engineering and data science and supports discovery, characterization, design, and development of materials through artificial intelligence (AI) and

V. Colla, M. Vannucci, I. Matino
TeCIP Istitute
Scuola Superiore Sant'Anna
56124 Pisa, Italy
E-mail: valentina.colla@sssup.it

R. Valentini
Dipartimento di Ingegneria Civile e Industriale
Università di Pisa
56122 Pisa, Italy

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/srin.202300374>.

© 2023 The Authors. Steel Research International published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

DOI: 10.1002/srin.202300374

the Jominy profile of gear steels.^[19] Further investigations exploited numerical analyses on the cooling path trend and considered its thermodynamics.^[20–22]

The parametric approach^[12] was refined and improved in subsequent studies. In the 1980s, a more complex parametric model was elaborated exploiting a nonlinear equation to calculate hardenability by considering the content of several microalloying elements.^[23] In the 1990s, a parametric analytical model was proposed,^[24] in which nonlinear equations provide an estimate of the parameters based on the steel metallurgy, but this approach is unsuitable to medium-carbon quench hardenable steels, as it does not appropriately consider how alloying elements interact with each other. An improved version of this model^[25] considers the interactions among alloying elements by means of empirically tuned interaction parameters, and is more accurate on hardenability curves of quenching and tempered steels, but proves to be reliable only on a few steel grades, as the parameters' tuning approach is empirical and merely numerical and does not embed any chemical or physical principle. In the same streamline, Jin et al.^[26] proposed a different parametric analytical function relating to hardness and distance from the quenched end, where a few parameters are present, whose values can be derived from the contents of C, Mn, Si, Cr, Mo, Ni, and B through further analytical formulas. This model as well was validated on a quite limited range of steels and the generalization of the formulas proving the parameters in neither proved nor straightforward.

In the same years, a method based on quench factor analysis^[27] was proposed to estimate the hardenability profile from simulated cooling curves, which shows good accuracy mostly for high hardness values.^[28]

In general, it can be observed that “traditional” numerical models estimating the hardenability curve show good performances only on a narrow range of steel grades considered for the tuning of their parameters, but their generalization properties are quite poor when applied for other grades, as the formulas correlating such parameters with the steel metallurgy are often empirical and hard to extend. Furthermore, often only some points of the profile are estimated with good accuracy, as the interactions among alloying elements are insufficiently addressed, also because they still are not fully known.

To overcome these limitations, since the 1990s ML-based models were investigated. In two pioneering articles,^[29,30] standard multilayer perceptron (MLP) neural networks (NN) were adopted for a pointwise estimation of the hardenability profile of some carbon steels based on their chemistry. Other similar investigations were carried out a few years later on different constructional steels^[31,32] and, more recently, on special steel grades.^[33,34] Moreover, a work concerning a particular type of pipeline steels used a NN that receives as inputs the contents of some chemical elements and a few fundamental mechanical properties (ultimate tensile strength, yield strength, percent elongation) that are assessed via standard tests.^[35]

However, all the methods to estimate the hardenability curve poorly consider the correlations among hardness measurements related to neighboring distance values. To overcome such issue, a parametric method was proposed,^[36] where the hardenability profile was approximated via a parametric analytical relationship linking hardness to distance from the quenched end, and wavelet NNs correlated the steel chemical composition to the function

parameters. More recently, the same authors presented a hierarchical NN-based Jominy profile predictor,^[37,38] which showed remarkable features in terms of reliability, robustness, limited computational burden, and maintainability. Such model estimates the hardenability profile in a pointwise way through a “cascade” of simple NNs, each one outputting one hardness value of the Jominy curve and being fed with a few input variables, which accounts for both influence of the microalloying elements on the single profile points and correlation among the neighboring hardness values, as previously computed hardness values are used to calculate the following ones. Nonetheless, training of this model requires the availability of a dataset holding “complete” Jominy profiles, i.e., curves formed by 15 points.

Recently, fuzzy systems were also adopted to estimate the Jominy profile using the steel metallurgy,^[39] but the research work focused on structural steels for quenching and tempering.

A completely different approach is followed in ref. [40], where the focus is on the estimate of the so-called total hardness, i.e., an aggregated measure of hardenability, via standard NNs using as inputs the contents of a few alloying elements, some heat treatment parameters, and the so-called specific Jominy distance, i.e., the value of the distance from the quenched end where 50% of the microstructure is martensite, which is a function of the C content.

ML techniques have also been used also to extract the Jominy profile based on the data provided by nondestructive characterization techniques applied to the Jominy sample, such as in ref. [41], where a combination of parameters characterizing the data provided by an eddy current and a hysteresis loop measurement system are used to feed a generalized regression NN. However, such measurement systems are not commonly used in steelworks to characterize Jominy samples, while from an industrial point of view tools that exploit data that are usually collected during the standard operating practice are more relevant and easier to deploy. This is also one of the basic ideas inspiring the work described in this article.

To sum up, with respect to standard analytical models, ML-based approaches provide the consistent advantage of better generalization capabilities and native solutions for implementing transferability (i.e., the learning procedures), but require a relevant volume of data in a suitable format to be trained. Especially pointwise models (i.e., the ones estimating single points of the curve) can be sensitive to anomalous or erroneous data, if this issue is not properly considered. Furthermore, incomplete experimental Jominy curves, which are very often found in the industrial practice, can seldom be used, although they still convey relevant and useful information. Moreover, retraining of the model using new data follows basically the same procedure of the former learning stage, and to somehow transfer the previously acquired knowledge, two solutions are available: 1) use of the internal parameters of the previous version of the model as starting point for the training, which can bias and negatively affect the final performance of the learning procedure, as it can get trapped into a “local minimum;” 2) exploit a larger dataset containing both previously available and new data, which might require relevant computational resources and still does not ensure optimality of the results, especially if the newly available data are limited in number.

The present work aims at covering this gap, by proposing an approach based on monodimensional (1D) convolutional neural network (CNN) with a threefold aim: 1) to provide a more accurate estimate of hardenability thanks to the ability of 1D CNN in learning the effects of steel metallurgy on the shape of the Jominy curve; 2) to improve robustness with respect to anomalous data by relating the steel metallurgy to the overall shape of the hardenability curve instead of its single points; and 3) to implement a transfer learning (TL) paradigm exploiting the knowledge acquired by training on a specific dataset for adapting the model to different steel grades, for which less data are available or data holding different features are provided.

As a further element of novelty of the present work, CNNs, which belong to the large category of deep learning (DL) approaches, have been never applied so far to steel hardenability prediction, although DL is receiving an ever-increasing interest in the scientific community to estimate of other mechanical properties of steel.

2. Experimental Section

2.1. Materials

Several steel grades have been considered, all belonging to the wider class of quenching and tempered steels for mechanical constructions. The available data, where each sample refers to a steel metallurgy and associated Jominy profile, were provided by two different steelworks producing steel grades that are only partly overlapping. Therefore, in practice, two different datasets are available: DS_A, formed by 1500 samples, and DS_B, including 250 observations.

The Jominy test was developed according to the ASTM standard for all the considered data.^[8]

Some specimens after the Jominy test were subjected to an optical metallographic analysis to assess the final microstructure in correspondence to the obtained hardness measurements.

Table 1 reports the main statistical parameters of the chemical composition of the steel samples considered in both datasets, **Figure 1** shows the related scatter plots, while **Figure 2** highlights the difference between the average values and punctual standard deviation of the Jominy profiles included in the two considered

datasets. **Table 2** depicts some exemplar metallurgies of steels belonging to DS_A and DS_B. DS_A mostly refers to carbon steels generally classified in the commercial categories C15, C20, C30, C35, C45, C50, C55, and C60, while DS_B is related to micro-alloyed steels commercially labeled as 41CrMo4, 36MnCr5, 35 MnCrB5, and 41Cr5.

2.2. CNNs

CNNs are a DL algorithm that is widely used in image^[42] and video recognition, natural language processing, and other applications that require the processing of sequential data. They are intensively applied also in steel production for a quite wide range of tasks, such as, for instance, shape and surface defects detection and classification,^[43–45] microstructure analysis and classification,^[46,47] prediction of the end point of the converter,^[48] and processing temperature data in continuous casting.^[49]

The key idea behind CNNs is to use a series of convolutional layers to extract features from an input image, followed by one or more fully connected layers to classify the image. The convolutional layers use a set of filters (also known as kernels or weights) to scan across the input image, extracting local features at each position.

The sequential composition of multiple convolutional layers can lead CNNs to learn increasingly complex patterns and detect structured objects in an image.

One of the main advantages of CNNs is their ability to automatically learn hierarchical representations of data, where lower level features (such as edges and corners) are combined to form higher level features (such as shapes and objects). This makes them particularly effective at tasks such as object recognition, where the input images may contain complex and variable backgrounds, lighting conditions, and object poses. CNNs are also used to obtain high-resolution images or other forms of bidimensional data from low-resolution images or measurements matrices, such as in the exemplar application described in ref. [50].

Overall, CNNs have proven to be highly effective in a wide range of applications and have significantly advanced the state of the art in areas such as image classification, object detection, and semantic segmentation.

Table 1. Main statistical parameters (average value, standard deviation, and minimum and maximum values) of the contents of the relevant chemical components for the steel grades considered in the present work.

Statistics	C	Mn	Si	P	S	Cr	Ni	Mo	Cu	Sn	Al	V	Nb	Ti	B
DS _A															
Average value	0.37	0.98	0.26	0.016	0.027	0.48	0.098	0.043	0.17	0.009	0.018	0.0039	0.0018	0.023	0.00074
Standard dev.	0.09	0.25	0.19	0.0037	0.012	0.38	0.18	0.062	0.025	0.0014	0.0068	0.0089	0.0025	0.017	0.0011
Min.	0.16	0.64	0.18	0.007	0.007	0.08	0.01	0.01	0	0.004	0.002	0	0	0.006	0
Max	0.6	1.49	2.05	0.03	0.17	1.23	1.32	0.28	0.25	0.022	0.05	0.016	0.044	0.055	0.0032
DS _B															
Average value	0.37	1.17	0.25	0.015	0.017	0.28	0.052	0.022	0.16	0.009	0.021	0.0036	0.002	0.047	0.0018
Standard dev.	0.019	0.099	0.021	0.037	0.005	0.11	0.013	0.025	0.027	0.0015	0.0028	0.002	0.0016	0.001	0.00068
Min.	0.32	0.9	0.19	0.008	0.007	0.13	0.04	0.01	0.01	0.006	0.011	0	0	0.009	0
Max	0.44	1.44	0.32	0.028	0.026	0.62	0.12	0.015	0.24	0.017	0.03	0.008	0.006	0.055	0.003

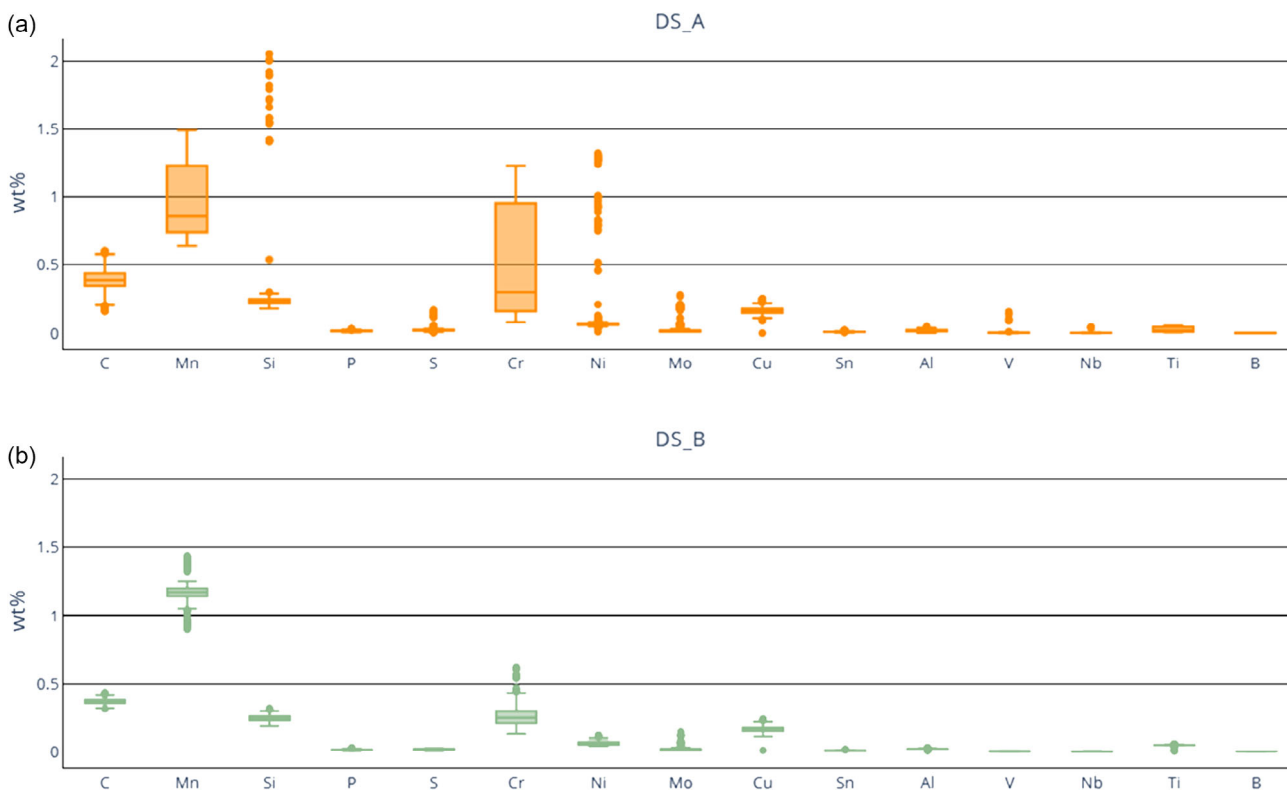


Figure 1. Scatter plots of the chemical compositions of the steel samples included in: a) dataset DS_A; b) dataset DS_B.

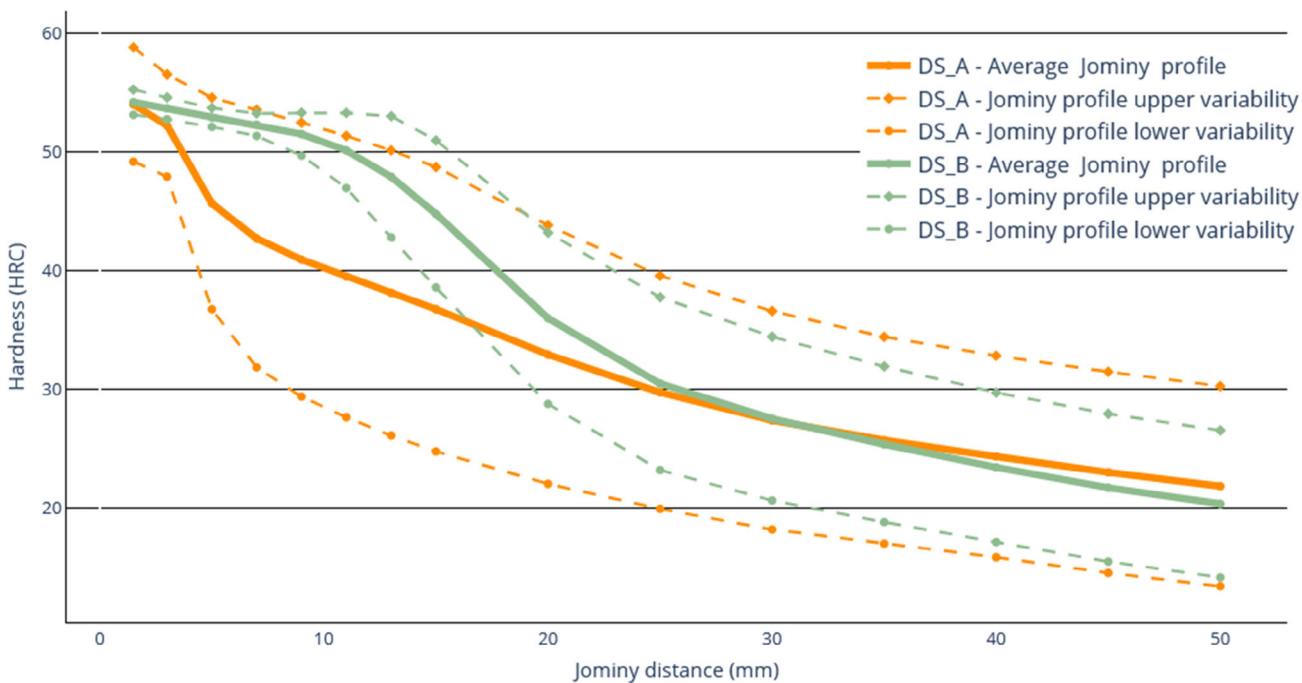


Figure 2. Pointwise average values and standard deviation of the Jominy profiles included in the two considered datasets.

Table 2. Exemplar chemical compositions of the steels belonging to DS_A and DS_B. The rows highlighted in yellow correspond to the Jominy profiles depicted in Figure 8.

Dataset	C	Mn	Si	P	S	Cr	Ni	Mo	Cu	Sn	Al	V	Nb	Ti	B
DS _A	0.45	0.72	0.21	0.012	0.017	0.16	0.07	0.01	0.18	0.008	0.014	0	0	0.01	0
	0.365	1.24	0.27	0.01	0.017	0.3	0.05	0.02	0.14	0.007	0.02	0.006	0.002	0.043	0.0015
	0.6	0.87	2.05	0.017	0.018	0.19	0.05	0.01	0.13	0.009	0.005	0.003	0.002	0.01	0
	0.49	0.8	0.24	0.009	0.023	0.17	0.06	0.01	0.13	0.007	0.016	0.001	0.001	0.011	0
	0.51	0.64	1.55	0.022	0.013	0.15	0.05	0.01	0.15	0.009	0.005	0.001	0.002	0.01	0
	0.455	0.75	0.23	0.014	0.022	0.15	0.06	0.01	0.17	0.01	0.012	0.002	0.001	0.013	0
	0.45	0.72	0.22	0.017	0.024	0.16	0.08	0.02	0.16	0.008	0.012	0.001	0.001	0.01	0
	0.445	0.73	0.21	0.013	0.025	0.13	0.06	0.01	0.14	0.008	0.005	0	0.001	0.01	0
	0.455	1.36	0.22	0.018	0.145	0.13	0.06	0.01	0.21	0.012	0.002	0.003	0.001	0.009	0
	0.41	0.73	0.2	0.013	0.022	1.02	0.07	0.18	0.15	0.01	0.013	0.005	0.002	0.011	0
	0.41	0.78	0.23	0.012	0.025	1.02	0.07	0.19	0.16	0.009	0.023	0.005	0.002	0.012	0
	0.45	0.74	0.21	0.022	0.025	0.17	0.08	0.02	0.16	0.009	0.012	0.003	0.001	0.011	0
	0.375	1.2	0.24	0.015	0.022	0.31	0.06	0.01	0.2	0.009	0.026	0.004	0.003	0.048	0.0016
	0.2	1.29	0.23	0.017	0.027	1.16	0.07	0.02	0.21	0.01	0.021	0.004	0.003	0.01	0
	0.17	1.2	0.23	0.012	0.022	1.05	0.07	0.01	0.023	0.01	0.022	0.005	0.003	0.011	0
DS _B	0.395	1.43	0.2	0.015	0.016	0.44	0.07	0.01	0.2	0.009	0.014	0.002	0.002	0.01	0
	0.405	0.96	0.23	0.018	0.016	0.15	0.07	0.01	0.24	0.01	0.022	0.003	0.003	0.052	0.0026
	0.39	1.21	0.25	0.018	0.018	0.28	0.06	0.01	0.15	0.008	0.021	0.004	0.003	0.045	0.0015
	0.345	1.34	0.24	0.02	0.015	0.32	0.06	0.01	0.23	0.009	0.02	0.004	0.003	0.048	0.0027
	0.41	0.94	0.24	0.011	0.012	0.14	0.06	0.01	0.15	0.01	0.021	0.003	0.002	0.048	0.0028
	0.435	0.97	0.25	0.012	0.013	0.15	0.06	0.01	0.15	0.008	0.021	0.002	0.002	0.055	0.0026
	0.38	1.11	0.23	0.014	0.009	0.21	0.06	0.02	0.14	0.007	0.023	0.005	0.004	0.049	0.0026
	0.36	1.13	0.27	0.013	0.014	0.57	0.07	0.07	0.15	0.009	0.023	0.006	0.003	0.05	0.0027
	0.355	1.12	0.22	0.013	0.021	0.2	0.06	0.02	0.15	0.008	0.022	0.005	0	0.05	0.003

In the present work, a 1D convolutional layer is adopted, which operates on 1D data that are commonly used for processing 1D input data, such as time series, speech signals, or sequential data. Like multidimensional layers, 1D convolutional layers use filters that are, however, in only one dimension and perform the convolution operation in only one dimension, scrolling the input through its length and producing a 1D array that is then passed to subsequent layers of the network (Figure 3). Therefore, here 1D convolutional layers are used to extract and learn the salient features from the processed Jominy profiles.

2.3. Autoencoders

Autoencoders (AE) are a type of NN architecture used for unsupervised learning and dimensionality reduction tasks. They are designed to encode input data into a lower dimensional representation and then decode it back to its original form. The goal of an autoencoder is to learn a compact data representation capturing their important features. The architecture of an autoencoder consists of two main and connected parts: an encoder and a decoder. The encoder takes the input data and maps it to a lower dimensional representation, often called latent space or latent domain, which contains a compressed version of the input data. The

decoder takes this compressed representation and reconstructs the original input data from it.

The encoder and decoder components are typically implemented using NNs. The encoder network consists of one or more hidden layers that progressively reduce the dimensionality of the input data. The decoder network is designed to mirror the encoder, with hidden layers that gradually increase the dimensionality back to the original input shape.

During training, the autoencoder aims to minimize the reconstruction error, which measures the difference between the original input and the reconstructed output. This is typically achieved by using a loss function such as mean squared error or binary cross-entropy, depending on the nature of the input data.

In data and image preprocessing, AE are applied for various purposes, such as dimensionality reduction and anomaly detection. In the former application, by learning a compressed representation of the data, AE can reduce the dimensionality of high-dimensional input data, and this is particularly useful for tasks such as, for instance, data visualization, variables selection,^[51] feature extraction,^[52] and denoising.^[53] Within anomaly detection tasks, AE can learn to reconstruct normal patterns in the input data. When presented with anomalous or unfamiliar data, the reconstruction error tends to be higher, making AE useful for detecting anomalous data. AE have been widely used in

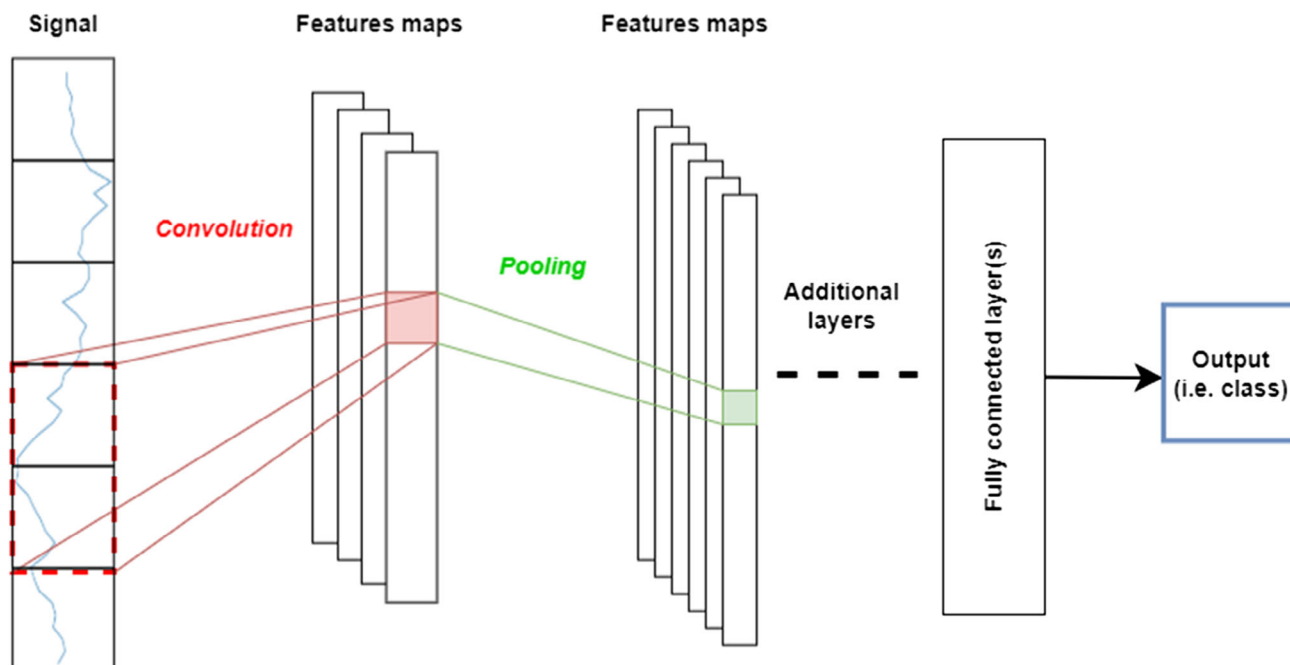


Figure 3. Exemplar 1D CNN for signal processing.

various domains, including steel production and manufacturing, and provide a powerful framework for learning meaningful representations of data and extracting useful features.

2.4. Structure of the Jominy Profile Estimator

The basic idea behind the proposed approach to estimate the Jominy profile to the steel metallurgy is similar to the one characterizing some of the previously mentioned parametric Jominy profile predictors, i.e., to relate steel metallurgy to the shape of the Jominy curve rather than to the single hardness measurements, and to subsequently tie such encoded shape to steel metallurgy. This approach is intended to improve the robustness and generalization capability of the predictor. However, the novelty here is that the “encoding” is not explicit, namely, it is not represented by a known mathematical formula, which shows limitations due being a schematic representation of the profile. Here, a first part of the model aims to learn the salient characteristics of different Jominy profiles (the shapes), while another part of the model aims to learn the relationship between chemistry and the encoded curve shape and makes that relationship available in a TL scenario.

TL is a ML approach where knowledge acquired from solving one problem is applied to a different but related problem.^[54] Instead of training a model from scratch on a specific task, TL leverages the knowledge gained from solving a task to improve performance on a different one. TL has been successfully applied in various domains and applications also for the steel sector, such as image analysis,^[55–58] natural language processing,^[59] and speech recognition.^[60] It allows models to benefit from previous knowledge and accelerate the development and deployment of ML systems. The idea behind TL is that models

can learn general features or representations from a large amount of data in one domain and then apply that knowledge to a different but related domain with less data. By transferring the learned representations, the model can effectively generalize and adapt to the new task more quickly and accurately.

This approach could prove beneficial in training models for novel steel variants when there is limited data available. It could also be applied to adapt the model for use in diverse steel plants, each having unique methods of gathering experimental data or conducting tests, potentially leading to changes in the mentioned relationship.

The Jominy curve estimator holds two main components: 1) an autoencoder (shown in **Figure 4** and referred to as AE_Jom in the following), which learns and encodes the different shapes of the Jominy profiles in a latent domain L , which is more compact than the standard 15-point profile space; 2) a deep neural network (DNN) (shown in **Figure 5** and referred to as DNN_Jom in the following), which provides the estimated Jominy curve based on the steel composition using AE_Jom, specifically mapping such chemistry in the latent domain L .

AE_Jom is a modified autoencoder designed to learn a condensed representation of a given dataset consisting of Jominy profiles, which “summarizes” the shape of the different profiles. The primary objective of AE_Jom is to accurately reconstruct the original profile from this compressed representation. This goal is pursued by exploiting the distinctive feature of AE_Jom constituted by its initial layer that employs a 1D convolution followed by a max-pooling operator to extract information on the input Jominy curves. The autoencoder follows the convolutional layer as an ensemble of N_1^{AE} fully connected layers, symmetrically arranged around the latent space L . This latent space serves as compact representation of the original Jominy profiles (with

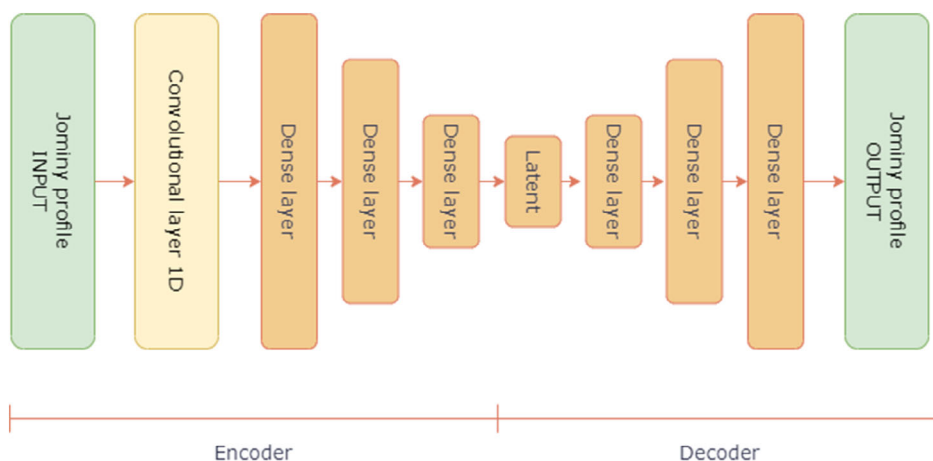


Figure 4. The AE_Jom architecture.

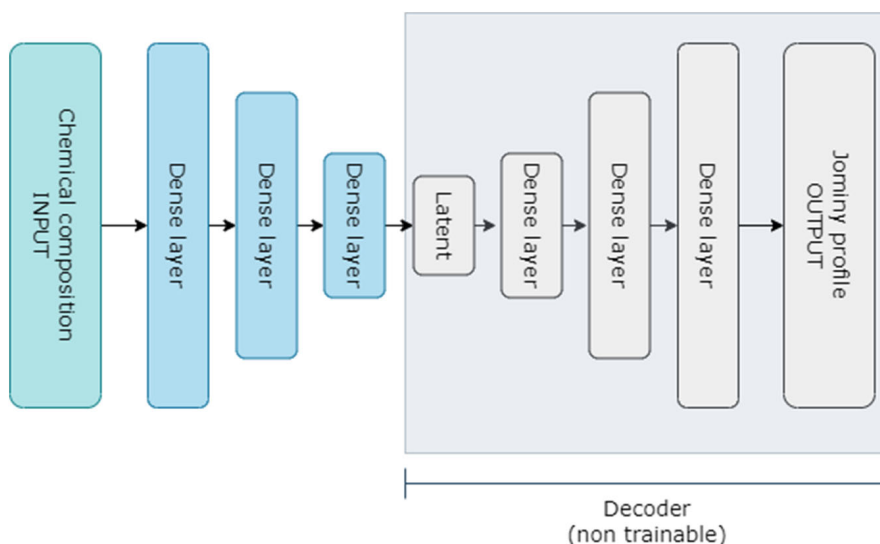


Figure 5. The DNN_Jom architecture.

hardness measured in the standard 15 points), encapsulating the shape details of the Jominy curves thanks to the 1D convolutional layer. The layers of AE_Jom employ a commonly used activation function in autoencoder implementations, namely, the LeakyRelu function. To sum up, AE_Jom shows two main symmetrical components, an encoder and a decoder: the former one learns and stores a condensed representation of Jominy profile in L , while the latter one reconstructs it.

Based on the variable section analysis performed in ref. [37], the contents of all the 15 chemical elements reported in Table 1 are fed as input of all the models.

The hyperparameters of the entire JNetwork were optimized by a grid-search process that evaluated the efficiency (in terms of mean absolute error [MAE]) of each of the tested combinations using the available training data. Candidate values were selected according to the dimension of the data available for the selection to avoid eventual overfitting issues. The optimal values of these hyperparameters are shown in Table 3, while the tested values are listed in Table 4.

During the grid search, all tested models were evaluated for potential overfitting issues by monitoring their performance on training and validation data. Specifically, in these tests, it was noticed that the predictive model's performance on validation data (as compared to training data) deteriorated by ≈ 0.05 – 0.15 hardness Rockwell C scale (HRC) (MAE), with a slightly greater decline in more complex models. However, this outcome does not indicate any specific overfitting issues in the considered models and resulted in the choice of the previously designated hyperparameters. The performance of the ten best performing combinations—out of more than 6000 tested—of these hyperparameters is shown in Table 5 in terms of the MAE obtained on the training and validation datasets.

2.5. Training Procedure of the Estimator

The training procedure of the Jominy estimator is basically divided into two parts. The first one trains the AE_Jom by using

Table 3. Hyperparameters values settings of the JNetwork (Jom AE and DNN Jom).

Hyperparameter	Value
Jominy AutoEncoder [Jom AE]	
Convolutional layer filters number	3
Convolutional layer filters dimension	2
Pooling method	Max
Pooling dimension	2
Number of dense layers (symmetrical respect to L)	2
Neurons per layer (symmetrical respect to L)	[20, 20]
Dense layers activation function	LeakyReLU
Latent space dimension	4
Deep Neural Network Jominy [DNN Jom—Chemical Encoder]	
Number of dense layers	2
Neurons per layer	[20, 20]
Dense layers activation function	ReLU

Table 4. Tested hyperparameters values for the tuning of the JNetwork (Jom AE and DNN Jom).

Hyperparameter	Tested values
Convolutional layer filters number	2, 3, 5, 10
Convolutional layer filters dimension	2, 3, 4
Pooling dimension	2, 3, 4
Latent space dimension	2, 3, 4, 5, 6
Jom AE—neurons per layer (symmetrical with respect to L)	(10,10), (10,10,10), (20,10), (20,20), (20,20,20), (30,30)
DNN Jom—neurons per layer	(10,10), (10,10,10), (20,10), (20,20), (20,20,20), (30,30)

a training dataset composed of Jominy curves to learn a compact representation of Jominy profiles shapes. The second stage learns a suitable mapping of steel metallurgy into L space. In both steps, training exploits the adaptive moments estimation algorithm which belongs to the family of stochastic optimization

methods, and is computationally efficient and well suited for nonstationary objectives and problems with very noisy or sparse gradients.

After the AE_Jom training stage, its decoder acquires the capability to reconstruct a Jominy profile stored by latent space L . In the second training stage, such decoder is integrated into a DNN called DNN_Jom (see Figure 5), which incorporates a sort of “chemical encoder” that maps the chemical composition to its corresponding L representation. By connecting the previously trained decoder with this new encoder within DNN_Jom, the DNN associates the steel metallurgy to the corresponding Jominy profile. Subsequently, DNN_Jom is trained using an experimental dataset derived from Jominy tests, which comprises the metallurgies of various steel specimens and their hardenability profiles. Noticeably, this second stage tunes only the parameters in the encoder layers, while the decoder within DNN_Jom keeps the same internal parameter values that were obtained during the training of AE_Jom. These adjustments aid in the mapping of chemical compositions to a latent representation. This latent representation, when fed into the “frozen” decoder, generates the estimated Jominy profile by leveraging the knowledge acquired through TL.

3. Experimental Results

The effectiveness of the estimator has been assessed by conducting an experimental campaign using the two datasets described in Section 2.1. The objective of these tests is to evaluate the accuracy of the model and its ability to implement TL as far as the Jominy curve shape is concerned via the decoder, as explained in Section 2.5.

To measure the performance of the estimator, two well-known indexes are used: 1) The MAE, which is a direct measure of the difference between estimated and real hardness value, regardless of its sign, for the i th point of the Jominy profile is computed as

$$MAE_i = \frac{1}{\text{Dim}_{ts}} \cdot \sum_{j=1}^{\text{Dim}_{ts}} |\hat{h}_{i,j} - h_{i,j}^T| \quad (1)$$

where $h_{i,j}$ and $h_{i,j}^T$ are, respectively, the estimated and target hardness value at the distance d_i of the j th Jominy curve in the test

Table 5. Performance achieved by the top-ten hyperparameters configurations in terms of MAE on dataset DS_A for training and validation sets.

Latent space dim.	Conv. layer filters num.	Jom AE hid. neurons	Conv layer filter dim.	Pooling dim.	DNN Jom hid. neurons	MAE Tr	MAE Vd
4	3	(20, 20)	2	2	(20, 20)	0.85	0.96
4	3	(10, 10)	3	2	(20, 20)	0.88	0.98
3	2	(20, 20)	2	3	(20, 20)	0.82	0.98
4	4	(10, 10)	2	3	(20, 20)	0.83	0.98
3	3	(10, 10)	3	2	(20, 20)	0.82	0.98
4	4	(10, 10)	2	2	(20, 20)	0.85	0.98
5	2	(20, 20)	2	3	(20, 20)	0.84	0.98
4	3	(20, 20)	3	3	(20, 20)	0.88	0.98
3	4	(20, 20)	2	2	(20, 20)	0.88	0.98
5	2	(10, 10)	3	3	(20, 20)	0.87	0.99

dataset holding Dim_{ts} samples. 2) The percentage absolute error (e), which considers the relative error, for the i th point of the Jominy profile is computed as

$$pe_i = \frac{100}{\text{Dim}_{\text{ts}}} \times \sum_{j=1}^{\text{Dim}_{\text{ts}}} \frac{|\hat{h}_{ij} - h_{ij}^T|}{h_{ij}^T} \quad (2)$$

The estimator ability to accurately predict the Jominy profile is primarily evaluated using DS_A . This dataset holds enough observations to support model training and a reliable evaluation of the performance on validation and test datasets. To achieve this, DS_A is split into two parts: 80% of the data are exploited for training and validation, while the 20% left are used for testing. These two datasets have been used to select the values of some hyperparameter of AE_Jom and DNN_Jom, namely, the number of layers N_1^{AE} and layers' neurons in the symmetrical architecture of the decoder and encoder of AE_Jom, the number of filters N_f^{AE} adopted in the 1D convolutional layer of AE_Jom, the number of layers N_1^{DNN} and neurons N_n^{DNN} per layer in the encoder of DNN_Jom, and the latent space dimension L within AE_Jom and DNN_Jom.

Each combination of the above-listed hyperparameters has been used to train and validate the corresponding model to find the best performing one. The selection of the data combination that yields a lower average prediction error across the points of the Jominy curve is chosen to be compared to the performance obtained by other NN-based approaches. Furthermore, to assess the capability of the proposed estimator in a TL context, DS_A and DS_B are jointly utilized. After training DNN_Jom using DS_A as previously described, the corresponding decoder is extracted and embedded in a DNN_Jom trained with DS_B . This allows for leveraging knowledge extracted from DS_A through the original AE_Jom application.

The achieved results are compared to those obtained by the sequential predictor described in ref. [37], by a NN of the MLP type with one hidden layer (indicated in the following as BaseMLP) including neurons with standard sigmoidal activation function, analogous to the one proposed in ref. [30], and by a fully connected DNN whose architecture has been optimized (in terms of number of layers and neurons in each layer) like DNN_Jom.

The comparison with BaseMLP is introduced, as ref. [30] is the seminal work where NNs were first introduced to estimate the hardenability profile based on the steel metallurgy, but some adaptations are needed to cope with the available database, namely, we designed a NN with 13 inputs and 15 outputs, as in both adopted datasets the Jominy profiles are formed by 15 hardness measurements, and the number of neurons in the hidden layer N_h^{MLP} has been selected by implementing a grid search which took into account a basic rule for the design standard NNs, i.e., to have a maximum total number of internal parameters, which is about one-fourth of the total number of samples that are used for the training. In the present case, the maximum possible value of N_h^{MLP} is 10; therefore, the grid search considered $5 \leq N_h^{\text{MLP}} \leq 10$ and the best performance were obtained with $N_h^{\text{MLP}} = 9$.

To evaluate the benefits in terms of knowledge transferability, the proposed estimator is compared to models trained exclusively exploiting DS_B . In this context, the combination of hyperparameters that achieves the highest accuracy employs two layers for both the encoder and decoder sides of the AE_Jom ($N_1^{\text{AE}} = 2$), where each layer is formed by 20 neurons, where $N_f^{\text{AE}} = 3$, $N_1^{\text{DNN}} = 2$ with 20 neurons in each layer and the latent space dimension is $L = 4$.

This configuration of the predictor achieves an average MAE value over the profile on DS_A of 0.96 HRC, which is 16% lower than the sequential model proposed in ref. [37], for which an average MAE value achieved is 1.10 HRC.

In addition, as a further term of comparison, to show the effectiveness of the encoder stage, a standard fully connected DNN has been evaluated using DS_A . The DNN architecture has been optimized by testing various combinations of layer numbers and hidden neurons, like the hyperparameter optimization process of the proposed estimator. The best configuration in this case is a three-layer network holding 20 neurons in each hidden layer, utilizing the ReLU activation function. This configuration achieves an average value of the MAE of 1.12 HRC.

Table 6 compares the MAE achieved by the four estimators on each point of the Jominy curve on the test set extracted by DS_A , while **Figure 6** propose a graphical comparison in terms of percentage error.

The effectiveness of the suggested TL method, which involves reusing the decoder component of a previously trained DNN_Jom, according to the TL procedure outlined in Section 2.5 has also been assessed using DS_B . Also in this case, DS_B was split in two parts: 80% of the data were used for training and validation, while the 20% left are used for testing.

Initially, a direct approach has been adopted where DS_B input samples are fed to a DNN_Jom trained only on DS_A , which provided a MAE exceeding the value of 4 HRC along the whole

Table 6. Average MAE achieved on the test data set extracted by DS_A at each point of the Jominy profile by the three considered estimators.

Distance [mm]	Base_MLP	MAE [HRC]		
		Sequential	DNN	DNN_Jom
1.5	0.71	0.25	0.66	0.42
3	0.86	0.55	0.75	0.52
5	2.20	0.46	0.8	0.76
7	2.13	0.7	0.82	0.83
9	1.96	0.73	0.85	0.84
11	1.96	1	1.11	0.88
13	2.13	1.52	1.23	0.89
15	2.32	1.52	1.33	0.99
20	2.51	1.57	1.85	1.25
25	2.67	1.72	1.4	1.26
30	2.52	1.39	1.4	1.2
35	2.32	1.57	1.38	1.14
40	2.20	1.34	1.06	1.09
45	2.28	1.13	1.06	1.1
50	2.36	1.14	1.1	1.12

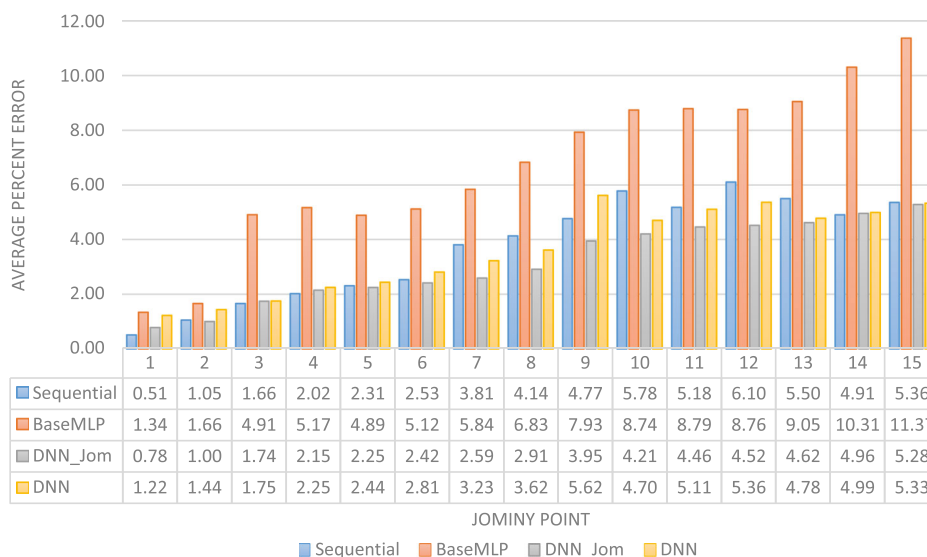


Figure 6. Average percent error achieved in baseline mode on DS_A by DNN_Jom.

profile. Such poor achievement is likely due to differences in chemistry and profile shapes between the two datasets.

As a result, the proposed approach is encouraged, and part of the DNN_Jom that was previously trained on DS_A was retrained according to the TL approach described in Section 2.5. Such approach was compared to the sequential model^[37] and to a fully connected DNN trained exclusively on DS_B . The DNN architecture was optimized in a similar manner to DS_A , resulting in a smaller network consisting of two layers with ten neurons each, activated by the ReLU function.

The average MAE value along the profile on the test data for DNN_Jom (with TL training) is 1.09 HRC. The sequential model exhibited an average MAE of 1.34 HRC (23% higher than the one provided by DNN_Jom), and the fully connected DNN an average MAE of 1.40 HRC (29% higher than the one provided by DNN_Jom). **Table 7** compares the performance of the three estimators in terms of MAE achieved on each point of the Jominy curve on the test set extracted by DS_B , while **Figure 7** propose a graphical comparison in terms of percentage error.

To qualitatively depict the performance of the estimator, **Figure 8** exemplarily compares some experimental Jominy profiles of far different shapes and their estimates provided by DNN_Jom. The corresponding steel metallurgies and databases are provided in Table 2. **Figure 9** shows the microstructure observed in correspondence to the fifth point (J5) of the Jominy curve for the steel specimen used to obtain the profile reported in Figure 8b: a fully martensitic structure (Nital etching) is shown.

4. Discussion

Figure 6 and 7 show that the performances of all models decrease with increasing values of the distance from the quenched end. This behavior is due to the higher variability of the data, also shown Figure 2, which depends on the fact that the farther the specimen area from the hardened end, the coarser and more

Table 7. Average MAE achieved on the test data set extracted by DS_B at each point of the Jominy profile by the four considered estimators, being DNN_Jom trained with the TL approach, while Base_MLP was simply retrained.

Distance [mm]	Error [HRC]			
	Base_MLP	Sequential	DNN	DNN_Jom [TL]
1.5	0.72	0.84	0.97	0.78
3	0.74	0.80	1.18	0.80
5	0.70	0.92	0.78	0.82
7	0.69	0.80	0.99	0.80
9	0.81	1.22	0.95	1.01
11	1.39	1.14	1.54	1.14
13	2.22	1.91	2.07	1.44
15	2.31	1.78	1.85	1.45
20	2.00	2.23	2.56	1.53
25	1.48	1.36	1.75	1.06
30	1.35	1.36	1.24	1.13
35	1.37	1.41	1.52	1.09
40	1.42	1.56	1.32	1.02
45	1.43	1.44	1.19	1.16
50	1.44	1.32	1.1	1.09

inhomogeneous the microstructure, as upper bainite and banded ferrite–pearlite microstructures are present. Consequently, the measured hardness value heavily depends on the particular measurement point, and a higher data dispersion is observed.

As expected, BaseMLP shows the worst performances especially on DS_A , while on DS_B its performances are comparable only for the lowest distance value, where the data variability is limited. This behavior is due to the fact that this model neglects the correlation among hardness values measured in neighboring points, which is the strength of the hierarchical estimator, and

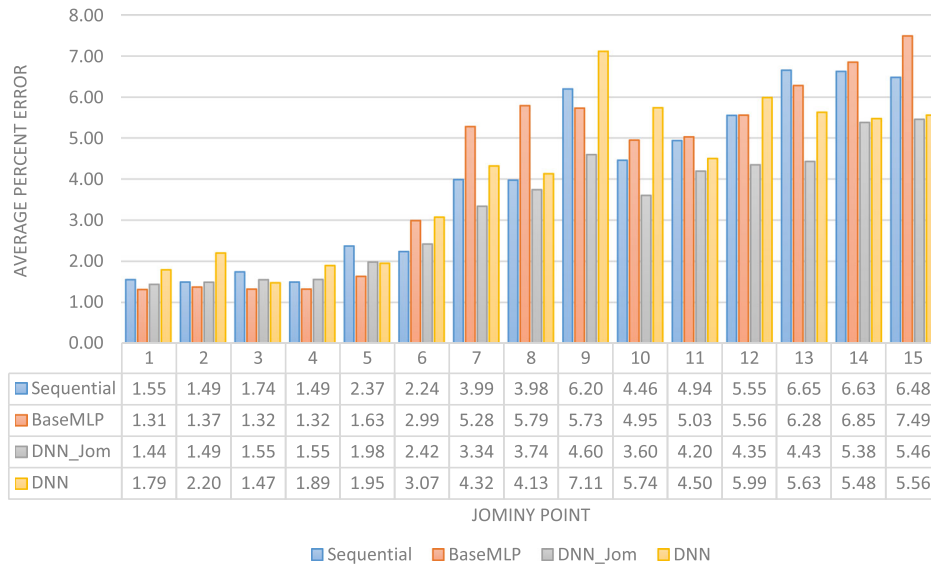


Figure 7. Average percentage error achieved on DS_B by DNN_Jom trained according to the proposed TL approach.

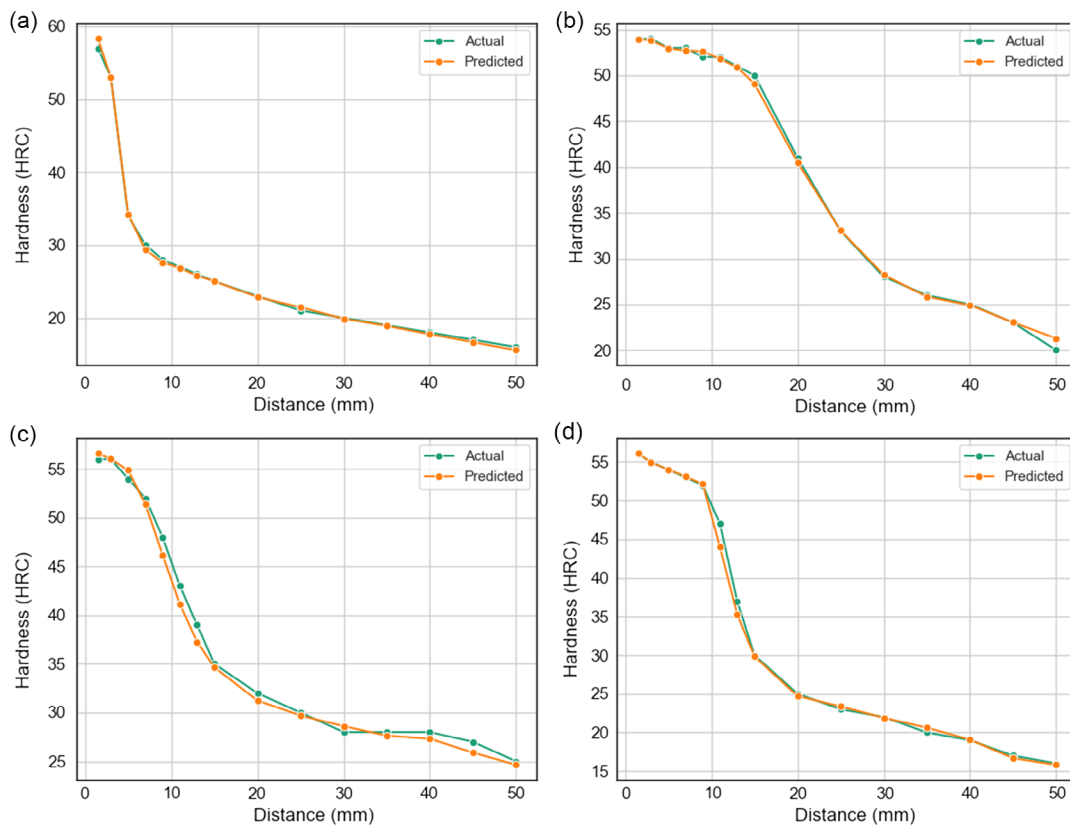


Figure 8. Exemplar comparisons between experimental Jominy profiles of far different shapes and their estimates provided by DNN_Jom. The corresponding steel metallurgies are provided in Table 2 (respectively, in a) row 1, b) row 2, c) row 16, and d) row 17).

does not benefit from the high number of degrees of freedom and the superior optimization procedures of DL architectures. Moreover, it does not show any ability to elaborate a compact representation of the profile, and this negatively affects its generalization capability in a TL context.

On the larger dataset (DS_A) the hierarchical estimator exhibits superior performance in the initial part of the jominy curve, which primarily pertains to a few specific components of the steel metallurgy. However, in percentage value, the decrease of performance conveyed by DNN_Jom over these points is negligible,

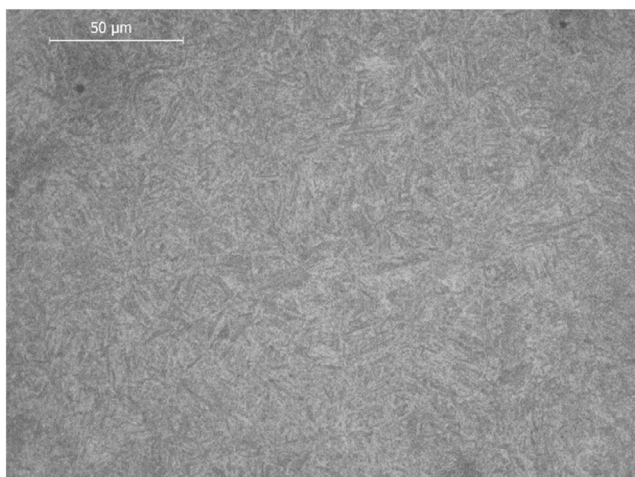


Figure 9. Microstructure observed at J5 on the steel specimen used to obtain the Jominy profile reported in Figure 8b: fully martensitic structure (Nital etching).

being hardness high in this area of the profile. Conversely, DNN_Jom outperforms the sequential estimator in the central part of the profile, which holds a significant industrial importance, being considered by most specifications for steel applications, and here the percentage error provided by DNN_Jom is significantly lower. Moreover, DNN_Jom achieves more balanced performances across the entire Jominy curve, showcasing its ability to effectively learn the impact of steel metallurgy on the overall profile shape. DNN_Jom also displays good qualitative performance, as depicted in Figure 7.

When the TL approach is applied to apply the estimator to the second dataset (DS_B), DNN_Jom outperforms the other estimators, by providing a slightly better accuracy on the initial part of the Jominy curve, but a far higher accuracy around the “flex point” on the profile, where the shape uncertainty is highest. The underperformance of the sequential and DDN estimators can be attributed to the limited number of data held by DS_B and used for networks training and validation, a limitation that is overcome by the proposed TL approach, which leverages valuable knowledge from the pretrained decoder, which is also computationally more efficient than a standard DNN, as only the encoder part needs to be retrained.

More importantly, it is not necessary that all the data included in the database adopted for the TL training are as numerous and as “complete” in terms of Jominy profile samples as the dataset used for the first training stage. In other words, DS_B can contain less data than DS_A and Jominy profiles comprising (reasonably) less than 15 hardness measures, as the latent space L in the pretrained network already codifies the main elements of the curve shape and the second training is mostly focused on mapping the steel metallurgy in such latent space. This feature makes the proposed TL approach very suitable to deployment in an industrial environment, as the estimator can straightforwardly be adapted to the evolution of the production while preserving the information acquired through its usage.

The training time of the entire model is affordable and fully compatible with practical applications. All tests were performed

on the same computing device, i.e., a standard PC holding an Intel CORE i9 8th generation processor with 8 GB RAM. The total training time with the available data and the identified optimal configuration is 75–80 s (about 60% for AE_Jom training), and such figure, compared to other literature approaches, is not particularly high. Training times for the BaseMLP model vary between 40 and 50 s, taking advantage of the low complexity of this approach, which corresponds to a limited number of parameters to be updated during the learning phase. The time needed for training the pure DNN is about 70–75 s, thus comparable with the proposed solution. The sequential model, on the other hand, requires a longer training time of around 90–120 s, probably due to the overheads of creating various NNs and handling the information they exchange. In the TL context proposed in this work, it is assumed that only the DNN_Jom needs to be trained, where the decoder part parameters are frozen and, thus, do not participate in the training, effectively streamlining the training process. In this work, where the dataset used for assessing the TL capabilities consists of a limited number of samples, this process takes about 8–10 s. Finally, the volume of the data that can be collected in the industrial practice concerning a steel grade/family prior of a retraining of the system, even in large steelmaking companies, is in the order of the hundreds or maximum a few thousands, and therefore the training time is foreseen to be in the order of minutes even in the worst case. As the training operation is not frequent, such time range is fully adequate for practical purposes.

In the perspective of providing a practical tool to be used in many steelworks worldwide to replace the Jominy end-quench test and/or as a component of a more complex system supporting the search or design of the most suitable steel metallurgy to achieve a desired shape of the Jominy profile, the exploitation of data coming from costly and/or not widespread analytical systems, such as, for instance, the one proposed by Akhlaghi et al.^[41] is not advisable, as the generation of a suitable database would be expensive and time-consuming and the system would be usable only in companies where the needed equipment is available. Indeed, although not yet explored, the integration of information concerning local crystal orientation or metallurgy associated to the hardness values to be estimated is expected to improve the accuracy of the model, also considering the outcomes of some recent literature results.^[61] In effect, the metallurgy data that are used in the proposed approach are process data taken on a liquid steel sample and not from the Jominy specimen. On the other hand, this requires additional analytical efforts and costs, which might balance the savings that are achievable with the ML-based solution. Therefore, the practical and economic viability of this kind of approach must be carefully assessed and goes beyond the scope of the present work.

5. Conclusion

A novel DL approach to estimate the hardenability profiles of medium-carbon quench hardenable steels based on steel metallurgy is proposed, which exploits a monodimensional CNN and an AE and is applicable to different steel grades. The basic idea behind such estimator is to correlate the chemical composition to the whole shape of the curve rather than to the single points. The

main contribution of the estimator is twofold: 1) the superior performances it exhibits compared to other efficient NN-based approaches, which is enabled by DL; 2) the capability to elaborate a compact representation of the curve, which is automatically learned and stored in a first stage, and is not related to specific analytical model, such as in traditional parametric approaches, which might be suitable to a narrower range of steel grade but is hard to generalize and extend. Such capability is exploited in a TL approach, which enables adaptation of the estimator to different steel grades with a limited number of additional data.

Therefore, the estimator is very suitable for practical use in an industrial context, as its extension to a wide range of steel grades can be implemented gradually and at limited computational and experimental cost.

Ongoing and future work focuses on the exploitation of the estimator in an optimization context, where, given a target Jominy profile, the steel metallurgy is computed which ensures its achievement by possibly minimizing the addition of costly microalloying elements. Moreover, the possibility will be investigated of a direct infusion of physical knowledge in the component of the estimator relating the steel metallurgy to the compact representation of the Jominy curve through hybrid AI approaches such as, for instance, physics-informed NNs.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available in the Supporting Information of this article.

Keywords

autoencoders, convolutional neural networks, deep learning, hardenability, Jominy, medium-carbon quench hardenable steels

Received: June 9, 2023

Revised: October 17, 2023

Published online:

- [1] V. Colla, *IFAC-PapersOnLine* **2022**, 55, 1.
- [2] O. Isayev, A. Tropsha, S. Curtarolo, *Materials Informatics: Methods, Tools, and Applications*, John Wiley & Sons, Hoboken, NJ **2019**.
- [3] K. G. Reyes, B. Maruyama, *MRS Bull.* **2019**, 44, 530.
- [4] K. Frydrych, K. Karimi, M. Pecelerowicz, R. Alvarez, F. J. Dominguez-Gutiérrez, F. Rovaris, S. Papanikolaou, *Materials* **2019**, 14, 5764.
- [5] M. Awtoniuk, D. Majerek, A. Myziak, C. Gajda, *Adv. Sci. Tech. Res. J.* **2022**, 16, 120.
- [6] M. Szala, L. Łatka, M. Awtoniuk, M. Winnicki, M. Michalak, *Processes* **2020**, 8, 1544.
- [7] L. Canale, L. Albano, G. E. Totten, L. Meekisho, *Nature* **1947**, 159, 853.
- [8] M. Umemoto, N. Komatsubara, I. Tamura, *J. Heat Treat.* **1980**, 1, 57.
- [9] E. J. Whittenberger, R. R. Burt, D. J. Carney, *JOM* **1956**, 8, 1008.
- [10] ASTM Standard A 255, Standard Test Method for End-Quench Test for Hardenability of Steel.

- [11] M. A. Grossmann, *Trans. TSS-AIME* **1942**, 150, 227.
- [12] E. Just, *Met., Prog.* **1969**, 96, 87.
- [13] R. A. Grange, *Metall. Trans.* **1973**, 4, 2231.
- [14] G. T. Brown, B. A. James, *Metall. Trans.* **1973**, 4, 2245.
- [15] C. T. Kunze, J. E. Russel, in *Hardenability Concepts with Application to Steel* (Eds: D. V. Doane, J. S. Kirkaldy), Chicago, IL **1977**, p. 290.
- [16] D. V. Doane, in *Hardenability Concepts with Application to Steel* (Eds: D. V. Doane, J. S. Kirkaldy), Chicago, IL **1977**, p. 351.
- [17] G. T. Eldis, W. C. Hagel, in *Hardenability Concepts with Application to Steel* (Eds: D. V. Doane, J. S. Kirkaldy), Chicago, IL **1977**, p. 397.
- [18] J. Komenda, R. Sandström, M. Tukiainen, *Steel Res.* **1997**, 68, 132.
- [19] W. Gong, Z. Jiang, D. Zhan, *Adv. Mater. Res.* **2011**, 233–235, 2352.
- [20] J. Sponzilli, C. Jatczak, J. S. Kirkaldy, T. Ericsson, G. Eldis, in *Hardenability Concepts with Application to Steel* (Eds: D. V. Doane, J. S. Kirkaldy) Chicago, IL **1977**, p. 483.
- [21] J. S. Kirkaldy, D. Venugopalan, in *Proc. Int. Conf. Phase Transformations in Ferrous Alloys*, AIME, Philadelphia, PA **1984**, p. 125.
- [22] N. Saunders, X. Li, A. P. Miodownik, J.-Ph. Schillé, in *Materials Design Approaches and Experiences* (Eds: J. -C. Shao, T. M. Pollock, M. J. Farhmann), TMS, Warrendale, PA **2001**, pp. 185–197.
- [23] B. H. Yu, *Iron Steel* **1985**, 20, 40.
- [24] Y. B. Hai, *Steels Computational Design*, Mechanical Industrial Press, Beijing **1996**.
- [25] Y. Song, G. Liu, S. Liu, J. Liu, C. Feng, *J. Iron Steel Res. Int.* **2007**, 14, 37.
- [26] M. Jin, J. S. Lian, Z. H. Jiang, *Acta Metall. Sin.* **2006**, 42, 265.
- [27] P. A. Rometsch, M. J. Starink, P. J. Gregson, *Mater. Sci. Eng. A Struct.* **2003**, 339, 255.
- [28] Zehtab Yazdi, S. A. Sajjadi, S. H. Zebarjad, S. H. Nezhad, *J. Mater. Process. Technol.* **2008**, 199, 124.
- [29] B. Chan, M. Bibby, N. Holtz, *Can. Metall. Q.* **1995**, 34, 353.
- [30] W. G. Vermeulen, P. J. Van Der Wolk, A. P. De Weijer, S. Van Der Zwaag, *J. Mater. Eng. Perform.* **1996**, 5, 57.
- [31] L. A. Dobrzański, W. Sitek, *J. Mater. Process. Technol.* **1998**, 78, 59.
- [32] L. A. Dobrzański, W. Sitek, *J. Mater. Process. Technol.* **1999**, 92–93, 8.
- [33] M. Knap, J. Falkus, A. Rozman, J. Lamut, *Arch. Metall. Mater.* **2008**, 53, 509.
- [34] M. Knap, J. Falkus, A. Rozman, K. Konopka, J. Lamut, *Arch. Metall. Mater.* **2014**, 59, 133.
- [35] H. Pouraliakbar, M.-J. Khalaj, M. Nazerfakhari, G. Khalaj, *J. Iron Steel Res. Int.* **2015**, 22, 446.
- [36] V. Colla, L. M. Reyneri, M. Sgarbi, *Integr. Comput. Aided Eng.* **2000**, 7, 217.
- [37] S. Cateni, V. Colla, M. Vannucci, M. Vannocci, in *Proc. IASTED Int. Conf. Artificial Intelligence and Applications*, IASTED, Calgary, Canada **2013**, p. 169.
- [38] V. Colla, M. Vannucci, L. Bacchi, R. Valentini, *Metall. Ital.* **2020**, 112, 47.
- [39] W. Sitek, A. Irla, *Arch. Metall. Mater.* **2016**, 61, 797.
- [40] S. Smokvina Hanza, T. Marohnić, D. Iljkić, R. Basan, *Metals* **2021**, 11, 714.
- [41] Akhlaghi, S. Kahrobaee, A. Akbarzadeh, M. Kashefi, T. W. Krause, *Nondestr. Test. Eval.* **2020**, 36, 459.
- [42] L. Xu, J. S. J. Ren, C. Liu, J. Jia, *Adv. Neurol.* **2014**, 2, 1790.
- [43] M. Vannocci, A. Ritacco, A. Castellano, F. Galli, M. Vannucci, V. Iannino, V. Colla, *Lect. Notes Comput. Sci.* **2019**, 11507 LNCS, 220.
- [44] Zhang, S. Li, Y. Yan, Z. Ni, H. Ni, *Steel Res. Int.* **2022**, 93, 5.
- [45] Yi, G. Li, M. Jiang, *Steel Res. Int.* **2017**, 88, 176.
- [46] Xie, W. Li, L. Fan, M. Zhou, *Steel Res. Int.* **2023**, 94, 2200204.
- [47] Wei-Gang, C. Jing-Cheng, F. Li-Xia, X. Lu, *J. Iron Steel Res. Int.* **2020**, 32, 33.
- [48] Z. Wang, Y. Bao, C. Gu, *Steel Res. Int.* **2023**, 94, 2200342.
- [49] G. W. Song, B. A. Tama, J. Park, J. Y. Hwang, J. Bang, S. J. Park, S. Lee, *Steel Res. Int.* **2019**, 90, 1900321.

- [50] D. Wójcik, T. Rymarczyk, B. Przysucha, M. Gołabek, D. Majerek, T. Warowny, M. Soleimani, *Energies* **2023**, *16*, 1387.
- [51] Z. Jiang, J. Zhu, D. Pan, H. Yu, K. Zhou, W. Gui, *Steel Res. Int.* **2022**, *94*, 2200680.
- [52] S. Youkachen, M. Ruchanurucks, T. Phatrapomnant, H. Kaneko, in *Proc. 10th Int. Conf. Information and Communication Technology for Embedded Systems*, IEEE, Los Alamitos, CA **2019**, p. 8695928, <https://doi.org/10.1109/ICTEmSys.2019.8695928>.
- [53] J. Liu, J. Wu, Y. Xie, W. Jie, P. Xu, Z. Tang, H. Yin, *J. Ind. Inf. Integr.* **2022**, *30*, 100410.
- [54] Q. Yang, Y. Zhang, W. Dai, S. J. Pan, *Transfer Learning*, Cambridge University Press, Cambridge, UK **2020**.
- [55] A. Bouguettaya, Z. Mentouri, H. Zarzour, *Int. J. Adv. Des. Manuf. Tech* **2023**, *125*, 5313.
- [56] H.-V. Nguyen, J.-H. Bae, Y.-E. Lee, H.-S. Lee, K.-R. Kwon, *Sensors* **2022**, *22*, 9926.
- [57] A. Boudiaf, K. Harrar, S. Benlahmidi, R. Zaghoudi, S. Ziani, S. Taleb, in *Proc. 7th Int. Conf. Image and Signal Processing and their Applications*, IEEE, Los Alamitos, CA **2022**, <https://doi.org/10.1109/ISPA54004.2022.9786361>.
- [58] R. F. Ivo, D. de A. Rodrigues, G. M. Bezerra, F. N. C. Freitas, H. F. G. de Abreu, P. P. Rebouças Filho, *J. Mater. Res. Technol.* **2020**, *9*, 8580.
- [59] E. Kim, K. Huang, A. Saunders, A. McCallum, G. Ceder, E. Olivetti, *Chem. Mater.* **2017**, *29*, 9436.
- [60] J. Oliveira, I. Praca, *IEEE Access* **2021**, *9*, 9699.
- [61] K. Karimi, H. Salmenjoki, K. Mulewska, L. Kurpaska, A. Kosińska, M. J. Alava, S. Papanikolaou, *Scr. Mater* **2023**, *234*, 115559.