INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA E GESTÃO DA CADEIA DE ABASTECIMENTO




DEVELOPMENT OF A MODEL TO FORECAST TRANSPORT COSTS IN MODULAR CONSTRUCTION

MARIA JOSÉ RIBEIRO DA SILVA PEREIRA
outubro de 2024

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

DEVELOPMENT OF A MODEL TO FORECAST TRANSPORT COSTS IN MODULAR CONSTRUCTION

Maria José Ribeiro da Silva Pereira

2024

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Mecânica

P.PORTO


P.PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

DEVELOPMENT OF A MODEL TO FORECAST TRANSPORT COSTS IN MODULAR CONSTRUCTION

Maria José Ribeiro da Silva Pereira

Student Number 1220291

Dissertation submitted to the Instituto Superior de Engenharia do Porto to fulfill the requirements for a Master's Degree in Engineering and Supply Chain Management, under the supervision of Doctor Marisa João Guerra Pereira De Oliveira and co-supervision of Doctor Maria Teresa Ribeiro Pereira.

2024

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Mecânica

P.PORTO


P.PORTO

“Apenas quando somos instruídos
pela realidade é que podemos
mudá-la."
Bertolt Brecht


P.PORTO


isep

ACKNOWLEDGEMENTS

À Professora Doutora Marisa Guerra Pereira, a minha orientadora, das melhores professoras com quem já me cruzei no meu percurso académico, com os mais vastos conhecimentos e talento para o ensino. A que tornava as aulas de sexta à noite em momentos divertidos e leves.

À Professora Doutora Maria Teresa Pereira, a minha co-orientadora, muito rica em conhecimentos, com a maior vontade em ver os seus alunos prosperar e sempre pronta a ajudar. Uma professora amiga.

Ao Professor Doutor Eduardo Oliveira, o meu orientador no INEGI, que me transmitiu ensinamentos muito importantes, aconselhando-me durante todo o caminho e desafiando-me todos os dias a melhorar e explorar as minhas capacidades.

A toda a equipa do INEGI, em especial ao GEIN/EC, por me terem acompanhado neste desafio e por tornarem esta experiência ainda melhor. Sempre divertidos e acolhedores, mostraram-me o quão importante é cruzarmo-nos com boas pessoas no mundo profissional.

Aos meus Pais, pelo amor, pela força e por serem o meu porto seguro, que sempre me ampararam e guiaram, deixando-me voar e seguir os meus sonhos. A pessoa que sou hoje a vocês o devo.

À minha irmã Ana, a minha melhor amiga, a pessoa que me inspira todos os dias e que me mostra o que é o amor na sua forma mais pura.

Ao Salvador, o meu amor de quatro patas e confidente de muitas horas, que esteve sempre ao meu lado.

Aos meus Amigos de sempre e aos que fiz nesta jornada, pela amizade e apoio incondicional, por tornarem todo o meu percurso académico mais brilhante e pela motivação que sempre me deram. Os momentos que convosco vivi jamais esquecerei.

A todos os que já cá não estão, mas que sei que olharam sempre por mim.

A Deus, aos meus Guias e aos meus Anjos, por me iluminarem todos os dias.

blank page

ABSTRACT

The construction sector, one of the largest global industries, significantly impacts the economy of any country. Modular Construction (MC) emerges as a solution to the inefficiencies of conventional construction, which still faces challenges in productivity, sustainability, and safety. With innovative techniques, MC addresses critical issues and improves logistics operations, optimizing processes and enhancing the efficiency and sustainability of projects, thereby revolutionizing traditional construction methods and gaining global prominence.

With the growing interest in MC, it became essential to gather relevant data on the adoption of this method in the construction sector. The use of Artificial Intelligence is crucial for collecting and analyzing these data, especially regarding transportation costs, a field still underexplored. This information gap led to the investigation of Machine Learning (ML) methods to predict transportation costs in MC.

In this study, two ML models were developed and evaluated, namely Deep Learning (DL), with different data preprocessing techniques, aiming to predict transportation costs in MC. To address this need, the implementation of ML techniques was explored and applied to two different scenarios: road transportation and sea transportation. The models were validated using specific data preprocessing techniques.

The model that performed best was the one applied to road transportation, which, after applying a specific filter for full truckload shipments, achieved a Mean Absolute Percentage Error of 9.3%. This result is due to the model's ability to capture the dynamics of the full truckload market, reducing the impact of market fluctuations and improving prediction accuracy. Conversely, the model applied to sea transportation faced significant challenges due to the complexity and volatility of costs in this sector, resulting in a higher Mean Absolute Percentage Error of 45.78%. Further analysis revealed that it was not the model architecture that limited its performance, but rather the commercial complexities of maritime transport.

The work done with raw material transportation data provides a solid foundation for future research, where the application of transfer learning is expected to further enhance the model's predictive capability on modules transportation data.

KEYWORDS

Deep Learning, Modular Construction, Transportation Cost Prediction, Transfer Learning.

blank page

Resumo

O setor da construção, uma das maiores indústrias a nível global, tem um impacto significativo na economia de qualquer país. A Construção Modular surge como uma solução para as ineficiências da construção convencional, que ainda enfrenta desafios de produtividade, sustentabilidade e segurança. Através de técnicas inovadoras, a Construção Modular aborda questões críticas e melhora as operações logísticas, otimizando processos e aumentando a eficiência e sustentabilidade dos projetos, o que está a revolucionar o método tradicional de construção e a ganhar destaque a nível global.

Com o crescente interesse pela Construção Modular, tornou-se essencial reunir dados relevantes sobre a adoção deste método no setor da construção. A utilização de Inteligência Artificial é crucial para a recolha e análise desses dados, especialmente no que diz respeito aos custos de transporte, um campo ainda pouco explorado. Esta carência de informação levou à investigação de métodos de *Machine Learning* (*ML)* para prever os custos de transporte na Construção Modular.

Neste estudo, foram desenvolvidos e avaliados dois modelos de *ML*, *nomeadamente de Deep Learning (DL),* com diferentes técnicas de pré-processamento de dados, e com o objetivo de prever os custos de transporte na Construção Modular. Para abordar essa necessidade, foi explorada a implementação de técnicas de *ML*, que foram aplicadas em dois cenários distintos: transporte terrestre e transporte marítimo. Os modelos foram validados através de técnicas específicas de pré-processamento de dados.

O modelo que se destacou com o melhor desempenho foi o aplicado ao transporte terrestre, que, após a aplicação de um filtro específico para cargas de camião completo, alcançou um Erro Percentual Absoluto Médio de 9,3%. Este resultado deve-se à capacidade do modelo em capturar a dinâmica do mercado de camiões completos, reduzindo o impacto das flutuações de mercado e melhorando a precisão das previsões. Por outro lado, o modelo aplicado ao transporte marítimo enfrentou desafios significativos devido à complexidade e volatilidade dos custos neste setor, resultando num Erro Percentual Absoluto Médio mais elevado de 45,78%. Análises mais detalhadas sublinham que não foi a arquitetura do modelo que limitou o seu desempenho, mas sim as complexidades comerciais do transporte marítimo.

O trabalho efetuado com dados de transporte de matérias-primas fornece uma base sólida para investigação futura, onde se espera que a aplicação de *transfer learning* melhore ainda mais a capacidade de previsão do modelo em dados de transporte de módulos.

PALAVRAS-CHAVE

Deep Learning, Construção Modular, Previsão de custos de transporte, *Transfer Learning*.

blank page

INDEX

blank page

INDEX OF FIGURES

blank page

TABLE INDEX

blank page

LIST OF ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
CWGT	Chargeable Weight
CRISP-DM	Cross Industry Standard Process for Data Mining
CUDA	Compute Unified Device Architecture
DL	Deep Learning
DNN	Deep Neural Networks
DSRM	Design Science Research methodology
DTL	Deep Transfer Learning
ELU	Exponential linear unit
FTL	Full Truckload
INEGI	Institute of Science and Innovation in Mechanic Engineering and Industrial Engineering
ISEP	Instituto Superior de Engenharia do Porto
LTL	Less than Truckload
KCV	K-fold Cross-Validation
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MEP	Mechanical, plumping, and electrical
ML	Machine Learning
NGBoost	Natural Gradient Boosting
NNM	Neural Networks Model
PPVC	Prefabricated Prefinished Volumetric Construction
P.Porto	Polytechnic Institute of Porto
PReLU	Parametric leaky version of rectified linear unit
PRR	Plan of Recovery and Resilience
ReLU	Rectified linear unit
RMSE	Root Mean Squared Error
SHAP	Shapely Additive Explanations
UDA	Universal domain adaptation

blank page

1. INTRODUCTION

Unlike the methods used in conventional construction, where all the work is done on site, in Modular Construction (MC) the buildings are fabricated in modules offsite and then assembled onsite. This innovative method is still penetrating the market, but due to its efficiency, sustainability and cost-effectiveness it is gaining popularity globally. However, the logistics and transportation costs involved in MC can be complex and difficult to predict, since in this method it is necessary to transport large, prefabricated modules that entail logistical challenges. Projecting these transportation costs accurately is fundamental to the overall budgeting and planning of modular construction projects. While the adoption of modular construction is increasingly widespread, there is a notable shortfall in research and detailed models focused on predicting these very specific costs. (Bertram et al., 2019; Thai et al., 2020; Weiss et al., 2016).

This thesis aims to bridge this gap by developing a predictive model to forecast transportation costs of raw materials in modular construction. To estimate transportation costs for a different type of goods, a predictive model will be developed. As for future work, a modular construction model can employ transfer learning, a technique in which a pre-trained model for one task - in in this case, the model developed for other goods - is then adapted to a related task. The aim of using transfer learning is to transfer the knowledge learned from forecasting the transportation costs of other types of goods to the specific context of modular construction. This approach is particularly relevant, since modular construction may have little relevant data on transportation costs, while more detailed data is available for the transportation of other goods.

The motivation underlying this theme derives from the following key drivers:

- → Economic Efficiency.
- → Planning and Budgeting.
- → Sustainability.
- → Technological Advancement.
- → Industry Demand.

By attempting to close the gap in transportation cost predictions for modular construction, and opening way for the use of transfer learning, this thesis aims to provide a valuable tool that can increase efficiency, planning accuracy and the overall sustainability of the project.

The introductory chapter lays the groundwork for the entire thesis. It will begin by identifying the research problem and explaining its significance within a broader context. Following this, the research questions and objectives will be clearly defined, serving as a guide for what the study intends to accomplish. The chapter then will explore the different methodological options considered, justifying the chosen approach. Additionally, a concise overview of the company involved in the study will be provided to give necessary background information. The chapter will end with a description of the document structure, giving a preview of the upcoming chapters.

1.1. Framework and relevance

Modular construction is an area that is arising nowadays mainly because of the growing need for housing. The modular construction approach has three main steps: module fabrication in the factory, transportation of the final modules to the construction site and finally the assembly of the modules onsite. Being a very recent topic of interest, many studies and research have been made regarding the first and last step, however the transportation step does not have the same amount of relevant information and, consequently, there exists very little information regarding the modules' transportation and how the modules' dimensions impact the transportation costs. This dissertation focuses on this step, more precisely on the transportation costs associated with modular construction, which may need special transport due to the dimensions of the modules. Thus, the need to develop a predictive model to forecast the transportation cost that comes with this new concept became urgent. (Bertram et al., 2019; Thai et al., 2020).

By predicting the transport costs for the modules before building them, considering the module specifications, it is possible to understand what changes can be made to minimize the total transport costs and test those changes beforehand. Thus, the attention is led to the first step, since in the first step it is possible to optimize the sizes and forms of the modules so that the costs of transportation are minimized.

However, there are little to no data regarding transportation costs in modular products, which makes the target domain poor in labels and conditions the development of prediction models. That’s why transfer learning is proposed to solve this issue. Through the vast amount of data regarding transportation costs in other type of goods, also denominated source domain, it is possible to cover the lack of data concerning the target. These techniques will serve as a bridge between the differences of the two domains, using the information on the transportation cost of other goods to help predicting the cost of transporting the modules. These two domains are very similar in terms of features so transfer learning is expected to easily extract the needed knowledge

from the source to the target. So, the prediction of transportation costs of raw materials in modular construction emerges as an intermediate step to achieve the main goal (Torrey, 2010; Weiss et al., 2016). This solution is quite relevant since to the best of our knowledge, this is the first study that studies the development of a model to forecast the transportation costs of prefabricated products, hence its relevance to the scientific community (Sousa et al., 2023).

1.2. Research question and objectives

After the presentation and framing of the problem and its relevance, the following research questions exposed on Table 1 have arisen:

Table 1 - Research Questions

Research Question 1	What are the key factors that influence the transferability of knowledge between different transportation domains?
Research Question 2	How does the available data on transport costs for other goods contribute to forecasting transport costs in modular construction?
Research Question 3	How can deep learning models for predicting transportation costs of other goods be optimized based on performance analysis and data-driven improvements?

For these questions to be answered, it is necessary to fulfil specific objectives, such as:

- → Literature review of transfer learning methods for tabular data, focused on domain shift, to determine the necessary model requirements. Analysis and improvement of the existing Deep Learning (DL) models developed for predicting the transportation costs of other type of goods.
- → Development of a predictive model for the transportation cost of modular construction's raw materials, based on deep learning.

Such objectives will also enable the accomplishment of the general objective of utilizing the knowledge learned through the development of predictive models focused on the transportation cost of raw materials in MC, to facilitate the future development of predictive models for modules' transportation cost.

1.3. Methodological options

To analyze, investigate and interpret data, there are a variety of research methodologies that can be utilized. Key methodological options include quantitative research, qualitative research, mixed-methods research (combination of quantitative and qualitative approaches) and experimental research (focused on cause-and-effect relationships). These traditional scientific methods are appropriate for studying phenomenon and testing hypotheses in established frameworks of knowledge (Williams, 2007).

However, these methods may not be well suited to contexts that require creativity, practicality, cross-disciplinary collaboration, and a focus on practical solutions to real-world problems. Design Science Research methodology (DSRM) is a very useful methodology that becomes very valuable when investigating more practical problems. This methodology aims to design, develop, and evaluate artifacts, such as software, methods, or models to generate knowledge. Peffers (2007) created a DSRM process that follows specific steps, as can be seen on Figure 1:

1) **Problem identification and motivation**: Define the actual research problematic and its relevance and the necessary performance of the proposed solution.
2) **Define the objectives**: Specify the required objectives needed to accomplish the chosen solution.
3) **Design and development**: Define the desired functionality and architecture of the artifact and then create it.
4) **Demonstration**: Demonstrate that the artifact solves the problem.
5) **Evaluation**: Measure the observed results and compare them with the objectives.
6) **Communication**: Communicate all the necessary information, problem, solution, and effectiveness.

Since the problem addressed in this project do not have a clear solution or precedent, and which objective is to utilize the knowledge generated by predictive models focused on other type of goods to facilitate the learning of predictive models for modules' transportation cost, it was decided to follow the DSR methodology.

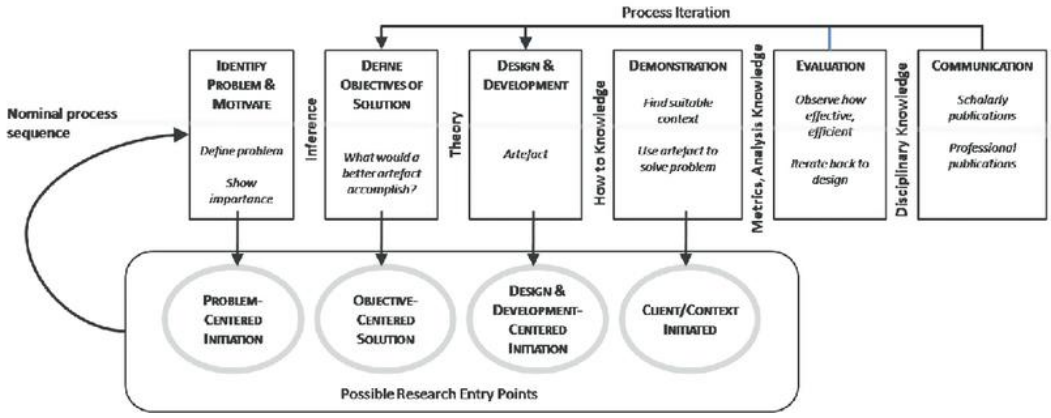

Figure 1 - DSR Methodology.

Source: Peffers (2007)

In parallel, it will be used Cross Industry Standard Process for Data Mining (CRISP-DM). This is a project management framework often used by scholars and experts in data science projects (Saltz & Hotz, 2020).

In Figure 2 are shown the five iterative steps of CRISP-DM:

- → **Business Understanding**: Understand the problem and the customer's needs.
- → **Data Understanding**: Explore the original data without any modification in order to form the first insights.
- → **Data preparation**: Clean and transform the data into the final dataset.
- → **Modelling**: Select algorithms that will be used in the models.
- → **Evaluation**: Check through evaluation metrics if the model's performance meets the needs defined previously.
- → **Deployment**: Deployment of the models to be used by the end customers.

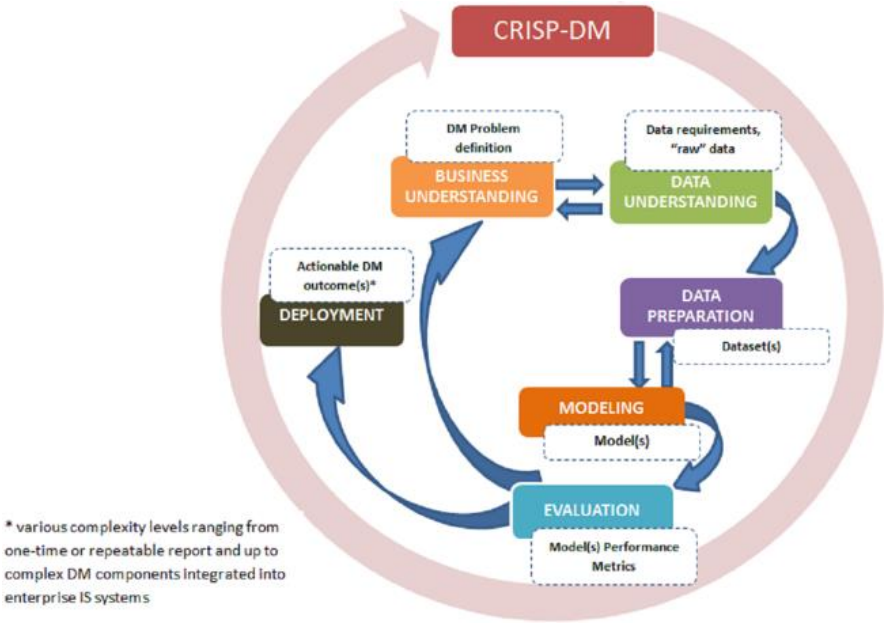

Figure 2 - CRISP-DM steps.

Source: Plotnikova et al. (2020)

1.4. Company presentation

INEGI – Institute of Science and Innovation in Mechanic Engineering and Industrial Engineering is a Center of Technology and Innovation that aims to add, through valuable investigation activities, technological innovation, technology transfer, technological consulting and services, precious contributions for the development of industry and economic in general. Their infrastructure is placed in the Campus of the Faculty of Engineering of University of Porto. All the work developed by INEGI has the goal to increase the efficiency level in operations and process of their clients. This work is developed in the context of a research initiation grant in INEGI, included in the Management and Industrial Engineering area, more specifically on the R2U Technologies / modular system project (INEGI, 2024).

R2U Technologies / modular system is a project approved by the Plan of Recovery and Resilience (PRR), initiated in January of 2023. This project aims for the creation of an industrial cluster for modular construction, where the modules are developed in the factory and posteriorly sent to the construction work. This project involves many other companies, namely a large construction company and a large logistics operator, both from the north of Portugal.

1.5. Document structure

Below is a detailed description of each chapter's content and purpose, providing a comprehensive overview of the organization and structure of this thesis.

The second chapter discusses the existing literature and current developments in the relevant fields. It begins with an exploration of modular construction, reviewing its principles and applications. Next, the chapter covers Machine Learning (ML) and Deep Learning, with a particular focus on neural networks. In addition, the chapter discusses transfer learning and provides an overview of deep learning libraries used in the implementation phase. Finally, the chapter addresses the application of AI in modular construction, by connecting the technologies discussed with the research context.

The third chapter describes the methodological approach and the practical application of the research. It begins with the data understanding and data preparation steps, detailing the sources and characteristics of the road transport dataset and the sea transport dataset, as well as the techniques used to prepare the data. The chapter continues with the training and modeling phase, explaining the techniques and processes used to train, develop and refine the models

Chapter four presents the results obtained in the modelling and application phase. It follows with a discussion of the implications of these results, and the potential impact in the field.

The final chapter summarizes the main conclusions and contributions of the research. It reflects on the research questions and answers them based on the research findings. The chapter also discusses the limitations of the study and suggests areas for future research, providing a way forward for ongoing research in this area.

blank page

2. LITERATURE REVIEW

In the next subchapters will be reviewed the literature in pertinent fields. The chapter starts with the exploration of the principles and practical applications in modular construction. Next, will be reviewed both machine learning and deep learning topics, with a special focus on neural networks. Then it is addressed transfer learning, and it is also provided an overview of deep learning libraries used during the implementation phase. Finally, it will be investigated the use of AI in modular construction, connecting the discussed technologies to the research context.

2.1. Modular Construction

Conventional construction, that uses brick walls, is still the most utilized construction method since it has lower labor cost, does not require changes that companies are not ready to start doing due to the necessary expenses, and does not need qualified workers to deal with new processes and the needed technology for the transition to modular construction. As the construction industry is one of the factors with the biggest impact in a country's economy, the visible low productivity in conventional construction led to greater attention to modular construction (Liew et al., 2019).

Modular construction or prefabricated home buildings, first related in 1624 when England sent houses to the now Massachusetts state, is becoming a popular concept around consumers and builders. This popularity is mainly because this process can reduce waste and construction time in comparison to traditional methods. Hence, the adaptation of modular construction can save time and labor (Altaf et al., 2018; Boafo et al., 2016).

Modular construction consists of building the elements off-site and then assemble them on construction sites. In Figure 3 it can be seen an example of a Prefabricated Prefinished Volumetric Construction (PPVC) process. PPVC is one type of modular construction where the structural frame (walls, floors, and ceilings), mechanical, plumping, and electrical (MEP) elements are prefinished before the module assemble at site (Liew et al., 2019).

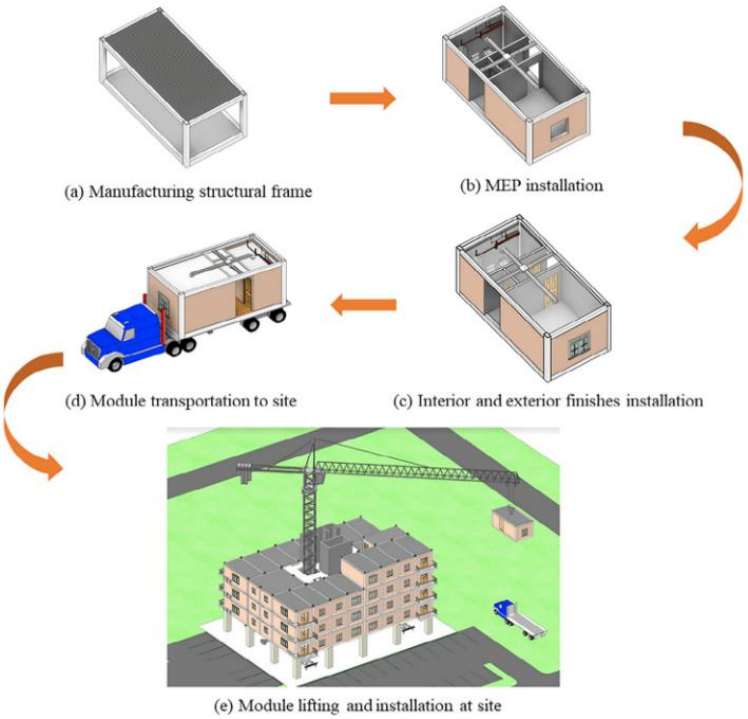

Figure 3 - Example of Prefabricated Prefinished Volumetric Construction.

Source: Liew et al. (2019)

Another very used type of modular construction is when most or all elements are only assembled in site, that is, the structural frame and MEP are only assemble in the local of the construction site, as can be seen in Figure 4 (Thai et al., 2020).

Figure 4 - Example of Modular construction process in site.

Source: Thai et al. (2020)

3D modules have two different types regarding load transfer mechanisms. For concrete building is commonly used load bearing walls, that have the purpose to transfer gravity to the base and resist to the lateral loads (see Figure 5). The other type, illustrated in Figure 6, utilizes steel corner supported modules, where the transferred gravity loads go to the slab, edge beams, corner columns and base (Liew et al., 2019).

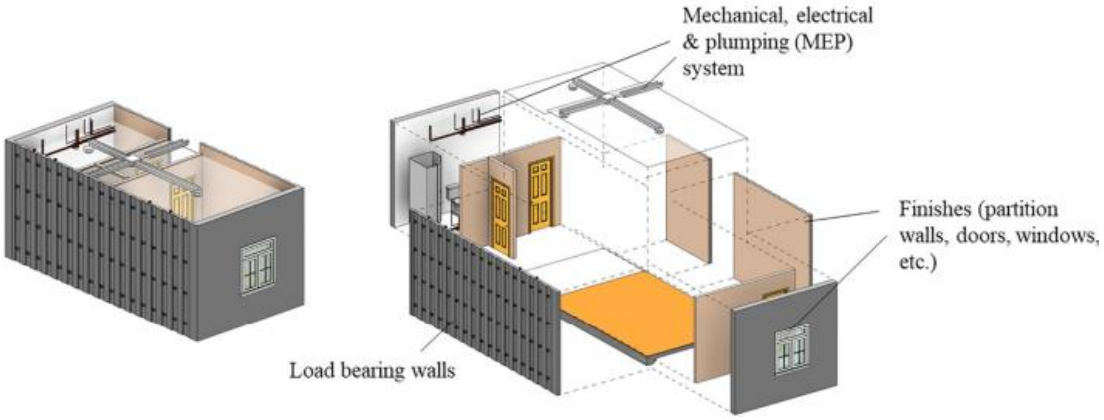

Figure 5 - Load bearing system.

Source: Liew et al. (2019)

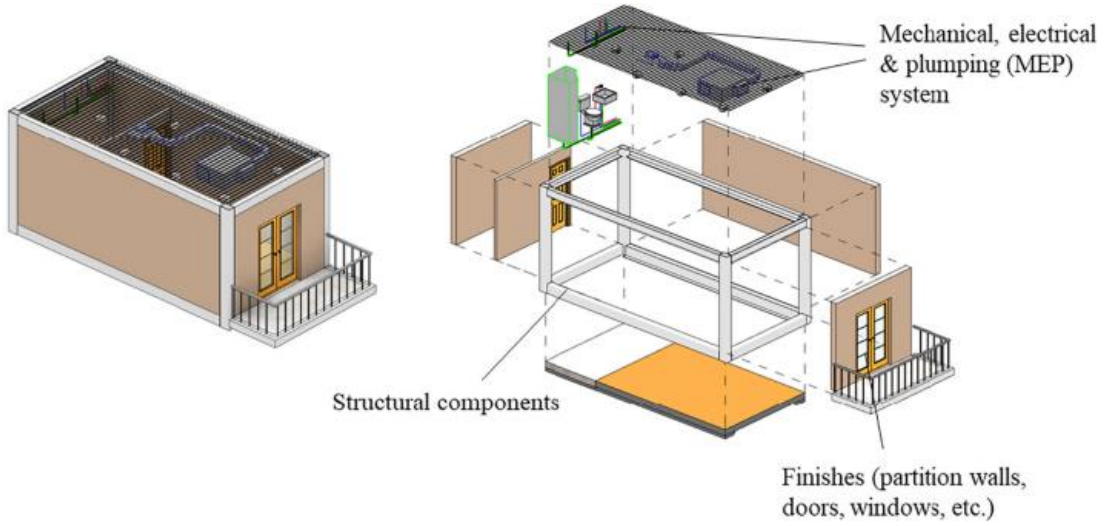

Figure 6 - Steel corner support system.

Source: Liew et al. (2019)

In terms of prefabrication degree classification, Moradibistouni et al. (2017) divided modular construction in five sub-categories (see Figure 7), namely: component, panel, volume, hybrid, and complete building.

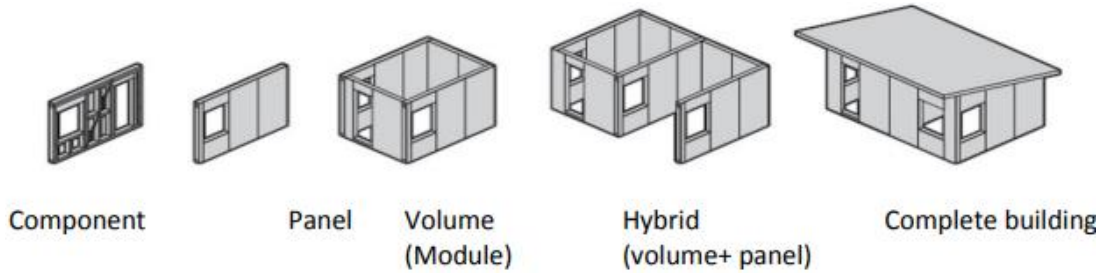

Figure 7 - Categories of prefabrication.

Source: Moradibistouni et al. (2017)

Based on Arturo Garza-Reyes et al. (2012), Lawson et al. (2012), Boafo et al. (2016) and Yi et al. (2020), the benefits of MC are:

→ Better controlled inventory.

→ Better protected materials.

→ Smaller number of suppliers.

→ Better and safer working conditions for the workers.

→ Less impacts on the environment.

→ Faster response from the maintenance team.

→ Faster building process and better manufacture quality.

However, the decision of adopting MC is influenced not only by the benefits of it but also by the balance between possible benefits and potential risks and limitations that come with its implementation. The study made by El-Abidi et al. (2015) lists some limitations regarding the implementation of modular construction in specific countries (see Table 2).

Table 2 - Limitations regarding the implementation of modular construction in specific countries

Negative Perception	- United Kingdom - European countries - Malaysia - Australia
Reduced Number of Skilled and Experienced Workers	- European countries - United States of America - Hong Kong - Australia
High Costs	- United Kingdom - Malaysia
Transport, Design and Government Support	- United Kingdom - United States of America - Malaysia - Hong Kong

Many sectors of the construction industry around the globe, but mainly Hong Kong, Japan, Singapore, and USA, are adopting modular construction due to its low building time, better quality, and productivity, namely in student residences, other type of residences, schools, offices, hotels, and hospitals, mainly structures with repeated components (Liew et al., 2019).

There are a vast number of studies regarding the adoption of MC in the Asian and North America markets, on the other hand the lack of information regarding the implementation of MC on European's markets is notable. Ribeiro et al. (2022) analyzed Portugal's current panorama, and concluded that, similar to other European countries, Portugal also fight some critical barriers to the adoption of MC. The main difficulties presented by the authors were the following ones:

→ Construction industry's reluctance to innovate.

→ Lack of accredited organizations to certify the quality of the fabricated modules.

→ Low levels of research and development systems in the construction industry.

Despite all the benefits mentioned previously, the construction sector still does not feel 100% safe adopting this new approach because of all the unknown risks that come with the needed adjustments. This aversion, consequence of the fact that MC is still an exception in most countries, contributes to the lack of data regarding MC (McKinsey, 2019).

The transportation of the modules benefits from ample availability of trucks and containers, supported by the logistics operator involved in this case study. The size and capacity of these trucks and containers characteristics are depicted in Figure 8 and Figure 9, respectively.

Measurements and Useful Load of Trucks

TYPOLOGY REFERENCE TIR TRUCKS	110 m^3 Jumbo Truck (Truck & Trailer)	100 m^3 Mega Truck	90 m^3 Semi-Trailer	Normal Semi-Trailer
Payload	24 Tons	25 Tons	25 Tons	25 Tons
Euro Pallets capacity	38 Euro PL	33 Euro PL	33 Euro PL	33 Euro PL
Useful Interior				
Length	7,45 m + 7,60 m	13,6 m	13,6 m	13,6 m
Width	2,45 m + 2,45 m	2,45 m	2,45 m	2,44 m
Height	2,50 m + 2,95 m	3 m	2,8 m	2,75 m

Figure 8 - TIR Trucks characteristics.

Source: Logistics operator

Measurements and useful load maritime containers

TIPOLOGY REFERENCE CONTAINERS SEA	Standard 20' x 8' x 8'6	Standard 40' x 8' x 8'6	Standard High Cube 40' x 9' x 9'6	Standard High Cube 45' x 8' x 9'6	Open Top 20' x 8' x 8'6	Open Top 40' x 8' x 8'6	Flat tracks Standard 20' x 8' x 8'6	Flat tracks / Platforms Standard 40' x 8' x 9'6
Payload	28,180 Tons	28,750 Tons	28,560 Tons	27,600 Tons	28,120 Tons	30,140 Tons	28,470 Tons	40 Tons
Pallet capacity	10 PL 1.2*1.0 11 PL 1.2*0.8	21 PL 1.2*1.0 25 PL 1.2*0.8	21 PL 1.2*1.0 25 PL 1.2*0.8	24 PL 1.2*1.0 27 PL 1.2*0.8				
Length	5,898m	12,032m	12,032m	13,556m	5,898m	12,024m	5,940m	12,132m
Width	2,352m	2,352m	2,352m	2,352m	2,345m	2,340m	2,345m	2,400m
Height	2,393m	2,393m	2,698m	2,698m	2,346m	2,244m	2,346m	2,135m
Door Opening (width)	2,340m	2,340m	2,340m	2,340m	2,300m	2,324m		
Door Opening (height)	2,280m	2,280m	2,585m	2,585m	2,215m	2,324m		
Roof Opening (width)					2,184m	2,184m		
Roof Opening (length)					5,492m	11,874m		

Figure 9 - Maritime Containers characteristics.

Source: Logistics operator

Figure 8 provides detailed information about the measurements and payload capacities of various trucks. The Jumbo truck, which includes both a truck and a trailer, has the largest volume. While the Mega truck has slightly less volume compared to the Jumbo truck, its length, width, and height make it more suitable for transporting larger modules. The Semi-Trailer and the Normal Semi-Trailer are very similar to each other and to the Mega truck, with the primary difference being in their height. Figure 9 presents comprehensive information regarding the measurements and payload capacities of maritime containers. All the containers have two variants. The two Standard containers have different lengths. The Standard High Cube containers also differ in their length and payload, in comparison to the previous ones these two can accommodate higher modules. The larger Open Top container has more than twice the length of the smaller one and have a significant higher payload capacity. Additionally, the larger Flat Track Standard container is also the one with the biggest payload capacity.

2.2. Machine Learning and Deep Learning

Artificial intelligence (AI) has sought to emulate the human brain. Machine learning is a subfield of Artificial Intelligence and Deep Learning is a subset of ML (Shinde & Shah, 2018).

ML has several learning algorithms, the most used ones are the following ones: Linear Regression, Logistic Regression, Naïve Bayes, Bayesian Network, Support Vector Machines, Decision Tree, Random Forest, AdaBoost, Bootstrapped Aggregation (Bagging), k-Nearest Neighbor and Artificial Neural Network (ANN). ML applications vary from computer vision, prediction, semantic

analysis, natural language processing to information retrieval. The numerous fields where ML can be implemented are shown in Figure 10.

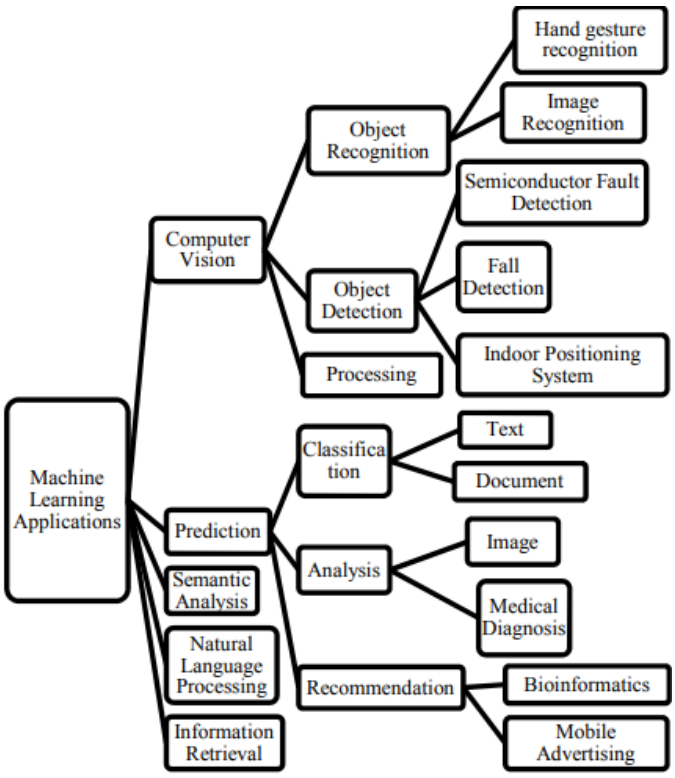

Figure 10 - Machine Learning applications.

Source: Shinde et al. (2018)

Deep learning is a ML concept based on neural networks models (NNM). Deep neural networks (DNN) have unavoidably more than one hidden layer and are organized on more complex and dense architectures, with more advanced operations. These deeper neural networks make it possible to obtain better results in comparison with the values obtained on shallow neural networks (that have only one hidden layer) when working with a high level of data (Cawley & Talbot, 2010).

Shinde et al. (2018) refer several deep learning techniques, such as: Convolutional Neural Networks, Recurrent Neural Networks, Distributed Representation, Autoencoder, Generative Adversarial Neural Network, Transformers, Diffusion Models and Variational Autoencoder. DL applications vary from computer vision, prediction, semantic analysis, natural language processing, information retrieval to customer relationship management. The numerous fields where DL can be implemented are shown in Figure 11. This study, in particular, will use deep learning for tabular data.

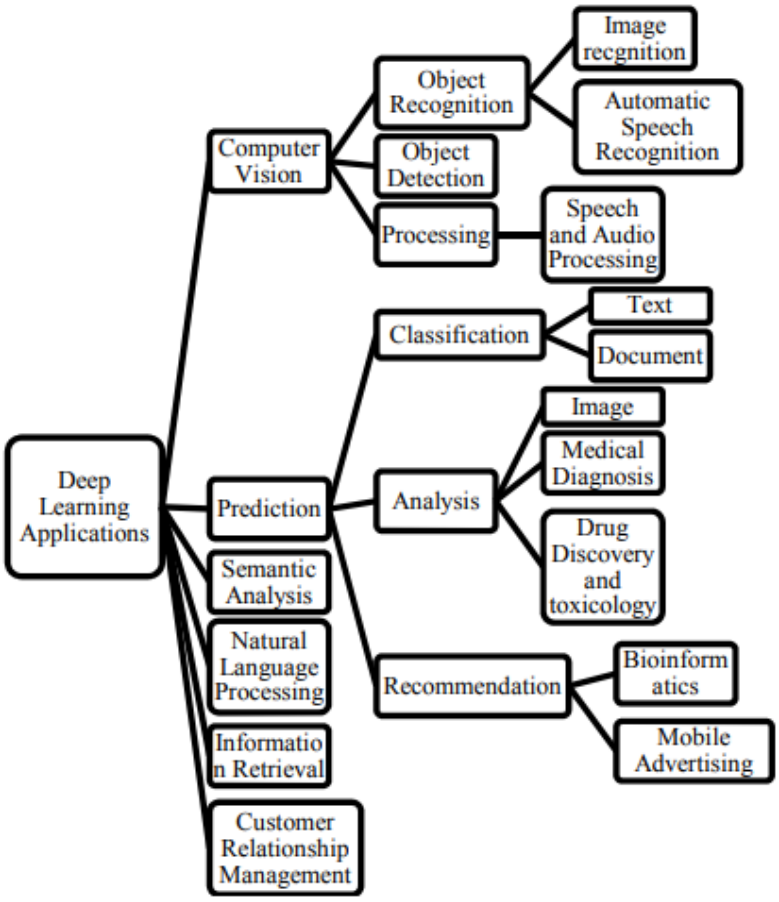

Figure 11 - Deep Learning applications.

Source: Shinde et al. (2018)

2.2.1. Neural Networks

Conventionally, it is the programmer that, while programming, tells the machine what to do and how to do it. However, ML models do not receive that type of commands and are not informed on how to solve the problems. What really happens on these models is that they learn how to solve the problem through observing data. In this study, neural networks are used not only based on its learning capabilities but also on its good suitability to the problem, in particular its good performance and the fact that it allows a knowledge transfer. In this way, a contextualization regarding the neural networks is presented, what they are and how they function.

Fundamentals

Neural networks models are inspired by the human brain and how the biological process of the neuron works. Thus, neural networks also have artificial interconnected neurons. These neural networks are constituted by three layers – input layer, hidden layer, and output layer.

For a better comprehension of the analogy between Artificial Neural Networks (ANN) and biological neural networks is important to know how the human neuron works. According to Basheer and Hajmeer (2000), the human neuron has three principal functional parts – dendrites, cellular body, and axon. The dendrites get signals from the other neurons and transmit them to the cellular body. After that, the cellular body passes them to the axon that is responsible to transport them through synapse to the dendrites of the next neuron. These signals help or inhibit the firing of the neuron. That is, depending on the receptive neuron characteristics, such as the amount of signal that goes through it, synaptic strength and its threshold, the neuron will or not fire, since this only happens if the received signal exceeds the neuron threshold. In the case of artificial neuron, the process is very identical. The connections between nodes represent dendrites and axons, the weights that express the inputs importance represent synapse, and the threshold has the same job.

A perceptron is a unique component of the neural networks however nowadays is more commonly used sigmoid neurons to build ANN's. Thus, it is crucial to comprehend how these neurons operate. According to Figure 12, a perceptron can have multiple binary inputs (x_i) that will later result in a unique binary output. The weights (w_i), real numbers that express the importance for the output of the respective inputs, are what will determine the output (Michael A. Nielsen, 2015).

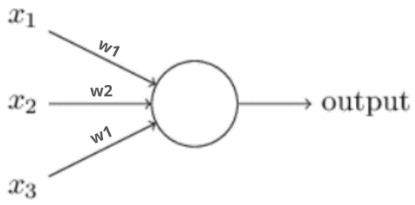

Figure 12 – Perceptron.

Source: Adapted from Michael A. Nielsen (2015)

The output formula for a perceptron (1) showed below says that if the obtained value surpasses the defined threshold the output equals 1, otherwise is 0.

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases} \quad (1)$$

For the case where a single input is considered, the expression above can be written in the following way (2):

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases} \quad (2)$$

In a synoptic way a perceptron can be defined as being a dispositive that takes decisions through inputs ponderation. A variation on the weights or on the threshold will generate different

decision-making models. On Figure 13 it can be seen that an ANN is constituted by three layers. The first layer makes decisions considering inputs ponderation, the second layer makes decisions through first layer's results. That is, in the second layer more complex decisions are made, which means a network with more hidden layers can take more complex decisions (Michael A. Nielsen, 2015).

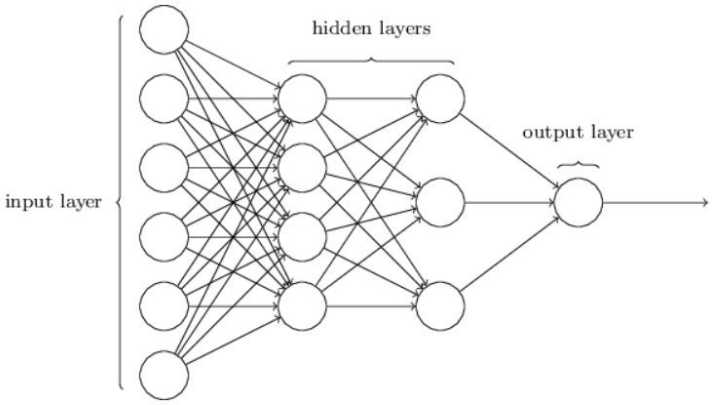

Figure 13 - Neural network with multiple hidden layers.

Source: Michael A. Nielsen (2015)

It is still difficult to understand how a perceptron learns, since a little modification on its weights or bias will most likely cause the network to start acting out of control. Because of this adversity, it was created the sigmoid neurons. These neurons are very similar to perceptron, but the big difference is that, when their weights and bias are modified, it generates a little modification in the output. Thus, a sigmoid neural network is capable of learning and have more than one hidden layer.

Due to not having an activation function, nowadays perceptrons are not used. Like perceptrons, sigmoid neurons also have inputs and their respective weights and bias. Figure 14 is a representation of sigmoid neuron.

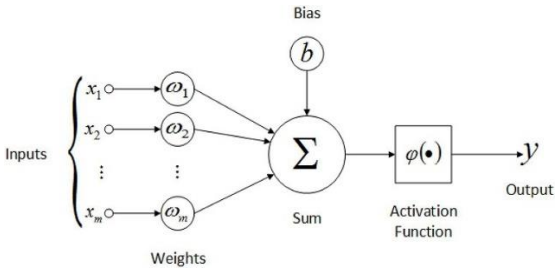

Figure 14 - Activation Function in ANN.

Source: Adapted from Haykin (2008)

Another difference is that sigmoid neurons inputs can take values between 0 and 1 and the output is calculated by the expression (3):

$$\sigma(z) = \frac{1}{1-e^{-z}} \qquad (3)$$

If z is high and positive $e^{-z} \cong 0$ and $\sigma(z) \cong 1$, if z is very negative $e^{-z} \to \infty$ and $\sigma(z) \cong 0$. Therefore, the divergence between both neuron is noted only when z has a modest value. Thus, little changes on the weights and bias generate little changes on the output (Michael A. Nielsen, 2015).

Training

To train the data in ML, it is first necessary to divide the data into training and test sets. Dividing the data into training and test sets is essential for accurately evaluating the model's performance, ensuring that the model can effectively generalize to unseen data and avoid overfitting (Muraina, 2022). ANN's are susceptible to errors, therefore it's essential to train the model for minimizing these errors. Such errors are also called losses. The Loss Function quantifies the difference between the predicted value and the real value of the model. The reason why this function will be necessary for training the model is because neural networks train usually consists of the minimization of the Mean Squared Error (MSE), metric used in regression (Wang et al., 2022).

In order to optimize the losses, it's necessary to find the weights and bias that reach the smaller loss. For that, the gradient descendent algorithm is a possible solution, as it updates the weights while the loss is minimized, and the generated value gets closer to the desired one. This process, commonly denominated as backpropagation, is applied in loop until the cost function is minimized. However, applying the gradient descendent to the whole dataset makes the process too slow. To fight this disadvantage a batch approach can be used. This approach will apply the gradient descent only to a batch of examples from the dataset. The purpose of training the model is making it fit to the data, but not to the point where the model cannot generalize to other datasets. The objective is to avoid overfitting and underfitting (MIT, 2023; H. Zhang et al., 2019).

Authors as Jabbar et al. (2015) and Bashir et al. (2020) say that overfitting occurs when there is high variance and low bias, i.e., the algorithm starts to memorize the noise and characteristics of the training data, fitting the train data so well that the results cannot be trusted. On the contrary, underfitting results of the incapacity of the model to capture the variability of the data due to high bias and low variance. More complex models tend to overfit while simpler models have more probability to underfit (see Figure 15).

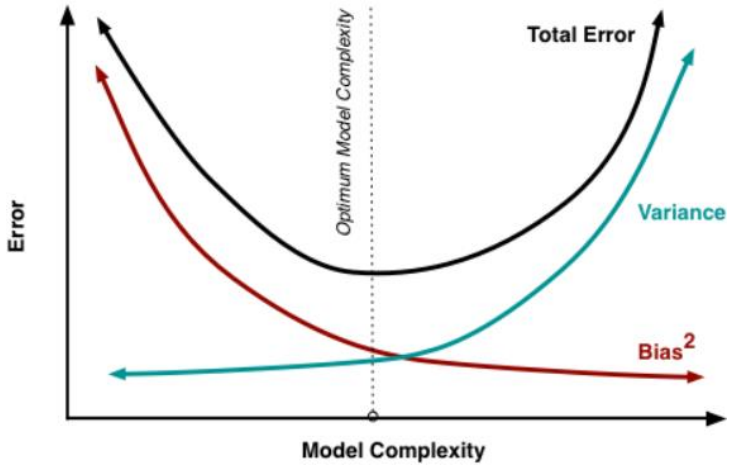

Figure 15 - Variation of model complexity with the bias and the variance.

Source: Fortmann-Roe (2012)

One of the techniques used to guarantee a well-balanced network is regularization. There are two techniques to achieve regularization, namely dropout and early stopping. During train, dropout chooses randomly some subsets randomly and tries to prune them using some random probabilities and after that these neurons fire or are shut down in different iterations of the train. Early stopping consists of stopping the train before overfitting. As can be seen in Figure 16, initially the error from the training set and the validation set decrease, however the train should stop before the validation error starts to increase while the training error decreases (Jabbar & Khan, 2015).

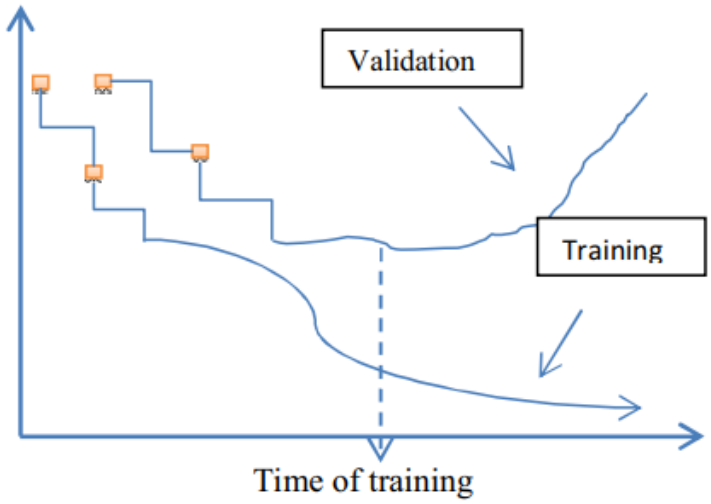

Figure 16 - Time of training to avoid overfitting.

Source: Jabbar & Khan (2015)

Another very often used technique is cross-validation. Cross-validation is a statistic method that estimates the model's performance on unseen data. It consists in dividing the dataset into multiple subsets and using one of them as a test set and the other ones as training sets. The process repeats itself until every subset has been used as the test set. After all validations, in a way to create a more robust estimate of the model's performance, the results are averaged. Cross-validation main

purpose is to prevent overfitting since it provides the generalization error. There are numerous types of cross-validation, including Holdout Method, *K-fold* Cross-Validation (KCV), Stratified *K-fold* Cross-Validation and Leave-P-Out Cross-Validation (Cawley & Talbot, 2010).

In the KCV method, the data is divided into *k* equally sized parts, or "folds". In each round of the process, one of these folds is set aside as a validation set, and the model is trained on the remaining *k*-1 folds. Then the mean squared error (MSE) is calculated on the held-out fold. This process is repeated *k* times, each time using a different fold as the validation set. The final estimate of the test error is the average of the different MSE values, one from each round. Typically, the choice of *k* is 5 or 10, since it has been shown that these values return test error rate estimates with low bias and low variance. The more generalizable results and the easy interpretation make this method one of the most used to evaluate the model design (James et al., 2013; Kuhn & Johnson, 2013).

Hyper-parameters

Hyper-parameters can be defined as variables, determined before training, that impose the model structure and control its behavior. Some examples of the most common hyper-parameters are the number of hidden layers, the number of neurons, the batch size, the activation function, the dropout rate, the learning rate, and the number of epochs (Goodfellow, 2016). These hyper-parameters can be optimized through the application of some algorithms. As stated by Bergstra (2012), Grid Search and Random Search are the most used algorithms for the optimization.

With the purpose of understanding how each referred hyper-parameter influences the time complexity and the efficiency of the results, next is presented a more focused explanation regarding this matter.

The task of selecting the number of hidden layers is complex, since a wrong choice can generate overfitting or underfitting. This hyper-parameter also severely affects the computation time and the global results efficiency of the model negatively. When the number of hidden layers in the neural network surpasses the complexity of the problem it can lead to overtraining, which will lead the model to lose its ability of generalization and increase the training time. Contrary to this is underfitting. Underfitting occurs when the number of hidden layers is less than the complexity of the problem. Consequently, this will generate undertraining and very bad predictive performance. In this way, computation time become very low (Uzair & Jamil, 2020).

Given that the number of neurons is in part related to the number of hidden layers, if the network has a large number of hidden layers, it will also have a large number of neurons. The same

happens if the number of neurons is low. Thus, as Yotov et al. (2020) explicated, a network with a high number of neurons will have a high computation time and inefficient results. Therefore, a network with a low number of neurons will have a low computation time, but low predictive performance. The study made by Ramos et al. (2023) evidenced that when a big number of neurons were selected for the first hidden layer the trained model had bad results since the model caught noise.

Another hyper-parameter analyzed is the batch size (number of training samples). Qasem et al. (2020) tested two deep learning algorithms to address how modifying different hyper-parameters can affect the performance of models. After analyzing the batch size, it became clear that increasing batch size until its optimal size increases the accuracy too, but when that optimal point is surpassed, the accuracy starts to fall. Even if large batch sizes can lead to faster computation times, the risk of overfitting is too high.

Another hyper-parameter is the number of epochs, which is the number of passes of the algorithm during the training phase. Complex data requires more sophisticated models and consequently these models need more epochs compared to less sophisticated models. However, as shown in Figure 17, if the number of epochs is too high the model risks overfitting, or underfitting, if the number of epochs is insufficient (Qasem et al., 2020; H. Zhang et al., 2019).

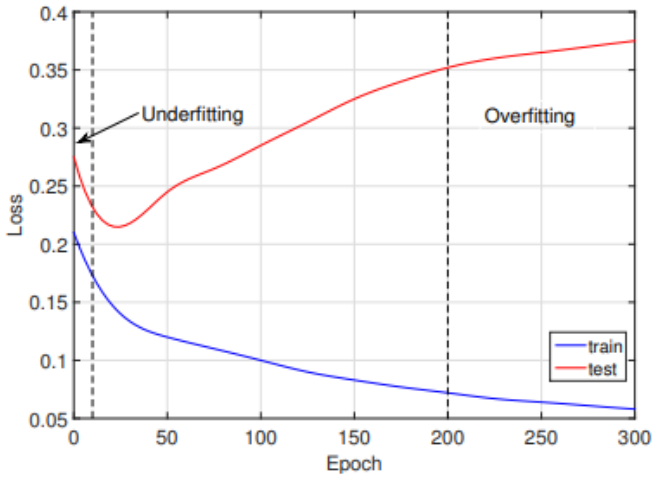

Figure 17 - Variation of over and under fitting with epochs.

Source: Zhang et al. (2019)

A very important hyper-parameter is the activation function. These functions are necessary to prevent linearity and to secure that the model is capable to learn complex representations. The most used activation functions are the following ones: sigmoid, hyperbolic tangent, rectified linear unit (ReLu), parametric leaky version of rectified linear unit (PReLU), exponential linear unit (ELU), scaled exponential linear unit, swish and mish (Rasamoelina et al., 2020).

According to Ding et al. (2018) , the utilization of hyperbolic tangent activation functions leads to a faster convergence time than those with sigmoid activation functions. The ReLU activation functions have a faster convergence too, but because their average output is positive, it causes a bias shift in the next layer. Considering that ReLU function is left-hard-saturating, meaning that the relative weights can stop being updated, causing neuron death, these can impact very negatively the convergence of the model. However, PReLU converges faster and has lower training error. Applicating ELU has shown that these activation functions learn faster and have a better performance than sigmoid, ReLU and PReLU.

With the purpose of reducing overfitting, the dropout technique can be applied. Pauls et al. (2018) emphasize that an optimal range for dropout rate is from 0 to 1 and a small percentage of dropout is more helpful that no rate at all even on small datasets. The use of dropout will increase the computing time, since this technique requires a larger number of interactions to converge.

An equally relevant hyper-parameter is the learning rate. Learning rate is responsible for controlling the adjustments made to the weights of the neural network, always respecting the loss gradient. The challenge when choosing the learning rate is that a very small value can contribute to long training times that eventually can get stuck on local optimal. On the other hand, a very high value can lead to a sub-optimal set of weights too fast, impacting very badly the performance of the model (Raitoharju, 2022).

A tunned set of hyper-parameters is essential to achieve a good performance, since regulated models that learn better do not usually get stuck on local optimal. To achieve this, it can be used a variety of different tools, for example Optuna, Hyperopt, Spearmint, Sequential model-based algorithm configuration, Autotune and Vizier (Akiba et al., 2019).

Evaluation Metrics

There are several evaluation metrics for neural networks regression problems: Coefficient of Determination, Adjusted Coefficient of Determination, Mean Squared Error, Root Mean Square Error, Mean Absolute Error, and Mean Absolute Percentage Error are the most used evaluation metrics. Table 3 presents for each metric their description, expression, advantages, and disadvantages (Plevris et al., 2022).

Table 3 - Evaluation Metrics

Metric	Description	Expression	Advantages	Disadvantages
Coefficient of Determination (R^2)	Quantifies the variance explained by the model, where the output value is between 0 and 1. The closer to 1 the output value is the better is the model performance.	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$	Identifies linear relations between the model and the data.	Is biased; can be applied perfectly only on univariate models; R^2 alone is not capable to determine if a model is efficient or not.
Adjusted Coefficient of Determination (R^2 Adjusted)	Follows the same principle as R^2, however was modified to become unbiased.	$1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$	Can be used to evaluate models with more precision; can be used on multivariate models; is unbiased.	Is indicated for simpler models.
Mean Square Error (MSE)	Measures the mean of the squares of the differences between the values predicted by the model and the actual values.	$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$	Is a good metric to evaluate models that don't tolerate big errors.	Difficult to interpret and is sensitive to outliers.
Root Mean Square Error (RMSE)	Follows the same principle as the MSE.	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$	The output is on the same unit that the destiny variable, so it's easier to interpret.	-
Mean Absolute Error (MAE)	Consists of the mean of the absolute differences between the predicted	$\frac{1}{n}\sum_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert$	Ideal for models that measure a lot of data or seasonal	Is not recommended for delicate models

	values and the real ones.		data; is easy to interpret.	since it does not square the differences.
Mean Absolute Percentage Error (MAPE)	This metric has as an output a percentage, the smaller the value the best is the model performance.	$\frac{1}{n}\sum_{i=1}^{n}\left\|\frac{y_i - \hat{y}_i}{y_i}\right\|$	Very easy to interpret.	Does not deal well with values close to 0.

Data Pre-processing

Non-treated data can contain noise, missing values, and inconsistencies. As such, data pre-processing is a crucial step to accomplish an efficient performance of the model. This preparation and transformation of the initial dataset includes methods respecting four different approaches, namely data cleaning, data integration, data transformation and data reduction (Sivakumar & Gunasundari, 2017).

Many authors as Bhaya (2017), Huang et al. (2015) and Garcia et al. (2016) refer that the recommended data pre-processing englobes a vast number of techniques. The main points can be divided in the following way:

→ Get an overview of the dataset for a better understanding of the dataset structure and identify irregularities through mean, median, variance, standard deviation, and quantiles.

→ Identify missing values, outliers, and inconsistencies.

→ Make sure that the data type can be processed by machines.

For handling missing values, different methods can be used, such as dropping those missing values, replace them with zero or with mean, median, mode and it can also be used interpolation or extrapolation. Noisy data also needs to be treated or eliminated. Thus binning, regression and clustering are some of the techniques that can be useful to do so. In the case of outliers, they should be removed from the data set or transformed, techniques as imputation, binning or discretization can be used. In case there are imbalanced datasets, to avoid biased models, ensemble methods and resampling are solutions, however this last method can lead to overfitting or underfitting.

Real world problems deal with data that can present in very different rages, so to be possible to compare all variables on common grounds it must be applied standardization, normalization

techniques (min-max, z-score). Another problem that can appear is that not always data is in a format that can be processed by machines. To solve this problem is needed to encode categories. It can be used classical encoders as label encoding for ordinal values, one-hot-encoding when variables don't have an inherent order or Bayesian encoders.

2.3. Transfer Learning

Machine learning and data mining are both fields that have been used in many applications, where past information can give essential patterns to successfully predict future results. Traditionally, in machine learning, train and test data feature space and distribution are the same. However, when this does not happen, the performance of the model becomes compromised. For that reason, the need to have a tool capable of learning for a certain domain from a related domain became a high priority. This need is the motivation behind transfer learning (Pan & Yang, 2010).

In transfer learning the target domain is the population of interest that can´t be trained generally because of the lack of labels. However, when data from a population similar to the target one is available it can be used as a source (Kouw & Loog, 2019).

So, when the target training data is rare, expensive, or inaccessible the need for transfer learning emerges. This approach is very attractive mainly because of the existing data repositories that allow using existing datasets. Transfer learning intends to extract from one or more sources the knowledge later applied to the target (Weiss et al., 2016).

To have a better understanding of the theme, firstly it is necessary to define what is a domain and a task. Pan and Yang (2010) introduced these concepts in the following way:

→ **Domain $\mathcal{D}$**: composed by a feature space X and a marginal probability distribution $P(X)$, where $X \in \mathcal{X}$;

→ **Task $\mathcal{T}$**: composed by a label space $\mathcal{Y}$ and an objective function $f(\cdot)$ (capable of making predictions on unseen data).

In simple terms, Pan and Yang (2010), only consider a source domain denoted as $\mathcal{D}_S = \{(x_{s_1}, y_{s_1}), \dots, (x_{n_s}, y_{n_s})\}$, where the data instance $x_{n_s} \in \mathcal{X}_S$ and the corresponding class label $y_{n_s} \in y_S$, and a target domain $\mathcal{D}_T = \{(x_{T_1}, y_{T_1}), \dots, (x_{n_T}, y_{n_T})\}$, where $\mathcal{X}_{T_i} \in \mathcal{X}_T$ is the input, and the corresponding output $y_{T_i} \in y_T$.

The transfer learning definition, illustrated on Figure 18, given by Pan and Yang (2010) is the following:

"Given a source domain $\mathcal{D}_S$ and learning task T_S, a target domain $\mathcal{D}_T$ and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ for $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and T_S, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $T_S \neq T_T$." (p.1347)

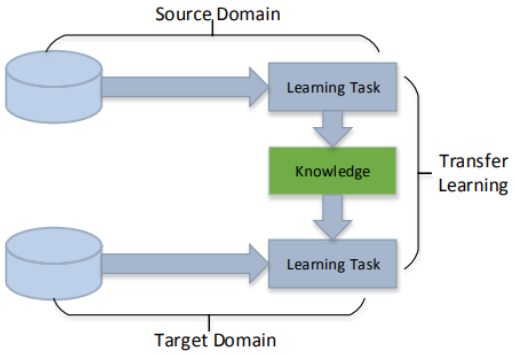

Figure 18 - Transfer Learning process.

Source: Tan et al. (2018)

While designing a transfer learning algorithm there are three questions that should be answered: when to transfer, what to transfer and how to transfer (Pan & Yang, 2010).

Blitzer et al. (2006) and Jialin Pan et al. (2008) categorized transfer learning in two ways, homogeneous transfer learning and heterogeneous transfer learning. When it is assumed that both the source and target domains data are in the same feature space it can be assumed that it is a case of homogenous transfer learning. On the other hand, heterogeneous transfer learning assumes the source and target domain are from different features spaces.

Transfer learning can also be categorized regarding the availability of labelled or unlabeled data on the target domain. In case no labelled data is available in the target domain and a lot of data is available on the source domain we have a case of transductive (supervised) transfer learning. Unsupervised transfer learning has only available unlabeled data. For the cases where labelled and unlabeled data are available for training, we have a case of inductive (semi-supervised) transfer learning (Pan et al., 2014; Pan & Yang, 2010).

Tan et al. (2018) presents another way of categorizing Deep Transfer Learning (DTL) based on several aspects of the applied approaches:

→ **Instance-based**: uses some parts of instances selected from the source data and applies different weighting strategies on the target data.

→ **Feature-based/mapping-based**: transforms instances from the source and the target data into more homogeneous data. This approach can also be divided in symmetric and asymmetric subcategories. Asymmetric approaches transform the source features to match

the target ones. Symmetric approaches aim to find a common latent feature to transform the source and target features into a new feature representation.

→ **Parameter-based/network-based**: uses the knowledge obtained through the model and applies different combinations of pre-trained layers.

→ **Relational-based/adversarial-based**: extracts transferable features from the source and the target data.

All these approaches can be seen below on Figure 19.

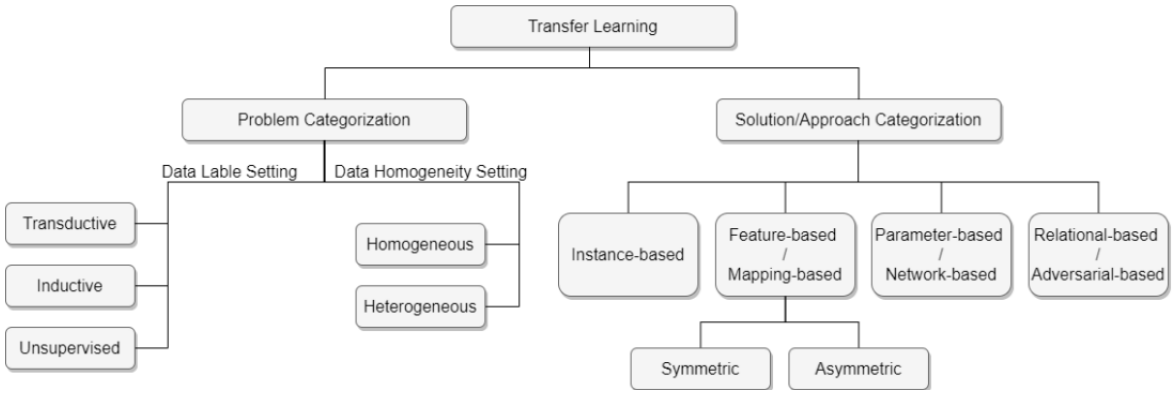

Figure 19 - Transfer learning approaches.

Source: Iman et al. (2023)

Parameter-based/model-based approaches can adapt the domain between the source and target data by adjusting the model. Thus, the approaches seen on Figure 20 are the most used ones on DTL. Pre-training, freezing, finetuning and adding fresh layers are some of the different techniques that can be applied to model-based approaches. Pre-trained technique is when a DL model is trained on source data. Contrary to this is freezing and finetuning techniques that use some or all layers of pre-trained models to train the model on the target data. Freezing layers consist of maintaining the values of the parameters/weights frozen. If the parameters/weights are initialized using the pre-trained values for some or all layers, it's a case of finetuning. Adding new layers to the model after freezing the pre-trained model to train it on the target data is a technique known as Progressive Learning/progressive neural networks (PNN's). To prevent loss in DTL, these techniques freeze the pre-trained model and train the layers newly added on top of the previously layers that were trained before. In DTL the best strategy is to freeze the first and middle layers from the previous pre-trained model and finetune the other layers for the new task that will be later added to the pre-trained model, since the first layers are the ones that do feature extraction at a high level of detail while the other layers do the predictions (Rusu et al., 2016).

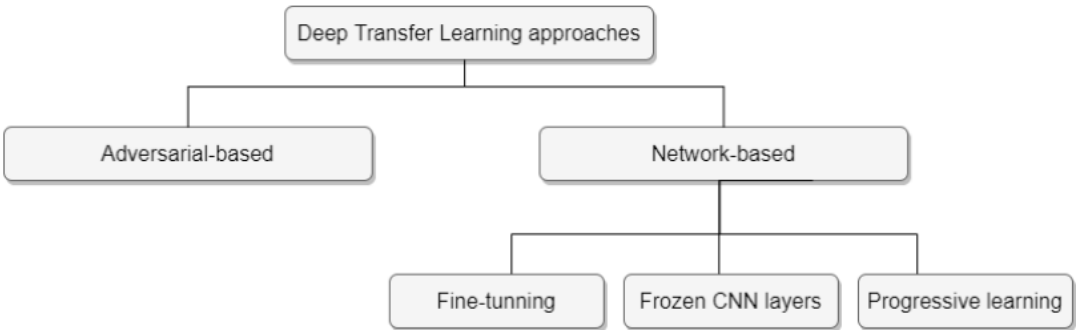

Figure 20 - Most used Transfer learning approaches.

Source: Iman et al. (2023)

Farahani et al. (2021) says that domain adaptation or domain shift is a special case of transfer learning where the feature and labels spaces are maintained and only the features are modified. So, domain adaptation was created to fight the problem where the source and target domains are related but from different distributions. However, it is very difficult and almost impossible to find a source and target domains with the same label space. If the label space is different the target domain assumes that the features classes represented by the source domain, and that do not exist in their space, are outliers, causing negative transfer which negatively influences the model performance. Therefore, not only distribution disparity - domain gap - but also differences on labels spaces - category gap - needs to be considered.

It is possible to divide domain adaptation into four categories considering the category gap: Closed Set, Open Set, Partial and Universal Domain Adaptation (UDA).

- → **Closed Set**: when both domains (source and target) have the same classes and still exists a gap between domains.
- → **Open Set**: when some of the labels are shared in the common label set and/or have private labels (Busto et al., 2019). Later it was introduced by Saito et al. (2018) a modified open set that considered the source label set as a subset of the target label set. This open set adaptation is useful in cases where exist multiple source domains that include a subset of target classes, making the performance of the model increase because of the utilization of the information contained on the shared source domain.
- → **Partial**: when the target label set is a subset of the source label set, contrary to open set, the source domain has numerous classes (Cao et al., 2018; J. Zhang et al., 2018).
- → **Universal Domain Adaptation**: in contrast to the above topics that need prior knowledge about both the source and target label sets, this adaptation does not require any prior knowledge. Firstly, UDA tries to find across the domains the label space that is shared, secondly and like open set and partial adaptation tries to align the data in the shared label set. Then with the objective to apply it safely to the unlabeled target data a classifier is

trained on the shared source labelled data. This trained classifier is going to accurately assign labels to the target samples on the shared label space and mark the outliers' classes as unknown, this process occurs on the testing phase (You et al., 2019).

In addition to this Farahani et al. (2021) categorized domain shift in three classes, namely: Prior Shift, Covariate Shift and Concept Shift.

→ **Prior Shift**: also known as class imbalance, refers to the cases where posterior distributions are equivalent and prior distributions are not equal between domains. To solve problems in this situation it is needed labelled data in both domains.

→ **Covariate Shift**: considers situations where marginal probability distributions are different between each other, and conditional probability distributions are equal across domains. This type of shift can be caused by sample selection bias and missing data. The domain adaptation techniques proposed on the paragraph above aim to solve this gap.

→ **Concept Shift**: concept shift or data drift do not change the data distributions and conditional distributions are different across domains. Like the first shift, concept shift also needs labelled data in both source and target domains.

In Figure 21 it is illustrated a very important question made when using transfer learning techniques about transferability: when will this technique work and to what extent? The feasibility and transferability of transfer learning are imperative topics (Bao et al., 2022).

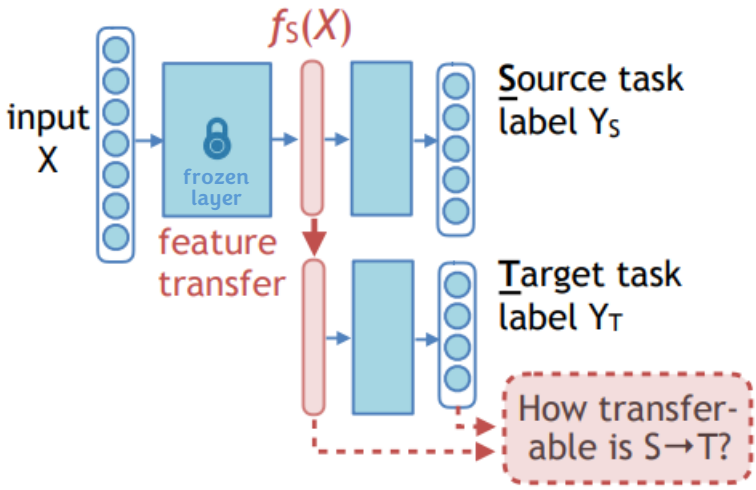

Figure 21 - Transfer learning model.
Source: Adapted from Bao et al. (2022)

The traditional way to measure transferability is through model loss or accuracy on the validation set, and there are a few studies that also mention task relatedness (Bao et al., 2022; Y. Tan et al., 2021; Yosinski et al., 2014).

Yosinski et al. (2014) say that transferability may be negatively affected by two factors: (i) the higher layer neurons specialize on their initial task, damaging the performance on the target task and (ii) splitting networks between co-adapted neurons raise optimization issues. However, the authors refer that these issues occur depending on the local of transfer, i.e., if the features transfer is from the bottom, middle or top of the network. Features transferability can also be harmed when the source task and target task are very distant.

Considering the study's purpose of transferring the knowledge learned from the transport costs in other type of goods predictive model to the transport costs in MC predictive model, Table 4 lists the details regarding the domain as well as the learning task for both the source and the target.

Table 4 - Details about the Source and the Target

Source	**Domain** (feature space)	Characteristics of Goods
	Learning Task (label space)	Predicted Costs on Other Type of Goods
Target	**Domain** (feature space)	Characteristics of Goods
	Learning Task (label space)	Predicted Costs on MC

In terms of homogeneity and heterogeneity, the case addressed in this dissertation corresponds to a homogeneous case, since the data from the source domain and the target domain come from similar feature spaces and may share some common features such as distance, weight, type of costs, transportation mode, etc. Regarding the problem, the same can be characterized as Supervised (Transductive), given that the target domain lacks labelled data in contrast to what happens in the source data. Thus, the most suitable approach to solve the problem is the Parameter-based. This method reutilizes part of the network that pre-trained in the source domain and transfer it to the network used in the target domain. The source task and the target task are not very far apart, so the transferability of this problem is expected to be good.

2.4. Deep Learning Libraries

Deep learning has a vast number of powerful libraries perfect to perform on large datasets, as per example: TensorFlow, Keras, Theano, MXNet, Lasagne, Caffe and PyTorch. One of the benefits of using Python is the facility to work with a wide range of libraries and thus offer the possibility to

code and work with deep learning, additionally it is a very easy to interpret and learn language. To build the proposed predictive model it will be used the PyTorch library. This library has a very simple interface and effortlessly integrates with Python core functions, allowing through Compute Unified Device Architecture (CUDA) the code to run on the graphical processor helping increase the performance of the ANN. In addition, PyTorch provides dynamic computational diagrams granting the user to manipulate them at runtime as well as simple and fast coding (Erickson et al., 2017; Imambi et al., 2021).

The three basic components of PyTorch to have into account are the following ones (Imambi et al., 2021):

→ **Tensor**: N-dimensional array that runs on GPU.

→ **Variable**: Node which stores data and gradient.

→ **Module**: Layer which stores state or learnable weights.

For data manipulation and analysis will be used the popular libraries Pandas, NumPy (also used in data and model parameters) and matplotlib for the creation of static visualizations. These libraries are rich when it comes to data structures and tools to work on many fields.

Scikit-learn library is also very popular for machine learning algorithms implementation and is very easy to use, so it will be used on the model alongside NumPy and PyTorch. Scikit-learn has very key attributes and features has presented below (Pedregosa et al., 2011):

→ **Supervised and unsupervised learning algorithms**:
- K-Nearest Neighbor, Random Forest, etc. for classification.
- Linear regression, Lasso etc. for regression.
- K-means, mean-shift, etc. for Clustering.

→ **Pipelines**: give consistency and prevent data leakage as well as simplify the code.

→ **Tools to help in model selection**: metrics, cross-validation, and grid search.

→ **Tools for pre-processing**: helps in normalization, feature selection, one hot encoding etc.

Another library that will be used is tqdm. This library provides the possibility to add progress bars, facilitating loops visualization, since it can be a very difficult task while running the model (Costa-Luis, 2019).

2.5. Artificial intelligence applications in Modular Construction

With the projections pointing to the growth of modular construction, the construction industry started to search for AI solutions to help in different fields, such as quality check, scheduling, safety,

and design management (Reports and Data, 2019). Despite that, the number of construction companies that adopted AI techniques is extremely low and is behind other type of industries. Mainly because of the lack of skilled professionals but also appropriate tools, AI adoption in this sector still needs a lot of improvement (Blanco et al., 2018).

Table 5 cites some information regarding some articles where AI solutions were applied to MC in diverse scenarios and contexts. The table precedes a more detailed explanation about the selected articles methodology and results.

Table 5 - Articles regarding the application of AI to Modular construction

Title	AI solution	Evaluation metrics	Bibliographical Reference
"Applications of object detection in modular construction based on a comparative evaluation of deep learning algorithms"	Development of two deep learning algorithms, including a faster region-based convolutional neural network and single shot multi-box detector, for object detection in modular construction.	- Average recall and mean average precision by image pixels - Recall and precision by counting	(Liu et al., 2022)
"Towards Intelligent Agents to Assist in Modular Construction: Evaluation of Datasets Generated in Virtual Environments for AI training"	Development of a mechanism to create large datasets in virtual environments and evaluation of the performance of AI models, such as Multi-view Convolutional Neural Network, trained with the generated datasets.	- Accuracy - Precision - Recall - F1-score	(Park et al., 2021)
"Machine-Learning Approach to Predict Total Fabrication Duration of Industrial Pipe Spools"	Development of tree-based models, including Random Forest, Extremely Randomized Tree, and Gradient-boosted Decision Tree, to predict the total fabrication time of industrial pipe spools.	- MAPE - MAE - Pearson correlation coefficient	(Mohsen et al., 2022)

"Optimizing Modularization of Residential Housing Designs for Rapid Postdisaster Mass Production of Housing"	Application of a Multi-objective Genetic based solving Algorithm to optimize the prefabricated module design.	- Schott's spacing metric - Inverted generation distance	(Ghannad & Lee, 2023)
"Activity identification in modular construction using audio signals and machine learning"	Development of an audio-based task identification machine learning framework.	- Accuracy - Precision - Recall - F1-scrore	(Rashid & Louis, 2020)

Aiming to provide information regarding project quality and safety, Liu et al. (2022) built two deep learning algorithms for object detection in modular construction. There were selected four infrastructures and three modular objects for data collection, namely modular panels, safety barricades and site fences. The chosen algorithms were the faster region-based convolutional neural network and the single shot multi-box detector, where the first one showed better accuracy in detecting selected objects and contributed to improvements in monitoring object installation and helping avoid human errors.

With the increasing necessity to help the workers and minimize errors during the assembly step in modular construction, Park et al. (2021) overviews a technique that creates, in virtual environments, large datasets. These datasets are generated already containing the relevant contextual data to train the AI models used to identify the next step in modular assembly. The mechanism developed by the authors generated 84,000 images, resulting in 6,000 images to train the model. The authors evaluated the performance of a Multi-view Convolutional Neural Network and obtained a value of 0.97 for the evaluation metrics that were used. These results indicate that the model results are accurate for multiple scenarios and that the dataset generated by virtual environments is suitable.

In order to speed up the assembly process in modular construction companies use piping structures, prefabricated pipe spools that are arranged and then attached to a steel structure. However, the estimation time for their fabrication is very uncertain due to material availability, capacity factors and work performance and conditions. So, Mohsen et al. (2022) proposed a machine learning model to predict the total fabrication time of industrial pipe spools to help

planning the workforce and material necessities for each construction. Through the combination of product composition and real-time loading status of the shop, the authors developed and analyzed several tree-based machine learning models, such as Random Forest, Extremely Randomized Tree, and Gradient-boosted Decision Tree. For evaluation purposes the authors utilized records of 6,989 pipe spools. When compared to the original heuristic results the prediction accuracy was enhanced by achieving a 35% reduction in the MAPE and a 50% reduction in the MAPE. Adding to this, the author also found that the actual and predicted results had a strong positive correlation of 0.8. Between the developed models, both the Random Forest model and the Extremely Randomized Tree model obtained better results compared to the Gradient-boosted Decision Tree model. However, the Extremely Randomized Tree model outperformed the other two with a MAPE result of 26.6% compared to 28.6% for the Random Forest model and 31.6% for the Gradient-boosted Decision Tree model.

Post disaster housing reconstruction is an overly complex category of modular construction mainly because of the large-scale projects and the lack of resources after a disaster. To help the planners' teams responsible for these mass production projects, Ghannad & Lee (2023) applicated a Nondominated Sorting Genetic Algorithm, which is a Multi-objective Genetic based technique, to optimize the module design considering factors like manufacturing, transportation, and assembly processes. The selected metrics to evaluate the model's performance were the Schott's spacing metric and the inverted generation distance, where a model with a low value of the second one indicates more suitability. The results of 0.0045 for Schott's spacing metric and 0.424 for inverted generation distance metric confirmed that the proposed approach significantly improved optimization and decision-making of MC design and construction, enhancing the rapid responses to the demands of the post disaster process.

Being the study and evaluation of productivity a key factor for companies, Rashid & Louis (2020) developed and applied an audio signal and machine learning tool to identify, track and monitor different activities in a modular construction factory. First it was recorded the audio and video from a modular construction factory. After that all the data was treated and prepared to be trained. A multi-class support vector machine was the model implemented by the authors. After the training the model obtained a 96.6% F1-score, demonstrating the potential of the proposed method.

The literature background covering AI in MC confirms the lack of studies on the forecast of costs on MC. Several authors provided a lot of information regarding cost optimization, module design optimization and module fabrication time estimation, however there is still no tool capable of

predicting costs on MC, which enhances the importance of the development of a method capable of doing it.

blank page

3. DATA AND METHODS

In the following subchapters it will be outlined the research methodology and its practical application. First are covered the data understanding and data preparation steps, where are described the relevant information about the provided data for both Road and Sea inbound, following the description of the techniques employed in preparing the data to be trained by the model. The chapter ends with the modeling and training phase that detail the methods and processes used to train the data, develop, and fine-tune the model.

3.1. Data understanding and preparation for inbound

All data corresponding to the goods transportation from the suppliers to the construction company, also named inbound, was provided by the logistics operator supporting the case-study in the form of two csv files:

- → Road: contains all the information regarding the transportation of other types of goods made by road, with a total of 225,469 lines.
- → Sea: contains all the information regarding the transportation of other types of goods made by sea, with a total of 23,078 lines.

These files contained all the information collected by the logistics operator regarding transportation of other types of goods. The features (or variables) names and characteristics, from the road and the sea files, are registered in APPENDIX A and APPENDIX B, respectively.

Nonetheless, as expected the first received files were not the final versions, since a lot of changes needed to be made. On a first overview it was visible that the data contained a lot of null, missing, and inaccurate values that needed to be eliminated or treated. The next sections are going to explore and prepare the data to be trained. In data preparation the data from the csv files will be explored and prepared so it can be later used in the model.

3.1.1. Road inbound

After a simple data exploration was visible the big quantity of missing values, so the first step was to fill those values with 0. Since this work focuses on the prediction of transportation costs it does not make sense to have values equal or lower than zero for the revenue variable, so they will be eliminated. The same thinking will be used in the distance and volume variables. In the case of all the costs and sales variables, the negative values will be eliminated and the lines that simultaneously contain all costs and sales values equal to zero will be eliminated too.

Considering the large number of features in the road dataset it is necessary to do a selection so that only the relevant features are kept, helping to reduce the dimensionality of the data and also guaranteeing a good performance of the model. In order to successfully do this selection, it is necessary to understand how the different variables are related with each other, how they affect each other. The correlation heatmap in Figure 22 uses color coding to illustrate the strength and direction of correlations, with red indicating strong positive correlations and blue indicating strong negative correlations.

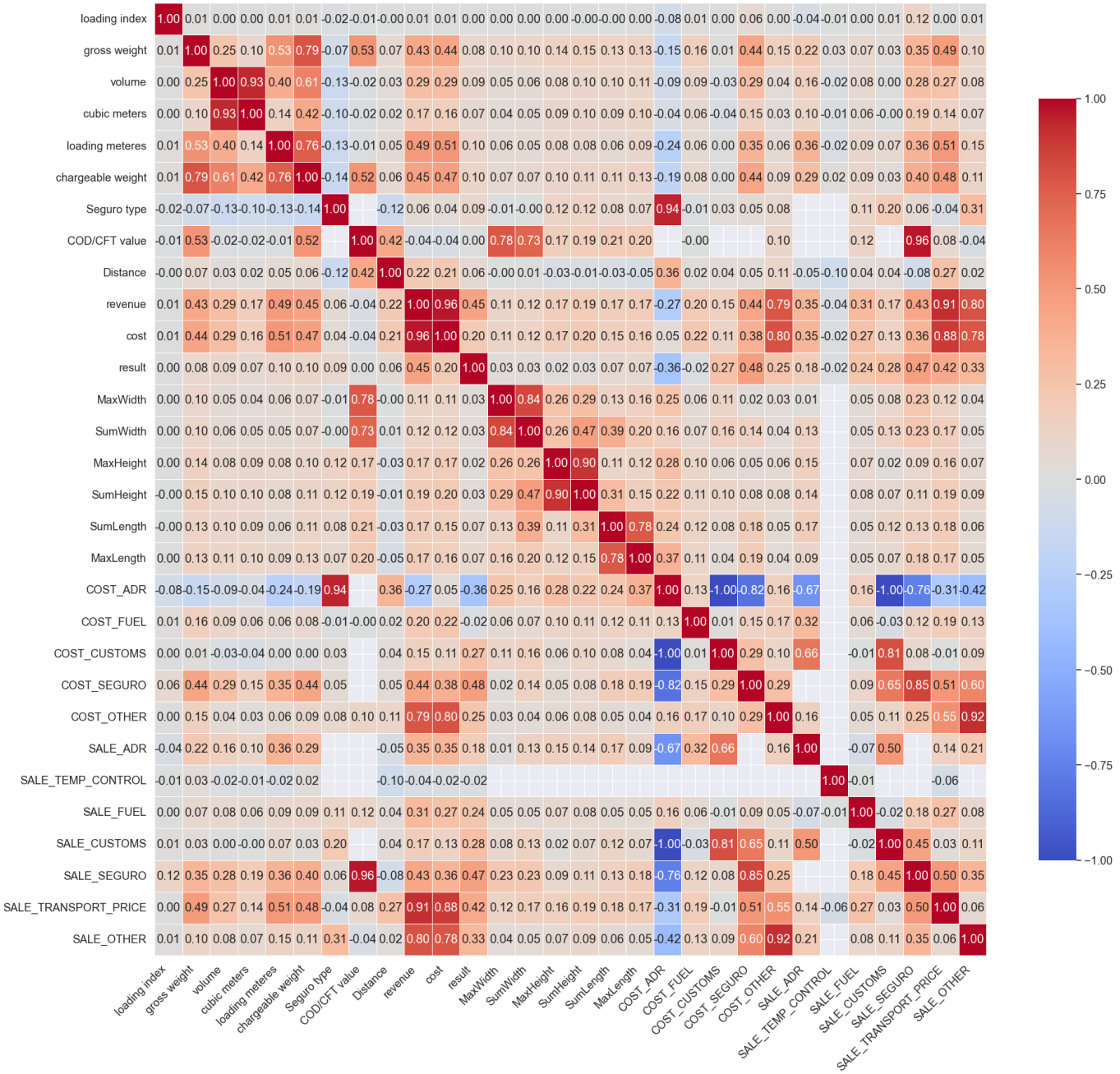

Figure 22 - Road inbound: Correlation Heatmap of Numeric Variables.

Upon reviewing the correlation heatmap, the following significant observations can be noted:

- → Gross Weight and Sale Transport Price: There is a moderate positive correlation (0.49). This indicates that as gross weight increases, sale transport price also tends to increase.
- → Gross Weight and Loading meters: Also have a moderate positive correlation (0.53), suggesting a similar relationship to that observed between gross weight and volume.
- → Gross Weight and Chargeable Weight: It is visible a strong positive correlation (0.79), showing that an increase in gross weight generally results in an increase in chargeable weight.
- → Cost ADR and Seguro Type: There is a very strong positive correlation (0.94), indicating a significant relationship between these variables.
- → Cost ADR and Cost Customs/ Sale Customs: There is a perfect negative correlation (-1) between the Cost ADR variable and the Cost and Sale Customs variables, which means they have a perfect inverse linear relationship.
- → Cost ADR and Cost Seguro: There is a strong negative correlation (-0.82), suggesting that as one variable increases, the other tends to decrease in a very predictable and strong way.

The matrix also indicates that characteristics related to physical dimensions (width, height, length) show expected correlations, with sum values correlating positively with each other and negatively with maximum values, and that financial costs (e.g. Cost ADR, Cost Fuel, Cost Customs, Cost Seguro) show complex relationships, with some strong correlations both positive and negative, indicating different influences and relationships between these costs.

It will be eliminated unstable variables with many categories and that do not have a big impact on the cost, to assure a trustworthy dataset, thus it will be only considered “Quotes” and “Invoiced” cases.

It was also suggested by the logistics operator supporting the case study that data from the last three months in the dataset should be eliminated, as these observations contained cost data that had not yet been approved. This step will also be included in the Sea inbound data preparation.

To make sure that all the variables are in their right format, they will be converted to their respective types, categorical or numeric. However, the variables that have a very high relationship with the target variable need to be eliminated after the cleaning otherwise they will create a bias that will negatively influence the efficiency of the model, giving good results but misleading. Therefore, all the costs and sales variables will be eliminated because they are extremely related to the target variable. Subsequently the variables chosen to be kept are the ones presented in APPENDIX C.

The inbound will be split into training and testing sets, with a test size of 20% and no shuffling. After this, encoding (for categorical variables) and standardization (for numeric variables) are necessary. For that it will be created a preprocessing pipeline with the objective to prepare the raw input data for the machine learning model, ensuring that both categorical and numerical data are transformed consistently and efficiently in a suitable format for analysis. Finally, it is necessary to efficiently prepare the data for training a neural network using PyTorch on a GPU-accelerated platform (CUDA), which can lead to faster training times and improved performance, especially for deep learning models operating on large datasets.

3.1.2. Sea inbound

After a simple data exploration was visible the big quantity of missing values, so the first step was to fill those values with 0. Since this work focuses on the prediction of transportation costs it does not make sense to have values lower than zero for the costs and invoices variables, so they will be eliminated. The lines that simultaneously contain all costs and invoices values equal to zero will be eliminated too. The cases where both variables of “Cwgt” and “Teus” are equal to zero but have values greater than zero for “Houseinvoice” will be eliminated since are considered errors. According to the logistics operator indication the lines that simultaneously have values equal to zero in the “Housecosts”, greater than zero in “Houseinvoice” and “Ship type” “D” will be eliminated as well as the lines with a “Master” value equal to the one present in the lines that have values equal to zero in the “Housecosts”, greater than zero in “Houseinvoices” and “Ship type” “H”.

Similar to what happened in the Road dataset, considering the large number of features in the sea dataset it is necessary to do a selection. To successfully do this selection will also be built a heatmap to analyze all the relationships between variables. Similar to the first, the colors in Figure 23 represent the correlation strengths and directions.

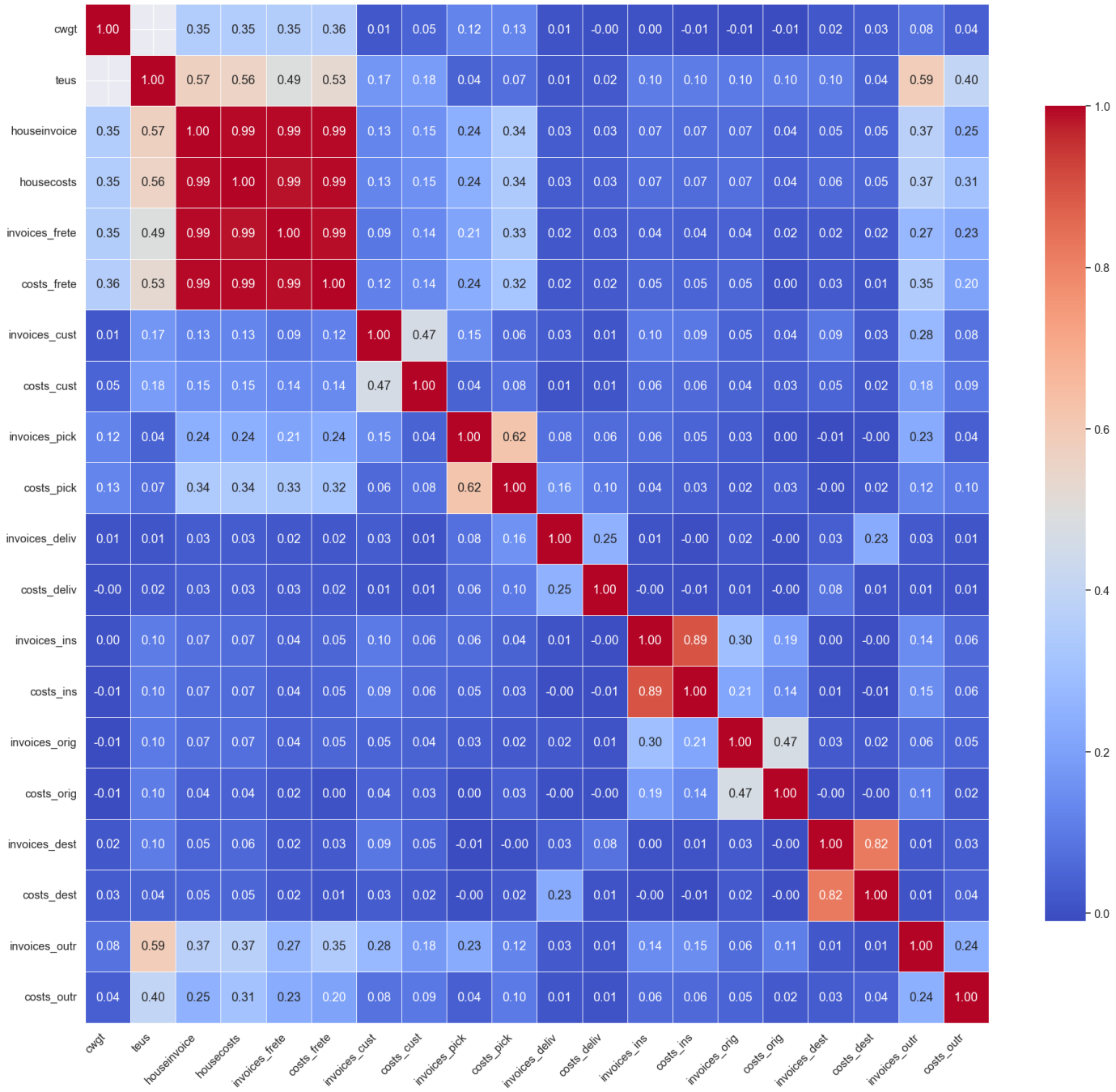

Figure 23 - Sea inbound: Correlation Heatmap of Numeric Variables.

After analyzing the correlation heatmap, we can summarize the key observations as follows:

→ Cwgt (Chargeable Weight): Shows moderate positive correlations with “Houseinvoice” (0.35) and “Costs frete” (0.36), suggesting some relation between the chargeable weight and the invoiced amounts and freight costs.

→ Teus: This feature is moderately correlated with “Houseinvoice” (0.57) and “Housecosts” (0.56), indicating that the value of teus is linked to invoicing and costs.

- → Houseinvoice: Displays very strong positive correlations with “Housecosts” (0.99) and “Invoices frete” (0.99), indicating that invoicing amounts are closely linked to total house costs and freight invoices.
- → Invoices Cust: Positively correlated with “Costs cust” (0.47), suggesting a direct relationship between customer invoicing and dispatch costs.
- → Invoices Pick: Shows a high positive correlation with “Costs pick” (0.62), implying that invoicing for pick-up services directly relates to pick-up costs.
- → Invoices Ins: Strongly correlated with “Costs ins” (0.89), indicating that insurance-related invoices are closely tied to insurance costs.

In the sea inbound logistics, the invoicing amounts are strongly related to their corresponding costs. Effective cost management in these areas can potentially improve the overall financial performance. The correlations suggest that optimizing weight and volume management can help in controlling the associated costs.

To make sure that all the variables are in their right format, they will be converted to their respective types, categorical or numeric. However, the variables that have a very high relationship with the target variable need to be eliminated after the cleaning otherwise they will create a bias that will negatively influence the efficiency of the model, giving good results but misleading. Therefore, all the costs and invoices variables will be eliminated because they are extremely related to the target variable. After the selection step and the pre-processing step are done the variables chosen to be kept are the ones presented in APPENDIX D.

The steps about splitting, encoding, standardization and preparation for PyTorch are identical to the ones referred in the last paragrapher in the subsection 3.1.1.

This dataset, which primarily focuses on sea shipments, also included observations about air shipments. However, after a more thorough analysis, it was concluded that the data on air shipments was not relevant in the context of modular construction, and therefore it was eliminated.

3.2. Inbound Modeling and Training

In this section the hyperparameters will be selected, and therefore it will be selected and validated the most suitable model for the inbound in both train and test set, based on criteria of performance. It will be tested two different scenarios for modelling regarding the loss function / evaluation metric.

Road and Sea inbound Model

The hyperparameters that will be utilized to model and train the two inbound datasets are the next ones:

- → ReLU Activation function.
- → Number and size of hidden layers.
- → Loss Function.
- → Input size.
- → Learning rate.
- → Stochastic Gradient Descent (SGD) optimizer.
- → Linear Learning Rate Scheduler.
- → Number of epochs.
- → Batch size.

The two scenarios mentioned above refer to the use of two evaluation metrics to validate the model, MAPE and RMSE. MAPE was chosen because it is easier to interpret, and in this case, the results are expected to be sufficiently positive and large, thus avoiding the issues MAPE encounters when dealing with values close to zero. In the case of RMSE, this evaluation metric was selected for its straightforward interpretability, as its output is in the same units as the target variable. Additionally, RMSE performs well in scenarios where values are near zero, a situation where MAPE tends to be less effective. In both scenarios, MAPE and RMSE were obtained through cross-validation with 5 folds, to ensure generalization. To ensure a good performance, the hyperparameter optimization will be done through random search for both Road and Sea inbound.

Since the model validation step is iterative, several tries were conducted in order to stabilize the MAPE and RMSE results, which lead to little modifications in the previous pre-processing done in the subchapters 3.1.1 and 3.1.2. The results and discussion about these tries will be presented in chapter 4.

3.3. Outbound

The inbound data pertains to the transportation of goods into the facility, which includes all logistics and supply chain activities required to bring raw materials, components, and products to the

destination for further processing or storage. The outbound data, on the other hand, relates to the transportation of 3D modules from the destination to its clients.

Due to unforeseen delays and dependence on external sources, it was not possible to obtain the data required for modeling and analyzing the transport of modular construction units. Despite this, it is important to outline the intended methodology and approach for the purpose of completeness and future reference.

As outlined in Chapter 2.3, the case addressed in this dissertation corresponds to a homogeneous case. Despite the differences in magnitude between the values in each scenario, the data from both the source domain (inbound transportation) and the target domain (outbound transportation) come from similar feature spaces and consequently share common features.

Given that the problem is homogeneous, parameter-based transfer learning is an appropriate choice. This method allows the adaptation of a pre-trained model from the source domain by refining it on the target domain data. Parameter-based transfer learning, already defined in Chapter 2.3, consists of transferring some of the parameters (weights) of a pre-trained neural network model from the source domain to the target domain. This is effective when both tasks are related, as it enables the model to benefit from the representations that were learned from the source task, while adjusting to the specifics of the target task.

So, the next steps planned involve implementing a parameter-based transfer learning approach to improve the prediction of outbound transportation costs. To date, only the first step was successfully completed:

1. **Initial Model Training on Inbound Data:**

A neural network model using the inbound transportation data was trained. This model has learned to predict transportation costs based on features such as distance, weight, and mode of transport.

Looking ahead, the following steps are planned:

2. **Transfer of Learned Parameters:**

After the first phase is completed and the outbound data is displayed, phase 2 can start. This step would then consist of transferring the learned parameters (weights) from the inbound model to a new model designed to predict outbound transportation costs. To correctly follow the chosen process, the initial layers of the model, which capture the general characteristics relevant to inbound and outbound logistics, will be retained (frozen) to leverage the shared knowledge. The later layers, which are more specific to the outbound task, will be prepared for finetuning.

3. Finetuning with Outbound Data:

Subsequently, the new model will be finetuned using data from outbound transportation, that is, adjusting the task-specific layers of the model to better adapt to the unique characteristics of outbound logistics, while keeping the general layers fixed.

4. Model Evaluation and Validation:

Finally, there will be an evaluation of the performance of the finetuned model using metrics such as MAPE and RMSE. These results will be compared with those from a baseline model that will be exclusively trained on outbound data to determine the effectiveness and improvements achieved by the transfer learning technique.

blank page

4. RESULTS AND DISCUSSION

In the next subchapters it will be detailed all the information regarding the different scenarios for both Road and Sea inbound, as well as the results for every scenario. The results and their respective implications will be analyzed and discussed.

4.1. Road inbound

Table 6 compiles the hyperparameter values and the model evaluation results for each applied filter and modification.

Table 6 – Road inbound: Hyperparameter settings and MAPE results

Loss Function - MAPE			
Filter / modification applied	**Hyperparameter**	**Optimum value**	**Average MAPE across all folds (%)**
Original pre-processing	Number of hidden layers	3	20.70
	Size of hidden layers	100, 100, 50	
	Learning rate	0.1	
	Number of epochs	200	
	Batch size	64	
Addition of seasonality	Number of hidden layers	3	20.24
	Size of hidden layers	100, 100, 50	
	Learning rate	0.1	
	Number of epochs	200	
	Batch size	64	
Filter by “Service level” = “FTL”	Number of hidden layers	3	9.30
	Size of hidden layers	200, 100, 50	
	Learning rate	0.1	
	Number of epochs	200	
	Batch size	32	
Incorporation of shipments that do not come from quotes	Number of hidden layers	3	13.57
	Size of hidden layers	200, 100, 50	
	Learning rate	0.1	

	Number of epochs	200	
	Batch size	32	

First, the model's performance was evaluated using only the standard pre-processing described in subchapter 3.1.1, without any additional modifications. A highly accurate prediction typically needs to have a MAPE value close to 10% (Lewis, 1983), however the model generated a MAPE value of 20.70%. The discrepancy between the training and test MAPE curves, that can be seen in Figure 24, with greater prominence at the end of the curves, may indicate that the model may be overfitting.

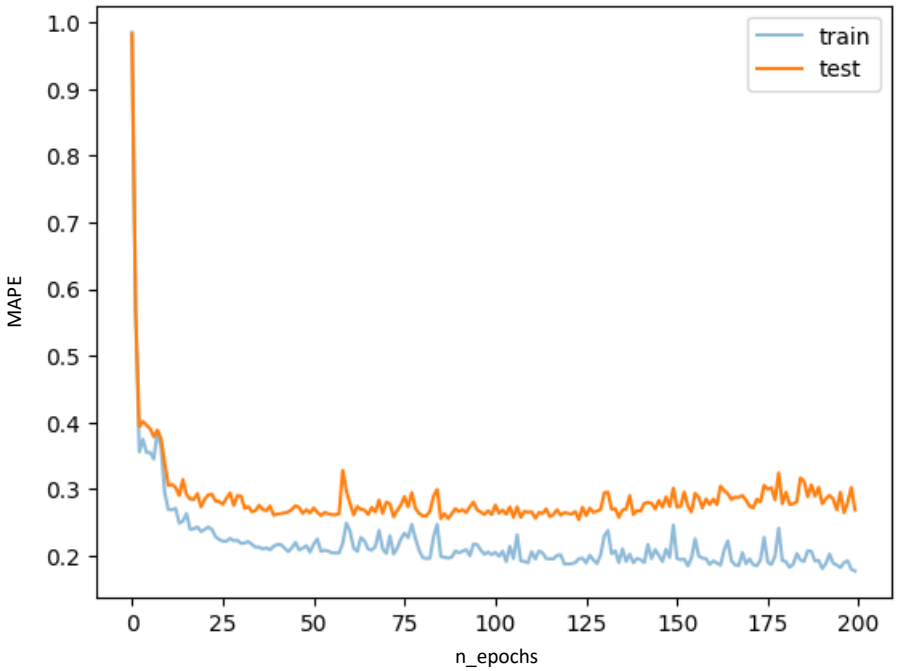

Figure 24 - Road inbound: MAPE curve in the original pre-processing.

In the first attempt to polish the model, additional variables were incorporated to account for the temporal aspect, specifically focusing on the loading date. These variables included the day, month, year, day of the week, and week of the year. This augmentation aimed to capture and integrate the seasonal fluctuations inherent to the transport service domain. However, the output generated by the model did not get significantly better, producing a MAPE value of 20.24%. Similar to what happen in the previous evaluation, the MAPE curves indicate that the model may be overfitting (see Figure 25).

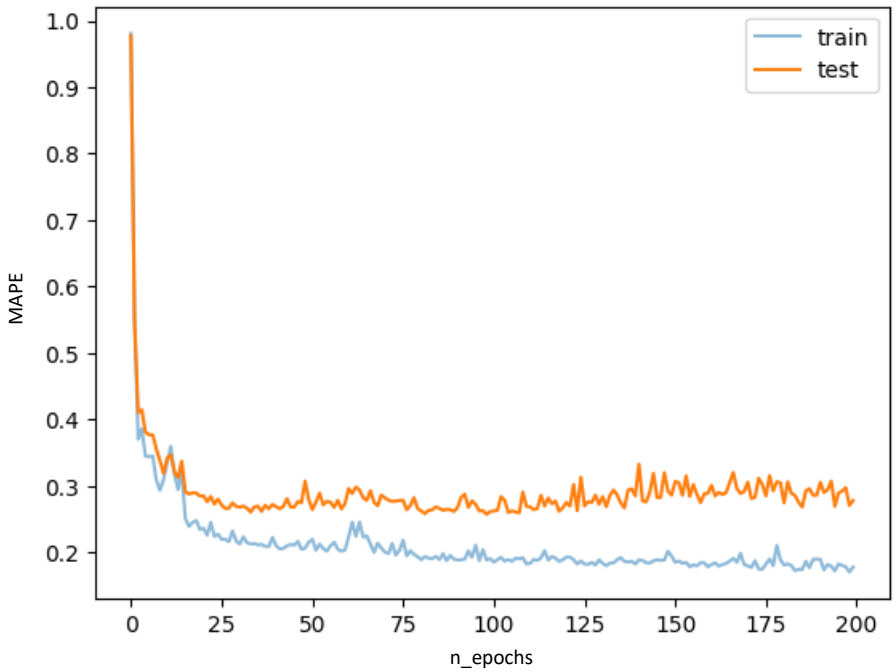

Figure 25 - Road inbound: MAPE curve after adding the seasonality.

Several attempts were made to refine the model and stabilize the MAPE results by applying filters related to various factors such as the "Unloading country," "Loading country", "Incoterms", "Seguro" and "Service level". However, only one of these filters, namely the "Service level" filter, yielded positive outcomes. So, in addition to what was done in the previous test, a filter that kept in the data set only observations relating to a service level of "FTL" (Full Truckload) was also added, as this considerably improved the results. In Figure 26 is represented the plot of loss over time during training of the best performance of the model with a MAPE value of 9.30%. The loss over time is decreasing through the 200 epochs indicating that the model is learning and improving its performance on the training data. It is visible some instability, the training and test performance are diverging a little, indicating some overfitting, and as such further training would not improve the model's performance. The application of the "FTL" filter generated an output that outperformed the previous ones, with a lower MAPE value as well as more stable MAPE curves and less overfitting.

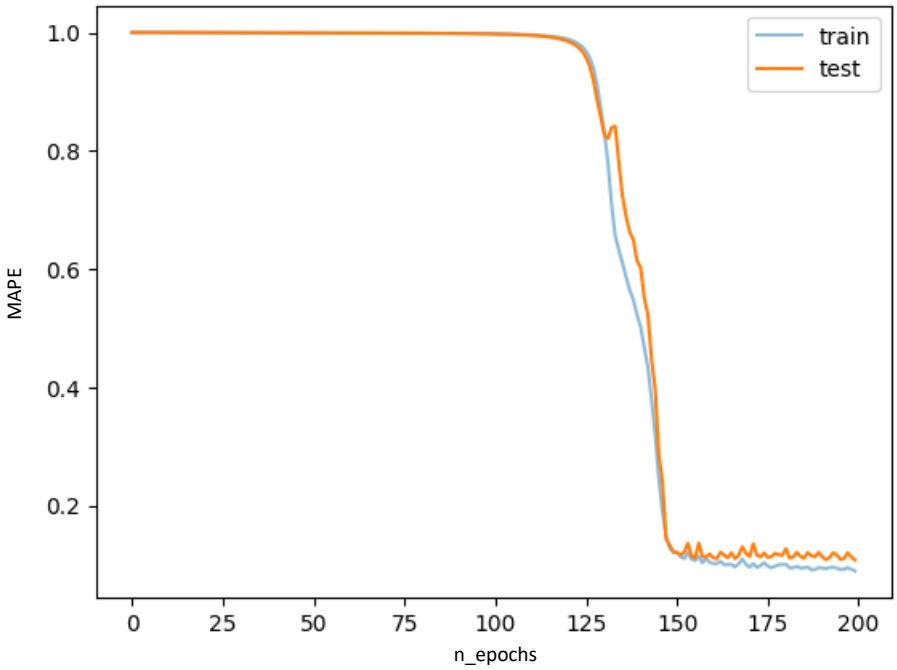

Figure 26 - Road inbound: MAPE curve after the application of the "Service level" "FTL" filter.

Given the significantly improved results from the “FTL” filter in the Road inbound, the model's performance was also evaluated using shipments originating from sources other than quotes. This attempt resulted in a MAPE value of 13.57%, although it did not surpass the lower value produced by the previous filter. However, a brief analysis revealed that the dataset without quotes contained significantly fewer lines compared to the dataset with shipments from other sources. The former had 1075 lines, while the latter had 6294 lines. Given that the difference in MAPE results was not significant and a larger dataset is better for training, it was decided to retain shipments from other sources in the dataset. The plot of loss over time during training is represented in Figure 27, showing that as the model trains over 200 epochs, its loss decreases, indicating learning and improved performance on the training data. However, there is some visible instability, like in the previous attempt, and the training and test performance show slight divergence, suggesting overfitting.

These results indicate that while the use of quotation observations alone produces slightly better results, the difference is minimal. This suggests that the model's performance is more influenced by market aspects related to truck availability rather than shipments from quotes, which means that using only FTL observations, consequently, results in better performance. These results can be explained by the following reasons:

→ Less impact from fluctuations associated with the Less than Truckload (LTL) market, since these can be very unpredictable due to multiple small shipments from a wide range of customers, in contrast to FTL customers who are prone to more stable shipment flows.

→ FTL carriers are able to negotiate long term fixed contracts, thus allowing for a greater control over tariffs.

→ FTL clients renew their contracts, despite market fluctuations, since they value predictability and reliability.

→ Less dependence on terminals and hubs that can be affected by market volatility while also reducing costs.

→ FTL transportation is more flexible and adaptable to any scenario. Besides, this transportation mode is more efficient and profitable in comparison to handling several partial loads.

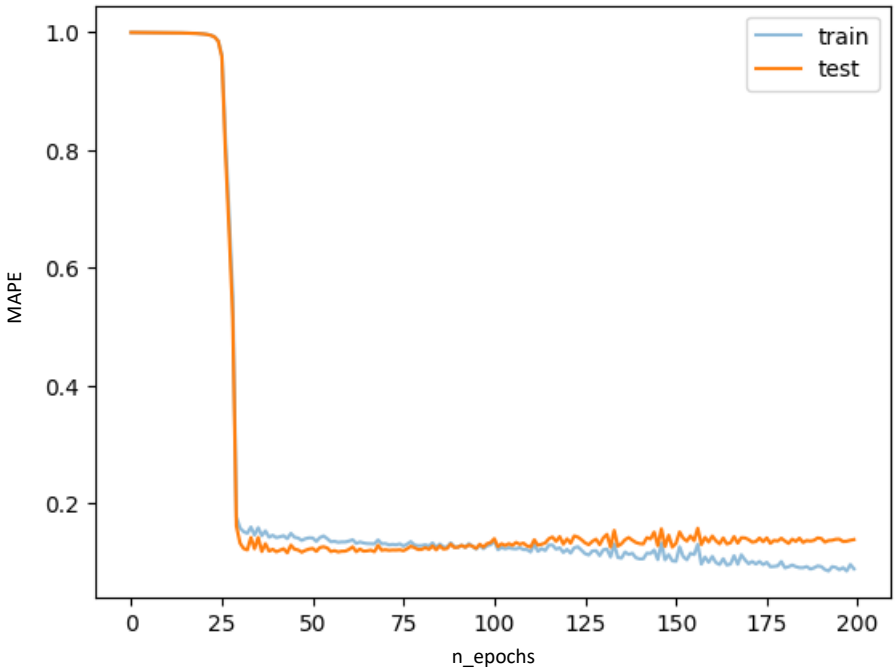

Figure 27 - Road inbound: MAPE curve after incorporating shipments that do not come from quotes.

After analyzing the MAPE results obtained for the FTL shipments, it was decided to analyze how these results translated into the RMSE output. As explained previously, RMSE is particularly effective in scenarios where values are near zero, a condition under which MAPE often falls short. This metric also allows us to better understand the magnitude of prediction errors, providing a clear perspective on model performance.

Table 7 presents a comprehensive summary of the hyperparameters employed and the corresponding model evaluation outcome, specifically focusing on RMSE as the primary evaluation metric. This compilation encompasses data from all sources, reflecting the decision to retain shipments from various origins within the dataset.

Table 7 – Road inbound: Hyperparameter settings and RMSE result

Loss Function - RMSE		
Hyperparameter	**Optimum value**	**Average RMSE across all folds (€)**
Number of hidden layers	4	271.94
Size of hidden layers	2000, 2000, 2000, 2000	
Learning rate	0.0000001	
Number of epochs	200	
Batch size	128	

Analyzing the RMSE plot depicted in Figure 28 reveals a consistent trend of gradual decrease as the model iterates through the training data, suggesting effective learning. Nonetheless, this descent in RMSE exhibits fluctuations, hinting at potential instability and susceptibility to overfitting. Eventually, the RMSE stabilizes at 271.94€, denoting a point of convergence where the model achieves optimal learning from the dataset. This result means that, on average, the model's predictions are deviated from the real values by approximately 271.94€.

In APPENDIX C, it is possible to see the minimum (0€) and maximum (2099.45€) values of the target variable "Costs", as well as the average (252.446€). With this information it is possible to conclude that, since the RMSE (271.94€) is a little higher than the average of the real costs (252.446€), the model still has some errors, which may indicate low accuracy. This suggests that the model can still be improved to provide more reliable estimates for the transport costs.

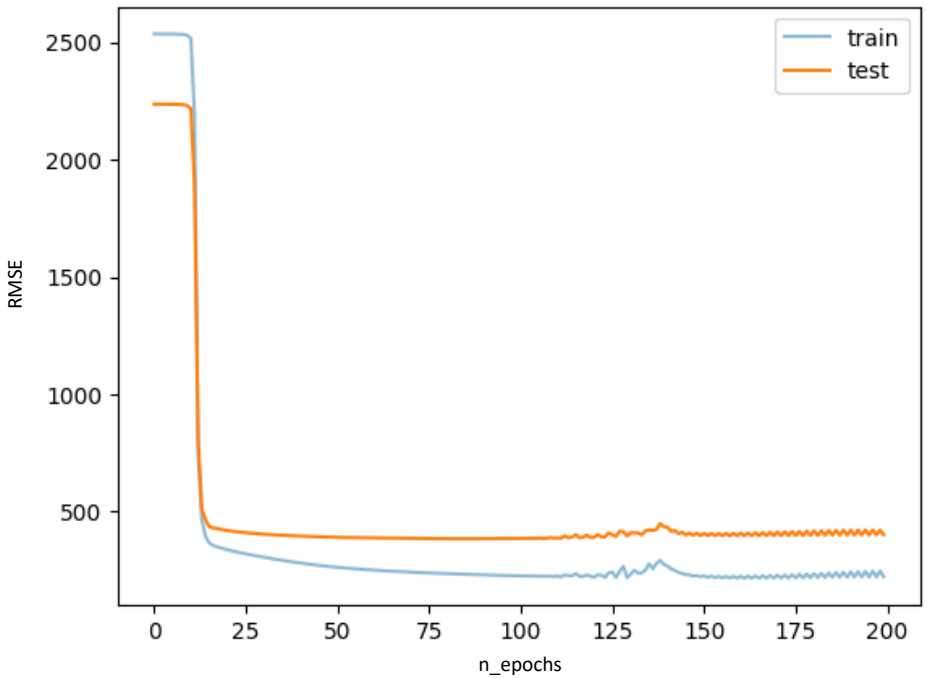

Figure 28 - Road inbound: RMSE curve.

4.2. Sea inbound

Table 8 compiles the hyperparameter values and the model evaluation results for each applied filter and modification.

Table 8 - Sea inbound: Hyperparameter settings and MAPE results

Loss Function - MAPE			
Filter / modification applied	**Hyperparameter**	**Optimum value**	**Average MAPE across all folds (%)**
Original pre-processing	Number of hidden layers	4	67.9
	Size of hidden layers	120, 60, 30, 15	
	Learning rate	0.1	
	Number of epochs	300	
	Batch size	128	
Addition of seasonality	Number of hidden layers	3	45.78
	Size of hidden layers	200, 100, 50	
	Learning rate	0.1	
	Number of epochs	200	
	Batch size	32	

Filter by “Shptype” = “D” (full containers)	Number of hidden layers	3	54.38
	Size of hidden layers	100, 50, 50	
	Learning rate	0.1	
	Number of epochs	40	
	Batch size	32	
Addition of containerized freight index	Number of hidden layers	3	55
	Size of hidden layers	200, 100, 50	
	Learning rate	0.1	
	Number of epochs	200	
	Batch size	32	

Similar to the Road case, the model's performance for the Sea case was initially evaluated using only the basic preprocessing steps outlined in subchapter 3.1.2, without any further modifications. The model produced a MAPE value of 67.9%. As shown in Figure 29, although the curves exhibit some instability, they converge and stabilize by the end of the 300 epochs.

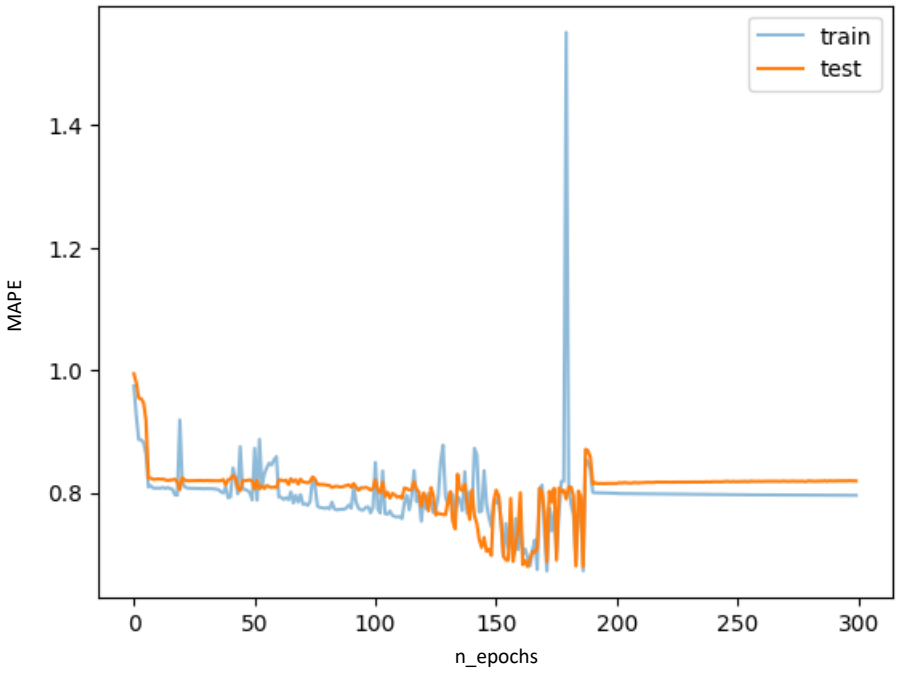

Figure 29 - Sea inbound: MAPE curve in the original pre-processing.

Incorporating additional data related to seasonality, similar to what happened in the Road inbound case, did not result in significant improvements. Although the MAPE value decreased by 22.12%, achieving a MAPE of 45.78%, this change was not substantial given that the result is still

very far from the desired one. The MAPE curves shown in Figure 30 display considerable instability, and the widening gap between them suggests potential overfitting.

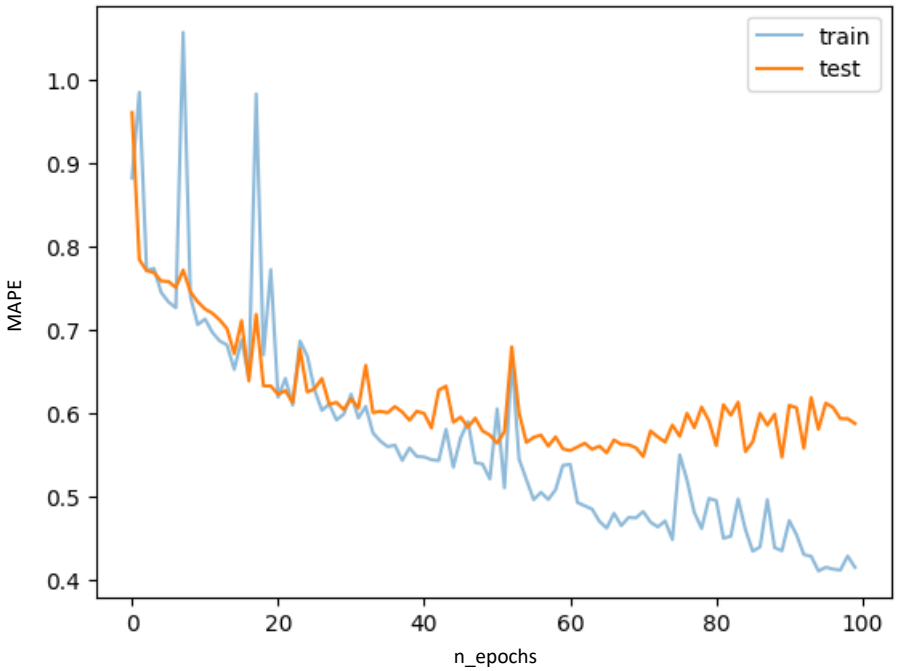

Figure 30 - Sea inbound: MAPE curve after adding the seasonality.

Similar to the efforts made for the Road inbound, various experiments were conducted for the Sea inbound to refine the model and stabilize the MAPE outputs. Given the success of the "FTL" filter in improving the model performance, a comparable filter was applied in this case. This filter involved retaining only the observations related to the transportation of full containers within the dataset. However, the results were not very encouraging, with a MAPE value of 54.38%. Additionally, Figure 31 shows that although the MAPE value decreases over the course of 40 epochs, there are signs of overfitting towards the end of the curve.

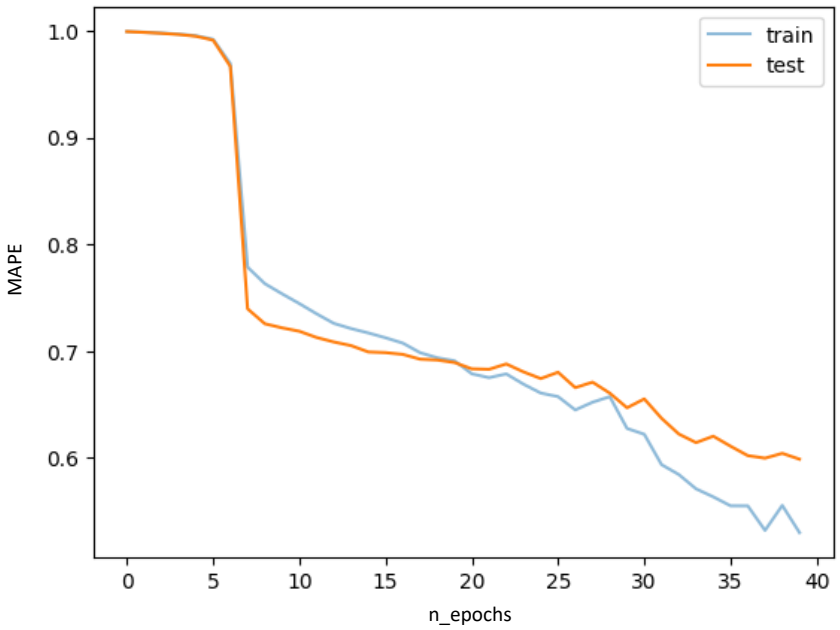

Figure 31 - Sea inbound: MAPE curve after the application of the "Shptype" "D" filter.

To try to improve the results, information on freight prices for container transport on the market was added to the data set, these data was taken from the Trading Economics website (2024). However, the results did not improve and generated a MAPE of 55%. As can be seen in Figure 32, there is some instability, and the increasing gap between train and test curves indicates the presence of overfitting. This result reinforces the complexity and volatility of maritime transportation. Based on expert advice, we have concluded that, inherent in maritime transportation, there are many factors that make it difficult to accurately predict operating costs, even for full containers. The main factors influencing maritime transport are the following ones:

→ Global economic fluctuations.

→ Variations in ship capacity.

→ Unpredictable weather impacts.

→ Port congestion.

→ Variable rates.

→ Constant change of regulations.

→ Changes in the carriers' internal policies.

→ Geopolitical instability.

These issues contribute to a high-cost variability and not only introduce significant uncertainties, but also make accurate modeling through machine learning techniques difficult,

since they affect costs in a non-linear way. Taking all these factors into account, and with the aim of improving the robustness of our model, it was decided to incorporate freight prices for container transportation, obtained from the selected data source mentioned above. Although the inclusion of information from sources such as Trading Economics was an important step, are needed more sophisticated analytical approaches that combine empirical data with an in-depth understanding of the economic and operational contexts of maritime transportation to achieve more accurate forecasts.

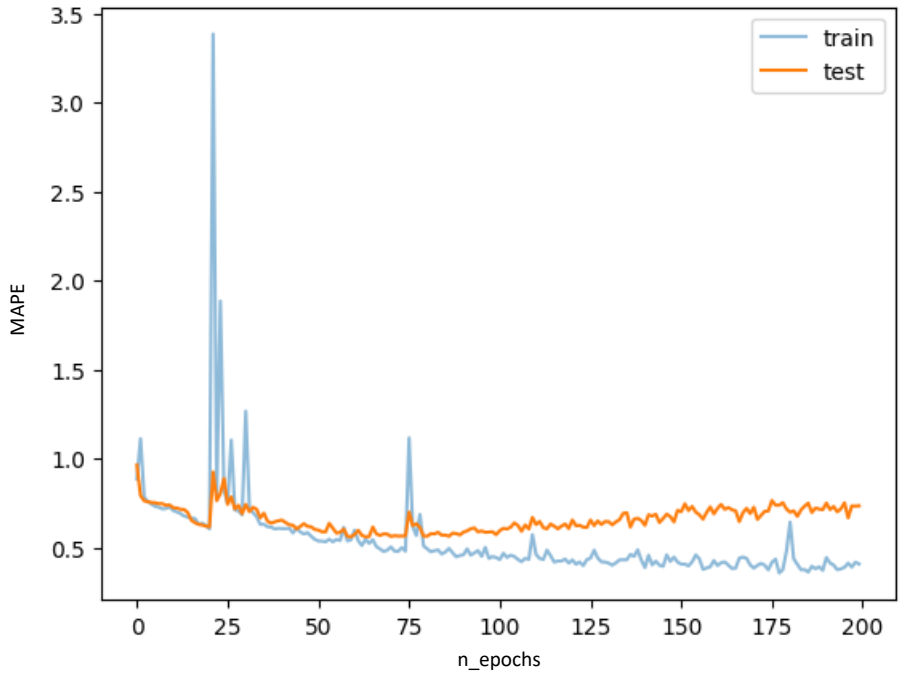

Figure 32 - Sea inbound: MAPE curve after adding the containerized freight index.

Similarly to the approach taken for road transportation costs predictions in subchapter 4.1, RMSE was employed for sea transportation costs predictions because it effectively explains the magnitude of prediction errors, offering a transparent assessment of model performance. This metric enhances our understanding of how well the model forecasts actual sea transportation costs patterns.

Table 9, as Table 8, presents the hyperparameters employed and the corresponding model evaluation outcome, specifically focusing on RMSE as the evaluation metric. This compilation encompasses all the filters and modifications done through the previous steps.

Table 9 - Sea inbound: Hyperparameter settings and RMSE result

Loss Function - RMSE		
Hyperparameter	**Optimum value**	**Average RMSE across all folds (€)**
Number of hidden layers	3	1524.4
Size of hidden layers	2000, 1000, 500	
Learning rate	0.0000001	
Number of epochs	200	
Batch size	32	

In Figure 33 it is initially evident that there might be underfitting, as indicated by the test curve being below the training curve. However, this underfitting appears to diminish towards the end of the 200 epochs. The curves also display fluctuations, suggesting potential instability in the model's performance. Despite this, the RMSE across all 5 folds is 1524.4€, which corroborates the MAPE results produced by the model. This result means that, on average, the model's predictions are deviated from the real values by approximately 1524.4€.

In APPENDIX D, it is possible to see the minimum (6.5€) and maximum (54204.09€) values of the target variable "Housecosts", as well as the average (2533.057€). With this information it is possible to conclude that, although the RMSE (1524.4€) is lower than the average of the real costs (2533.057€), there is still a significant margin of error in predictions. This suggests that the model, in its current form, is not providing reliable estimates for the transport costs, and consequently needs to be adjusted.

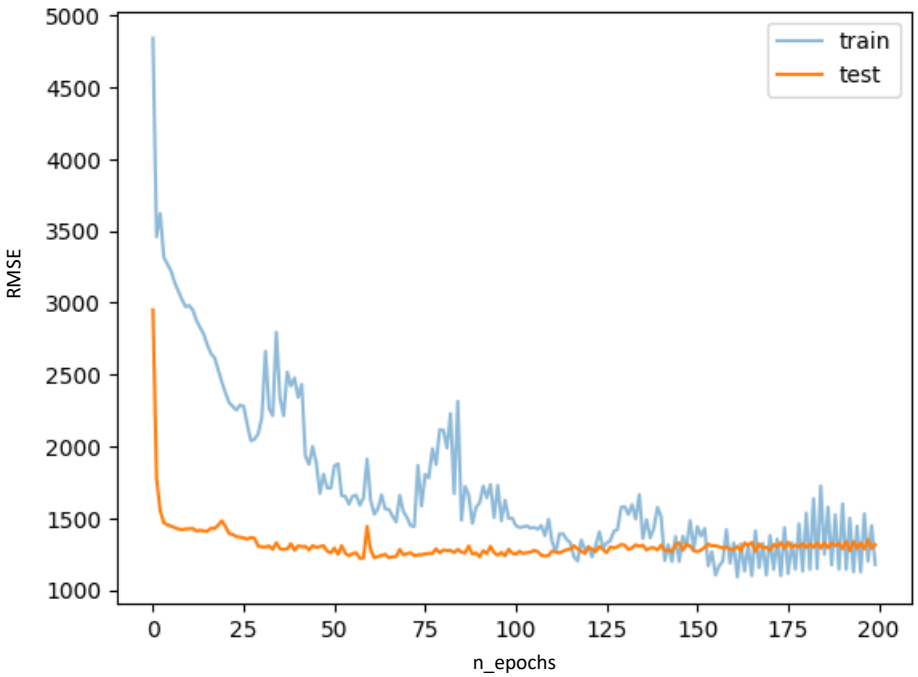

Figure 33 - Sea inbound: RMSE curve.

4.3. Computational resources

While developing and training ML models it is important to monitor the utilization of computational resources, since by using them efficiently, they promote a more optimized performance and help in managing costs. Additionally, to assign the appropriate hardware and resource allocation to the different ML models, understanding the required operational demands is essential.

The models developed in this work had specific resource utilization characteristics that were instrumental in evaluating its efficiency and scalability. During training, the models used, on average, the following computational resources:

- 3.55% of the total CPU capacity, which has a total processing power of 2.22 Gigahertz.
- 10.95 Gigabytes of RAM, indicating its memory demands.
- 21.91 Megabytes of GPU memory usage, with 45.88 Megabytes reserved for processing tasks.

Overall, the models demonstrate efficient resource usage with minimal CPU and GPU demands while utilizing a reasonable amount of RAM.

4.4. Discussion

After running both models and analyze the main results for the considered scenarios, it is necessary to organize and highlight the main insights and major findings. The primary focus was on developing a predictive model to estimate transportation costs for different type of goods to later use transfer learning to improve the accuracy of cost predictions for modular construction.

The first significant outcome was the successful training of a neural network model using Road inbound transportation data. This model demonstrated a strong ability to predict transport costs based on key variables such as distance, weight, and transport mode. The results showed that the model could accurately capture the relationships between these factors and the associated costs, providing reliable forecasts that could potentially optimize Road inbound logistics planning.

After running the model, the most accurate predictions were achieved by focusing exclusively on FTL shipments, resulting in a MAPE of 9.3%. The success of the FTL filter can be attributed to several key factors, such as the reduced influence of market fluctuations, better rate control, and enhanced customer loyalty. These elements allowed the model to more effectively capture the specific dynamics of the FTL market, leading to greater accuracy and reliability. For carriers, FTL transportation provides a more predictable and stable business model, characterized by operational efficiency, strong customer relationships, and better management of rates and routes. These advantages make this type of transportation less vulnerable to market volatility and ensure consistent truck availability. When comparing the first MAPE result of 20.7% with the one obtained when only FTL shipments were used (9.3%), it can be safely affirmed that the latter is a promising result, despite having a slightly high value for the RMSE (271.94€) compared to the average value of the real costs (252.446€), the improvements the model has displayed are still positive.

For the Sea inbound scenario, the results were less favorable compared to the Road inbound, with the best MAPE reaching only 45.78%. These less favorable outcomes were an important insight into the difficulty of capturing the volatile and intricate nature of maritime transport costs. Even after further analysis to understand the complexity of the maritime sector and the highly unstable cost structure, while also considering the interconnected factors that influence the maritime transport cost, the attempt to incorporate an external data source regarding freight prices for container transport on the market was unsuccessful and generated a MAPE value of 55%. This value aligns perfectly with the 1524.4€ of RMSE. These findings emphasize the need for more sophisticated modelling approaches in addition to standard machine learning technique. This can include the incorporation of different types of external data sources and/ or real-time data, adaptive algorithms and a comprehensive analysis of risk factors.

Ultimately, it was not the architecture of the model that limited its performance but rather the commercial complexities of maritime transport. Addressing these complexities requires models that can adapt to the rapidly changing variables of the shipping sector.

blank page

5. CONCLUSION

In this chapter are summarized the main conclusions and insights obtained from all the research made as well as the development and evaluation of the models for predicting transportation costs in construction. Additionally, are highlighted the implications and contributions of the results to the field of modular construction logistics. Finally, the limitations faced during the study are discussed and are proposed future directions in order to improve the model's effectiveness and applicability.

5.1. Final conclusions

In the making of this dissertation, it was possible to highlight many aspects and leverage the knowledge of a topic that is still under researched and has very little, if any, relevant information. Modular construction is still an exception in most countries, so it still raises many doubts about its adoption over conventional construction. As Liew et al. (2019) point out, it is mainly the major nations such as the USA, Japan and China that have already adopted this new approach to construction, mainly in structures with large, “repetitive” buildings, such as hospitals, offices, student residences, among others. However, this reality is very different in Portugal. According to Ribeiro et al. (2022), Portugal is limited by the lack of qualified professionals and the fact that the construction sector is not as advanced as it should be. This “aversion to the unknown”, also identified by McKinsey (2019), is also felt in other parts of the world. These contribute to the lack of data on MC, but more specifically on the transportation of modules, which due to their size may require special transportation. This was one of the motivations behind this study, which focuses on developing two predictive models for the transportation costs (road and sea) of raw materials in modular construction in order to close the gap in transportation cost predictions for modular construction and aggregate information in order to pave the way for the future implementation of transfer learning, which will extend this study to the modules specifically.

In response to the first research question “What are the key factors that influence the transferability of knowledge between different transportation domains?”, the literature review on transfer learning revealed that the transferability inherent in this approach, which can promote learning about a certain domain through a related domain and then transferring it, can be affected by different factors. Pan and Yang (2010) pointed out that when implementing this technique, the users must take into consideration three questions: “when to transfer”, “what to transfer” and “how to transfer”. Yosinski et al. (2014) stated that, depending on the location of the neural network where the transfer of features takes place (bottom, middle or top), the optimization of this

method can be affected. The authors also pointed out that transferability problems can also arise if the source task and target task are too far apart.

To develop the predictive models, we chose to follow the DSR methodology in parallel with the CRISP-DM methodology. In Chapter 3, additionally to the description of the data understanding, preparation, and modeling processes for the inbound data, it was also decided to outline the intended approach for the outbound data, which, due to unforeseen delays and dependence on external sources, could not be obtained. Based on this description, elaborated based on the transfer learning literature review, it was possible to verify that the data from the source domain and the target domain derive from spaces with similar features and may share some common features, such as distance, weight, type of cost and mode of transportation. This statement answers the second research question “How does the available data on transport costs for other goods contribute to forecasting transport costs in modular construction?”, as it suggests that the source task and the target task are not far from each other, which indicates good transferability for this problem.

After running both models and analyzing the main results for the scenarios considered, it was possible to emphasize the good performance of the neural network model using Road inbound transportation data, which was able to achieve a MAPE of 9.3% when only FTL shipments were used. This value reflects the stability, efficiency and good logistics of the FTL market. On the other hand, the model for the Sea inbound only generated a MAPE value of 45.78%. Although the results for sea transport were not equally positive, they were crucial for a deeper understanding of the difficulty of capturing the volatile and intricate nature of maritime transport costs. Even after numerous attempts to make the model more robust, such as providing the model with data related to freight prices for container transport on the market, it was not possible to optimize the MAPE.

The findings obtained after the analysis of the results, namely the improvement of performance with the use of FTL shipments in Road transport and the need to combine traditional neural networks with more sophisticated methods such as incorporation of external data sources and real-time data in Sea transport, offer a clear answer to the third research question: “How can deep learning models for predicting transport costs of different materials can be optimized based on performance analysis and data driven improvements?”

Although the research made notable progress, particularly in developing and training the model using inbound road transportation data, it was significantly compromised by delays in obtaining the outbound transportation data. This delay has impeded the immediate application of transfer learning, a key component in improving the model's ability to predict transportation costs in

modular construction. Despite this setback, the successful development of the input model lays a solid foundation for the future advances described in subchapter 5.2.

While the absence of output data has temporarily limited the scope of the project, the research has successfully demonstrated the potential of machine learning, particularly in the use of transfer learning to improve the prediction accuracy in logistics. This work offers a scalable solution for the prediction of transportation costs, opening the way for future innovations in modular construction logistics.

Overall, this work has contributed to the discovery of valuable insights related to MC by exploring an emerging topic, as well as the identification of gaps in relation to the transportation of modules and the prediction of their costs. This work also identified key points for future approaches in relation to the unpredictability of costs in maritime transportation, particularly in relation to market volatility and the complex nature of transportation costs. It was also possible to provide a baseline for future research, regarding both the integration of refinements to improve optimization and the future application of transfer learning.

In conclusion, although the project has made significant progress, the completion of the transfer learning approach will be essential to fully unlock the model's potential. Once the outbound data is available, this next step will eventually lead to more accurate and reliable cost predictions for modules transportation and provide a powerful tool for optimizing transportation costs. In the final analysis, this work supports the efficient and cost-effective management of modular construction projects, contributing valuable knowledge and methodologies to the field.

5.2. Limitations and future work

Despite the fact that this research has made significant contributions to the development of a model for predicting transportation costs in modular construction, it has encountered a significant number of limitations that have affected the extent and depth of the study. These limitations provide important context for the interpretation of the results and highlight areas that require further investigation.

A major challenge faced during this research was the limited available data and information on modular construction, particularly in the context of predicting transportation costs. Modular construction is a relatively new and evolving field and, as a result, there is a lack of relevant literature and industry reports on this subject. This lack of research and any existing data impeded the achievement of a comprehensive basis for the study and also limited the ability to compare the model's performance with similar studies. This limited available data has also restricted the process

of training and validating the model, affecting the accuracy and generalization of the results. In addition, the lack of precedents in the literature meant that the research had to rely on primary data collection and new methodologies, which caused extra complexity and uncertainty in the development of the models.

Other significant limitation was the lack of existing studies or models that specifically focused on predicting transportation costs in modular construction. At the same time that there is considerable research on transportation cost prediction in general logistics and supply chain management, these studies do not fully address the particular challenges and characteristics of modular construction logistics. This gap in the literature made it necessary to develop a new model from scratch, without the benefit of building the models with established methodologies or references to guide the process. This limitation also made it difficult to validate the model's performance, as there were no directly comparable studies to serve as a reference.

The delay in obtaining outbound transportation data, which was essential for completing the model and applying the planned transfer learning approach, was a critical limitation of this research. The unavailability of outbound data has limited the study to a partial application of the model, which focuses only on inbound transportation. Consequently, the planned transfer learning approach, which could have significantly improved the model’s adaptability and performance, could not be implemented within the scope of this study.

The research also encountered challenges in accurately predicting sea transport costs due to the complexity and variability inherent in sea transport costs. Factors such as global economic fluctuations, variations in ship capacity, weather conditions, port congestion and changes in carriers' internal policies introduce a high level of uncertainty into the cost structure, which the model struggled to effectively capture. The complexity and volatility of sea transport contributed to higher MAPE values in the case of inbound sea transport compared to inbound road transport. This limitation highlights the difficulty of applying standard machine learning techniques to highly variable domains.

Although some areas remain unexplored or require further refinement if the potential of the methodologies introduced is to be fully exploited, the research carried out in this thesis has established a solid basis for predicting transportation costs in modular construction. The key directions for future research have already been outlined in Chapter 3.3 and consist of one main point: the application of the parameter-based transfer learning approach when the outbound transportation data becomes available. The delay in obtaining this data prevented its inclusion in this study, but it is expected that its integration will provide good results by using the knowledge

learned in the inbound model and applying it to the specific outbound model. This method will reduce dependence on large amounts of outbound data, accelerating the development of the model and improving its adaptability to different transport scenarios.

Considering the ongoing difficulties in accurately predicting sea transport costs, future research should also explore the integration of additional external data sources. This may include real-time data on market conditions, fuel prices, regulatory changes and global economic indicators. The incorporation of this data could improve the model's ability to consider the high variability and complexity of sea transport costs.

The prediction of costs in the transportation of modular construction items has a direct influence on the selection of the mode of transportation. A potential step forward in this work would be to incorporate the prediction of CO_2 emissions associated with the transport of modules, making it possible not only to optimize costs, but also to reduce environmental impact. By predicting the emissions from different types of transportation it would be possible to determine which are the most sustainable alternatives and identify ways to reduce the carbon footprint of the transportation process and so promote sustainable building practices in line with global emissions reduction targets.

blank page

BIBLIOGRAPHICAL REFERENCES

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2623–2631. https://doi.org/10.1145/3292500.3330701

Altaf, M. S., Bouferguene, A., Liu, H., Al-Hussein, M., & Yu, H. (2018). Integrated production planning and control system for a panelized home prefabrication facility using simulation and RFID. *Automation in Construction*, *85*, 369–383. https://doi.org/10.1016/j.autcon.2017.09.009

Arturo Garza-Reyes, J., Oraifige, I., Soriano-Meier, H., Forrester, P. L., & Harmanto, D. (2012). The development of a lean park homes production process using process flow and simulation methods. *Journal of Manufacturing Technology Management*, *23*(2), 178–197. https://doi.org/10.1108/17410381211202188

Bao, Y., Li, Y., Huang, S.-L., Zhang, L., Zheng, L., Zamir, A., & Guibas, L. (2022). *An Information-Theoretic Approach to Transferability in Task Transfer Learning*. https://doi.org/https://doi.org/10.48550/arXiv.2212.10082

Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of Methods Microbiological Journal of Microbiological Methods*, *43*, 3–31. https://doi.org/https://doi.org/10.1016/S0167-7012(00)00201-3

Bashir, D., Montañez, G. D., Sehra, S., Segura, P. S., & Lauw, J. (2020). An Information-Theoretic Perspective on Overfitting and Underfitting. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12576 LNAI*, 347–358. https://doi.org/10.1007/978-3-030-64984-5_27

Bergstra, J. B. Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*, 281–305. http://scikit-learn.sourceforge.net.

Bertram, N., Fuchs, S., Mischke, J., Palter, R., Strube, G., & Woetzel, J. (2019). *Modular construction: From projects to products*. https://www.ivvd.nl/wp-content/uploads/2019/12/Modular-construction-from-projects-to-products-full-report-NEW.pdf

Bhaya, W. (2017). Review of Data Preprocessing Techniques. *Journal of Engineering and Applied Sciences*, *12*(16), 4102–4107. https://doi.org/10.3923/jeasci.2017.4102.4107

Blanco, J. L., Fuchs, S., Matt, P., & Ribeirinho, M. J. (2018). *Artificial intelligence: construction technology's next frontier*. Building Economist.

Blitzer, J., Mcdonald, R., & Pereira, F. (2006). Domain Adaptation with Structural Correspondence Learning. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), Association for Computational Linguistics*, 120–128. https://aclanthology.org/W06-1615.pdf

Boafo, F. E., Kim, J. H., & Kim, J. T. (2016). Performance of modular prefabricated architecture: Case study-based review and future pathways. In *Sustainability (Switzerland)* (Vol. 8, Issue 6). MDPI. https://doi.org/10.3390/su8060558

Busto, P. P., Iqbal, A., & Gall, J. (2019). *Open Set Domain Adaptation for Image and Action Recognition*. https://doi.org/https://doi.org/10.48550/arXiv.1907.12865

Cao, Z., Long, M., Wang, J., & Jordan, M. I. (2018). Partial Transfer Learning with Selective Adversarial Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2724–2732. https://doi.org/10.1109/CVPR.2018.00288

Cawley, G. C., & Talbot, N. L. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, *11*, 2079–2107. https://www.jmlr.org/papers/volume11/cawley10a/cawley10a.pdf

Costa-Luis, C. O. da. (2019). tqdm: A Fast, Extensible Progress Meter for Python and CLI. *Journal of Open Source Software*, *4*(37), 1277. https://doi.org/10.21105/joss.01277

Das, P., Kashem, A., Hasan, I., & Islam, M. (2024). A comparative study of machine learning models for construction costs prediction with natural gradient boosting algorithm and SHAP analysis. *Asian Journal of Civil Engineering*. https://doi.org/10.1007/s42107-023-00980-z

Ding, B., Qian, H., & Zhou, J. (2018, June). Activation functions and their characteristics in deep neural networks. *2018 Chinese Control And Decision Conference (CCDC)*. https://doi.org/10.1109/CCDC.2018.8407425

El-Abidi, K. M. A., & Ghazali, F. E. M. (2015). Motivations and Limitations of Prefabricated Building: An Overview. *Applied Mechanics and Materials*, *802*, 668–675. https://doi.org/10.4028/www.scientific.net/amm.802.668

Erickson, B. J., Korfiatis, P., Akkus, Z., Kline, T., & Philbrick, K. (2017). Toolkits and Libraries for Deep Learning. *Journal of Digital Imaging*, *30*(4), 400–405. https://doi.org/10.1007/s10278-017-9965-6

Farahani, A., Voghoei, S., Rasheed, K., Arabnia, H. R., Arabnia, H. R., & Deligiannidis, L. (2021). A Brief Review of Domain Adaptation. In *Advances in Data Science and Information Engineering. Transactions on Computational Science and Computational Intelligence* (pp. 877–894). https://doi.org/https://doi.org/10.1007/978-3-030-71704-9_65

Fortmann-Roe, S. (2012). *Understanding the Bias-Variance Tradeoff*.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics*, *1*(1). https://doi.org/10.1186/s41044-016-0014-0

Ghannad, P., & Lee, Y.-C. (2023). Optimizing Modularization of Residential Housing Designs for Rapid Postdisaster Mass Production of Housing. *Journal of Construction Engineering and Management*, *149*(7). https://doi.org/10.1061/(asce)co.1943-7862.0002390

Goodfellow, I. B. Y. C. A. (2016). *Deep Learning*. MIT Press.

Haykin, S. (2008). *Neural Networks and Learning Machines* (3rd ed.). Pearson PLC.

Huang, J., Li, Y. F., & Xie, M. (2015). An empirical analysis of data preprocessing for machine learning-based software cost estimation. *Information and Software Technology*, *67*, 108–127. https://doi.org/10.1016/j.infsof.2015.07.004

Imambi, S., Prakash, K. B., & Kanagachidambaresan, G. R. (2021). PyTorch. In: Prakash, K.B., Kanagachidambaresan, G.R. (eds) Programming with TensorFlow. *EAI/Springer Innovations in Communication and Computing*. https://doi.org/https://doi.org/10.1007/978-3-030-57077-4_10

Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A Review of Deep Transfer Learning and Recent Advancements. *Technologies*, *11*(2). https://doi.org/https://doi.org/10.3390/technologies11020040

INEGI. (2024). *Inegi - driving science and inovation*. https://www.inegi.pt/pt/

Jabbar, H. K., & Khan, D. R. Z. (2015). *METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY)* (J. Stephen, H. Rohil, & S. Vasavi, Eds.).

James, G. (Gareth M., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: with applications in R*. https://doi.org/10.1007/978-1-4614-7138-7

Jialin Pan, S., Kwok, J. T., & Yang, Q. (2008). Transfer Learning via Dimensionality Reduction. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence* , 677–682.

Kouw, W. M., & Loog, M. (2019). *An introduction to domain adaptation and transfer learning*. https://doi.org/https://doi.org/10.48550/arXiv.1812.11806

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling* (1st ed.). https://doi.org/https://doi.org/10.1007/978-1-4614-6849-3

Lawson, R. M., Ogden, R. G., & Bergin, R. (2012). Application of Modular Construction in High-Rise Buildings. *Journal of Architectural Engineering*, *18*(2), 148–154. https://doi.org/10.1061/(asce)ae.1943-5568.0000057

Lewis, C. D. (1983). Industrial and business forecasting methods. *Journal of Forecasting*, *2*(2), 194–196. https://doi.org/10.1002/for.3980020210

Liew, J. Y. R., Chua, Y. S., & Dai, Z. (2019). Steel concrete composite systems for modular construction of high-rise buildings. *Structures*, *21*, 135–149. https://doi.org/10.1016/j.istruc.2019.02.010

Liu, C., M.E. Sepasgozar, S., Shirowzhan, S., & Mohammadi, G. (2022). Applications of object detection in modular construction based on a comparative evaluation of deep learning algorithms. *Construction Innovation*, *22*(1), 141–159. https://doi.org/10.1108/CI-02-2020-0017

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *31st Conference on Neural Information Processing Systems (NIPS)*. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

McKinsey. Modular construction: From projects to products. (2019). https://www.mckinsey.com/capabilities/operations/our-insights/modular-construction-from-projects-to-products

Michael A. Nielsen. (2015). *Neural Networks and Deep Learning*.

MIT. (2023). *MIT Introduction to Deep Learning* [Video recording].

Mohsen, O., Asce, A. M., Petre, C., Mohamed, Y., & Asce, M. (2022). *Machine-Learning Approach to Predict Total Fabrication Duration of Industrial Pipe Spools*. https://doi.org/10.1061/JCEMD4

Moradibistouni, M., & Gjerde, M. (2017). Potential for Prefabrication to Enhance the New Zealand Construction Industry. *Back to the Future: The Next 50 Years, (51st International Conference of the Architectural Science Association (ANZAScA))*, *51*, 427–435.

Muraina, I. O. (2022). IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS: GENERAL CONCERNS FOR DATA SCIENTISTS AND DATA ANALYSTS. *7th INTERNATIONAL MARDIN ARTUKLU SCIENTIFIC RESEARCHES CONFERENCE*, 496–504. https://www.researchgate.net/publication/358284895

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Pan, S. J., Zhou, J. T., Tsang, I. W., & Yan, Y. (2014). Hybrid Heterogeneous Transfer Learning through Deep Learning. *Proceedings of the AAAI Conference on Artificial Intelligence, 28(1).* , 2213–2219. https://doi.org/https://doi.org/10.1609/aaai.v28i1.8961

Park, K., Ergan, S., & Feng, C. (2021). Towards Intelligent Agents to Assist in Modular Construction: Evaluation of Datasets Generated in Virtual Environments for AI training. *Proceedings of the ISARC*. https://par.nsf.gov/biblio/10388386

Pauls, A., & Yoder, J. A. (2018). *Determining Optimum Drop-out Rate for Neural Networks*. https://micsymposium.org/mics2018/proceedings/MICS_2018_paper_27.pdf

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Varoquaux, G., Gramfort, A., Thirion, B., Dubourg, V., Passos, A., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. In *Journal of Machine Learning Research* (Vol. 12).

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Plevris, V., Solorzano, G., Bakas, N. P., & Ben Seghier, M. E. A. (2022). INVESTIGATION OF PERFORMANCE METRICS IN REGRESSION ANALYSIS AND MACHINE LEARNING-BASED

PREDICTION MODELS. *World Congress in Computational Mechanics and ECCOMAS Congress*. https://doi.org/10.23967/eccomas.2022.155

Plotnikova, V., Dumas, M., & Milani, F. (2020). Adaptations of data mining methodologies: A systematic literature review. *PeerJ Computer Science*, *6*, 1–43. https://doi.org/10.7717/PEERJ-CS.267

Qasem, O. Al, Akour, M., & Alenezi, M. (2020). The Influence of Deep Learning Algorithms Factors in Software Fault Prediction. *IEEE Access*, *8*, 63945–63960. https://doi.org/10.1109/ACCESS.2020.2985290

Raitoharju, J. (2022). Convolutional neural networks. In *Deep Learning for Robot Perception and Cognition* (pp. 35–69). Elsevier. https://doi.org/10.1016/B978-0-32-385787-1.00008-7

Ramos, F. R., Pereira, M. T., Oliveira, M., & Rubio, L. (2023). THE MEMORY CONCEPT BEHIND DEEP NEURAL NETWORK MODELS: AN APPLICATION IN TIME SERIES FORECASTING IN THE E-COMMERCE SECTOR. *Decision Making: Applications in Management and Engineering*, *6*(2), 668–690. https://doi.org/10.31181/dmame622023695

Rasamoelina, A., Adjailia, F., & Sincak, P. (2020). A Review of Activation Function for Artificial Neural Network. *SAMI 2020 : IEEE 18th World Symposium on Applied Machine Intelligence and Informatics*, 281–286. https://doi.org/10.1109/SAMI48414.2020.9108717

Rashid, K. M., & Louis, J. (2020). Activity identification in modular construction using audio signals and machine learning. *Automation in Construction*, *119*. https://doi.org/10.1016/j.autcon.2020.103361

Reports and Data. (2019). *Artificial intelligence (AI) in construction market by technology (machine learning and deep learning, natural language processing), by component, by phase, by deployment type, by applications, by organization size, by end-use, and segment forecasts, 2016-2026*. www.reportsanddata.com/report-detail/artificial-intelligence-ai-in-construction-market

Ribeiro, A. M., Arantes, A., & Cruz, C. O. (2022). Barriers to the Adoption of Modular Construction in Portugal: An Interpretive Structural Modeling Approach. *Buildings*, *12*(10). https://doi.org/10.3390/buildings12101509

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., & Hadsell, R. (2016). *Progressive Neural Networks*. https://doi.org/https://doi.org/10.48550/arXiv.1606.04671

Saito, K., Yamamoto, S., Ushiku, Y., & Harada, T. (2018). *Open Set Domain Adaptation by Backpropagation*. https://doi.org/https://doi.org/10.48550/arXiv.1804.10427

Saltz, J. S., & Hotz, N. (2020). Identifying the most Common Frameworks Data Science Teams Use to Structure and Coordinate their Projects. *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, 2038–2042. https://doi.org/10.1109/BigData50022.2020.9377813

Shinde, P. P., & Shah, Dr. S. (2018). A Review of Machine Learning and Deep Learning Applications. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1–6. https://doi.org/10.1109/ICCUBEA.2018.8697857

Sivakumar, A., & Gunasundari, R. (2017). A Survey on Data Preprocessing Techniques for Bioinformatics and Web Usage Mining. *International Journal of Pure and Applied Mathematics*, *117*(20), 785–794. https://acadpubl.eu/jsi/2017-117-20-22/articles/20/68.pdf

Sousa, V. F. C., Silva, F. J. G., Pereira, T., Ferreira, L. P., Sá, J. C., & Nogueira, P. (2023). Development of an Affordable Cost Estimation Tool for Machined Parts. *Flexible Automation and Intelligent Manufacturing: The Human-Data-Technology Nexus. FAIM 2022. Lecture Notes in Mechanical Engineering*. https://doi.org/https://doi.org/10.1007/978-3-031-17629-6_32

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). *A Survey on Deep Transfer Learning*. https://doi.org/https://doi.org/10.48550/arXiv.1808.01974

Tan, Y., Li, Y., & Huang, S.-L. (2021). *OTCE: A Transferability Metric for Cross-Domain Cross-Task Representations*. https://doi.org/https://doi.org/10.48550/arXiv.2103.13843

Thai, H. T., Ngo, T., & Uy, B. (2020). A review on modular construction for high-rise buildings. In *Structures* (Vol. 28, pp. 1265–1290). Elsevier Ltd. https://doi.org/10.1016/j.istruc.2020.09.070

Torrey, L. S. J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242–264). IGI global.

Trading Economics. (2024, June 6). *Containerized Freight Index*. https://tradingeconomics.com/commodity/containerized-freight-index

Uzair, M., & Jamil, N. (2020, November 5). Effects of Hidden Layers on the Efficiency of Neural networks. *Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020*. https://doi.org/10.1109/INMIC50486.2020.9318195

Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2022). A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, *9*(2), 187–212. https://doi.org/10.1007/s40745-020-00253-5

Weiss, K., Khoshgoftaar, T. M., & Wang, D. D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1). https://doi.org/x

Williams, C. (2007). Research Methods. *Journal of Business & Economic Research-March*, *5*(3), 65–72.

Yi, W., Wang, S., & Zhang, A. (2020). Optimal transportation planning for prefabricated products in construction. *Computer-Aided Civil and Infrastructure Engineering*, *35*(4), 342–353. https://doi.org/10.1111/mice.12504

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). *How transferable are features in deep neural networks?* https://doi.org/https://doi.org/10.48550/arXiv.1411.1792

Yotov, K., Hadzhikolev, E., & Hadzhikoleva, S. (2020). Determining the Number of Neurons in Artificial Neural Networks for Approximation, Trained with Algorithms Using the Jacobi Matrix. *TEM Journal*, *9*(4), 1320–1329. https://doi.org/10.18421/TEM94-02

You, K., Long, M., Cao, Z., Wang, J., & Jordan, M. I. (2019). Universal domain adaptation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 2715–2724. https://doi.org/10.1109/CVPR.2019.00283

Zhang, H., Zhang, L., & Jiang, Y. (2019). Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems. *11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 1–6. https://doi.org/10.1109/WCSP.2019.8927876

Zhang, J., Ding, Z., Li, W., & Ogunbona, P. (2018). *Importance Weighted Adversarial Nets for Partial Domain Adaptation*. https://doi.org/https://doi.org/10.48550/arXiv.1803.09210

blank page

APPENDIX A

- **Road inbound variables' characteristics before pre-processing.**

Feature Name	Description	Type	Range (min – max)	Mean	Standard deviation	Nº Of Missing Values	Measurement Unit
File Number	Number of the shipping file	Numeric	-	-	-	0	
Quote Number	Number of the quote	Numeric	-	-	-	0	
Status	Status number of the shipping	Numeric	-	-	-	0	
Status Description	Satus description of the shipping (e g Invoiced, Delivered, Credit Note, etc)	Categoric	-	-	-	0	
Department	Department responsible of the shipment goods	Categoric	-	-	-	1	
Employee	Employee in charge of the shipping	Categoric	-	-	-	2	
Sale Type	Type of sale (direct or indirect sale)	Categoric	-	-	-	0	
Reference Principal	Reference of the client	Categoric	-	-	-	53798	
Agent Trip Reference	Reference of the trip agent (a third party that facilitates the transport of goods between two other parties)	Categoric	-	-	-	91716	
Agent Shipment Reference	Reference of the shipment agent (a third party that facilitates the transport of goods between two other parties)	Categoric	-	-	-	225468	
Cargo Collect Reference	Reference of the cargo collect	Categoric	-	-	-	212945	
Period No	Year and month of the loading	Numeric	-	-	-	0	
Service Level	Type of container transport (e g Full Truck load (FTL), Less than Truck load (LTL), etc)	Categoric	-	-	-	339	
Incoterm	International agreements on international transport of goods (e g EXW, DAP, etc)	Categoric	-	-	-	603	
Loading Country	Country where the loading was made	Categoric	-	-	-	376	
Unloading Country	Country where the unloading was made	Categoric	-	-	-	340	
Quantity	Quantity of packages	Numeric	(0 - 6725)	9.461	78.472	1	
Package	Type of package (coitainer, box, etc)	Categoric	-	-	-	400	

Tariff Type	Refers to the fees and rules that carriers apply for their services	Categoric	-	-	-	217685	
Principal	Client	Categoric	-	-	-	16	
Consignor	It is the party that sends a shipment to be delivered	Categoric	-	-	-	371	
Consignee	Entity that is financially responsible for receiving a consignment	Categoric	-	-	-	404	
Loading Date	Date when the loading was made	Date	-	-	-	28	
Loading Time	Time when the loading was made	Numeric	-	-	-	1727	
Unloading Date	Date when the unloading was made	Numeric	-	-	-	28	
Unloading Time	Time when the unloading was made	Numeric	-	-	-	79737	
Handling	Ground Handling Services	Categoric	-	-	-	1	
Loading Postal Code	Postal code where the loading was made	Categoric	-	-	-	387	
Unloading Postal Code	Postal code where the unloading was made	Categoric	-	-	-	597	
Loading Address	Address where the loading was made	Categoric	-	-	-	339	
Unloading Address	Address where the loading was made	Categoric	-	-	-	340	
Loading Address Place	Address place where the loading was made	Categoric	-	-	-	6542	
Unloading Address Place	Address place where the unloading was made	Categoric	-	-	-	4658	
Loading Hubzone	Hubzone where the loading was made	Categoric	-	-	-	1228	
Unloading Hubzone	Hubzone where the unloading was made	Categoric	-	-	-	1123	
Goods Number	Quantity of goods	Numeric	-	-	-	188326	
Goods Description	Type of goods	Categoric	-	-	-	50396	
Loading Index	It indicates how much weight the tires can carry	Numeric	(0 - 24000)	0.665	91.958	6	
Gross Weight	Gross cargo weight	Numeric	(0 - 304522.2	1730.222	5944.968	2	Kg

Volume	This value corresponds to the higher value between the cubic meters and the loading meters	Numeric	(0 - 3477.6)	2.699	10.522	9	m^3
Cubic Meters	Calculated volume of the cargo	Numeric	(0 - 3477.6)	2.006	10.421	114	
Loading Meteres	1 loading meter corresponds to 1 linear meter of cargo space on a truck	Numeric	(0 - 442.56)	1.11	3.385	45	m
Chargeable Weight	It is the weight proportional to the space (volume) that the goods occupy in truck, i e it is the weight that the customer actually pays for the transportation of the consignment or shipment	Numeric	(0 - 1158041)	2585.819	7688.299	309	Kg
Seguro	Indicates the presence or not of the insurance	Categoric	-	-	-	0	-
Seguro Type	Type of Insurance	Numeric	-	-	-	222337	
Despacho	Indicates the presence or not of dispatch	Categoric	-	-	-	0	
Plan Characteristics	Characteristics of the shipping plan	Categoric	-	-	-	220592	
Bordero Number	Number of the bordero (document detailing the movements or credits and debits of a period or a transaction)	Numeric	-	-	-	222028	
Cod/Cft	Cod/ Cft	Categoric	-	-	-	0	
Cod/Cft Value	Value of the Cod/ Cft	Numeric	-	-	-	225179	
Distance	Distance travelled	Numeric	(0 - 161841)	1719.017	1183.751	128999	Km
Revenue	Revenue with the shipping	Numeric	(-750 - 130695)	316.03	797.255	0	€
Cost	Cost with the shipping	Numeric	(-4205.22 1093000)	280.438	725.721	0	
Result	Result obtained after subtracting revenue from cost	Numeric	(-8365 - 21395)	35.592	215.547	0	
Max Width	Maximum cargo width	Numeric	(0 - 88)	0.658	0.802	109891	m
Sum Width	Sum of widths	Numeric	(0 - 240)	0.756	1.159	109891	
Max Height	Maximum cargo height	Numeric	(0 - 94)	0.798	1.173	109909	
Sum Height	Sum of heights	Numeric	(0 - 120)	0.913	1.402	109909	
Max Length	Maximum cargo length	Numeric	(0 - 80)	1.087	0.88	109892	
Sum Length	Sum of lengths	Numeric	(0 - 80)	1.27	1.426	109892	
Cost Adr	Costs with dangerous cargo	Numeric	(0 - 66.69)	16	12.932	225257	

Cost Temp Control	Costs with temperature control	Numeric	*			225468	
Cost Fuel	Costs with fuel	Numeric	(-282.05 - 3800)	1.29	14.768	145127	
Cost Customs	Customs costs	Numeric	(0 - 1590)	51.678	72.251	219472	
Cost Seguro	Costs with insurance	Numeric	(0 - 717.16)	12.695	37.266	219884	
Cost Other	Other costs	Numeric	(-5405 - 109300)	58.82	454.66	33894	
Sale Adr	Invoicing for dangerous cargo	Numeric	(46.378 - 250)	28.389	46.378	225222	€
Sale Temp Control	Invoicing for temperature control	Numeric	(0 - 300)	0.911	15.079	224974	
Sale Fuel	Invoicing of fuel	Numeric	(-136.9 - 7285.96)	5.96	49.655	45129	
Sale Customs	Invoicing of dispatch	Numeric	(0 - 4275)	91.214	148.665	217502	
Sale Seguro	insurance invoicing	Numeric	(0 - 2235.29)	60.383	111.431	221039	
Sale Transport Price	freight invoicing	Numeric	(-750 - 130695)	519.586	923.079	112805	
Sale Other	Other invoicings	Numeric	(-225.63 - 45957.99)	63.599	443.924	105613	

* It was not registered any data regarding this variable.

APPENDIX B

- **Sea inbound variables' characteristics before pre-processing.**

Feature Name	Description	Type	Range (min – max)	Mean	Standard deviation	Nº Of Missing Values	Measurement Unit
Unid	Number of the shipping file	Numeric	-	-	-	0	-
Jobno	Number of the shippment	Numeric	-	-	-	0	
Master	Shipping code	Categoric	-	-	-	12007	
Biztype	Shipping process (air export, sea import, etc.)	Categoric	-	-	-	0	
Shptype	Shipping type (house or direct)	Categoric	-	-	-	0	
Ownerid	id	Categoric	-	-	-	0	
Bizscope	Business scope	Categoric	-	-	-	77	
Poletddate	Date of loading	Categoric	-	-	-	682	
Podetadate	Date of unloading	Categoric	-	-	-	776	
Jobdate		Categoric	-	-	-	0	
Polctry	Country of loading port	Categoric	-	-	-	0	
Polcity	City of loading port	Categoric	-	-	-	0	
Podctry	Country of destination port	Categoric	-	-	-	13	
Podcity	City of destination port	Categoric	-	-	-	0	
Destctry	Country where the unloading was made	Categoric	-	-	-	14525	
Destcity	City where the unloading was made	Categoric	-	-	-	14517	
Incoterm	International agreements on international transport of goods (e.g. EXW, DAP, etc.)	Categoric	-	-	-	0	
Service Type	Service methods (door-to-door, door-to-port, etc.)	Categoric	-	-	-	0	
Carrier Code	Code of the carrier	Categoric	-	-	-	1483	
Carrier Name	Name of the carrier	Categoric	-	-	-	1498	
Cwgt	Chargeable weight. It is the weight proportional to the space (volume) that the goods occupy in truck, i.e. it is the weight that the customer actually pays for the transportation of the consignment or shipment.	Numeric	(0 - 80000)	179.164	1578.072	119	Kg

Teus	Twenty Foot Equivalent Unit. International standardised container type. A twenty foot unit measures about 6 meters.	Numeric	(0 - 60)	1.337	2.171	7915	-
Houseinvoice	Invoice total. Total of [invoices_xxxxx] columns.	Numeric	(0 - 720454)	2753.714	15690	0	
Housecosts	Total costs. Total of the [costs_xxxxx] columns.	Numeric	(-1257.74 - 593810.98)	2295.106	12842.124	0	
Invoices Frete	Freight invoicing	Numeric	(-111.98 - 720454)	2105.128	15063.999	0	
Costs Frete	Freight costs	Numeric	(-3798.4 - 589027)	1883.343	12380.868	0	
Invoices Cust	Dispatch invoicing	Numeric	(0 - 3075)	37.308	69.277	0	
Costs Cust	Dispatch costs	Numeric	(-288.83 - 6087.24)	32.073	67.687	0	
Invoices Pick	Pickup invoicing	Numeric	(0 - 10000)	25.484	195.64	0	
Costs Pick	Pickup costs	Numeric	(-375 - 12297.7)	37.717	228.71	0	
Invoices Deliv	Delivery invoicing	Numeric	(0 - 5865)	4.38	119.187	0	
Costs Deliv	Delivery costs	Numeric	(-285.2 - 11643.12)	4.63	119.504	0	
Invoices Ins	Insurance invoicing	Numeric	(-1163.54 - 8135.47)	28.737	169.585	0	€
Costs Ins	Insurance costs	Numeric	(-729.04 - 3051.59)	16.148	86.729	0	
Invoices Orig	Invoicing of origin expenses. Expenses included per obligation at the place of origin.	Numeric	(-40 - 34622.8)	42.894	312.123	0	
Costs Orig	Expenses of origin. Expenses included per obligation at the place of origin.	Numeric	(-990.13 - 32228.1)	46.006	437.907	0	
Invoices Dest	Invoicing of destination expenses. Expenses included per obligation at the place of destination.	Numeric	(0 - 36690)	59.565	391.655	0	
Costs Dest	Destination expenses. Expenses included per obligation at the place of destination.	Numeric	(-1502.74 - 34912.5)	51.743	417.654	0	

Invoices Outr	Other invoiced revenue. Do not bill as "Freight", but they are mandatory in transportation (e.g.: Bill of Landing).	Numeric	(-3315.65 - 75649.5)	450.218	1643.341	0
Costs Outr	Other costs. Do not bill as "Freight", but they are mandatory in transportation (e.g.: Bill of Landing).	Numeric	(-1399.52 - 151918.05)	223.446	1345.742	0

APPENDIX C

- **Road inbound variables' characteristics after pre-processing.**

Feature Name	Description	Type	Range (min – max)	Mean	Standard deviation	Nº Of Missing Values	Measurement Unit
Quote Number	Number of the quote	Numeric	-	-	-		
Status Description	Satus description of the shipping (e.g. Invoiced, Delivered, Credit Note, etc.)	Categoric	-	-	-		
Service Level	Type of container transport (e.g. Full Truck load (FTL), Less than Truck load (LTL), etc.)	Categoric	-	-	-		-
Loading Country	Country where the loading was made	Categoric	-	-	-		
Unloading Country	Country where the unloading was made	Categoric	-	-	-		
Loading Date	Date when the loading was made	Date	-	-	-		
Goods Number	Quantity of goods	Numeric	(0 - 52)	6.582	14.978		unit
Gross Weight	Gross cargo weight	Numeric	(0 - 17002.19)	884.759	1781.812		Kg
Volume	This value corresponds to the higher value between the cubic meters and the loading meters.	Numeric	(0 - 27.816)	2.056	3.042		m^3
Cubic Meters	Calculated volume of the cargo	Numeric	(0 - 27.55)	1.459	2.963		
Loading Meteres	1 loading meter corresponds to 1 linear meter of cargo space on a truck	Numeric	(0 - 14.16)	0.787	1.803		m
Chargeable Weight	It is the weight proportional to the space (volume) that the goods occupy in truck, i.e. it is the weight that the customer actually pays for the transportation of the consignment or shipment	Numeric	(0.666 - 2478)	1821.985	3170.816	0	Kg
Seguro	Informs if the shipping has seguro or not	Categoric	-	-	-		
Despacho	Indicates the presence or not of dispatch	Categoric	-	-	-		-
Plan Characteristics	Characteristics of the shipping plan	Categoric	-	-	-		
Distance	Distance travelled	Numeric	(2 - 4017)	1779.681	721.799		Km
Revenue	Revenue with the shipping	Numeric	(3.780 - 2134)	274.687	309.506		€

Cost	Total costs with the shipping	Numeric	(0 - 2099.45)	252.446	313.174		€
Max Width	Maximum cargo width	Numeric	(0 - 3.06)	0.547	0.44		m
Max Height	Maximum cargo height	Numeric	(0 - 2.9)	0.722	0.714		
Max Length	Maximum cargo length	Numeric	(0 - 3.9)	0.85	0.722		
Cost Adr	Costs with dangerous cargo	Numeric	*				€
Cost Temp Control	Costs with temperature control	Numeric	*				
Cost Fuel	Costs with fuel	Numeric	(0 - 16.57)	0.945	2.125		
Cost Customs	Customs costs	Numeric	(0 - 40)	0.588	4.171		
Cost Seguro	Costs with insurance	Numeric	(0 - 42.96)	0.463	2.703		
Cost Other	Other costs	Numeric	(0 - 668.61)	58.332	77.225		
Sale Adr	Invoicing for dangerous cargo	Numeric	*				
Sale Temp Control	Invoicing for temperature control	Numeric	*				
Sale Fuel	Invoicing of fuel	Numeric	(0 - 48)	3.933	7.18		
Sale Customs	Invoicing of dispatch	Numeric	(0 - 75)	1.309	8.593		
Sale Seguro	Insurance invoicing	Numeric	(0 - 95)	1.773	8.59		
Sale Transport Price	Freight invoicing	Numeric	(0 - 2100)	252.333	301.861		
Sale Other	Other invoicings	Numeric	(0 - 388.5)	0.854	10.363		

* It was not registered any data regarding this variable.

APPENDIX D

- **Sea inbound variables' characteristics after pre-processing.**

Feature Name	Description	Type	Range (min – max)	Mean	Standard deviation	Nº Of Missing Values	Measurement Unit
Master	Shipping code	Categoric	-	-	-		
Biztype	Shipping process (air export, sea import, etc.)	Categoric	-	-	-		-
Shptype	Shipping type (house or direct)	Categoric	-	-	-		
Cwgt	Chargeable weight. It is the weight proportional to the space (volume) that the goods occupy in truck, i.e. it is the weight that the customer actually pays for the transportation of the consignment or shipment.	Numeric	(0 - 5048)	111.245	359.657		Kg
Teus	Twenty Foot Equivalent Unit. International standardised container type. A twenty foot unit measures about 6 meters.	Numeric	(0 - 8)	1.37	1.48		-
Houseinvoice	Invoice total. Total of [invoices_xxxxx] columns.	Numeric	(0 - 56730.19)	3012.852	4167.306		
Housecosts	Total costs. Total of the [costs_xxxxx] columns.	Numeric	(6.5 - 54204.09)	2553.057	3741.502		
Invoices Frete	Freight invoicing	Numeric	(0 - 56385)	2244.39	3715.468		
Costs Frete	Freight costs	Numeric	(0 - 52794.09)	2118.497	3601.135		
Invoices Cust	Dispatch invoicing	Numeric	(0 - 280)	48.995	60.43		
Costs Cust	Dispatch costs	Numeric	(0 - 300)	38.485	35.746		
Invoices Pick	Pickup invoicing	Numeric	(0 - 710)	18.312	72.533		
Costs Pick	Pickup costs	Numeric	(0 - 894.5)	24.736	78.725	0	
Invoices Deliv	Delivery invoicing	Numeric	(0 - 330)	0.439	9.035		
Costs Deliv	Delivery costs	Numeric	(0 - 390.41)	1.073	12.551		
Invoices Ins	Insurance invoicing	Numeric	(0- 752.67)	30.401	83.812		
Costs Ins	Insurance costs	Numeric	(0 - 392.88)	17.348	48.64		€

Invoices Orig	Invoicing of origin expenses. Expenses included per obligation at the place of origin.	Numeric	(0 - 1374.02)	51.753	172.104		
Costs Orig	Expenses of origin. Expenses included per obligation at the place of origin.	Numeric	(0 - 1955)	39.014	156.22		
Invoices Dest	Invoicing of destination expenses. Expenses included per obligation at the place of destination.	Numeric	(0 - 965.25)	63.428	151.902		
Costs Dest	Destination expenses. Expenses included per obligation at the place of destination.	Numeric	(0 - 1083.76)	33.432	119.812		
Invoices Outr	Other invoiced revenue. Do not bill as "Freight", but they are mandatory in transportation (e.g.: Bill of Landing).	Numeric	(0 - 7023)	555.134	817.017		
Costs Outr	Other costs. Do not bill as "Freight", but they are mandatory in transportation (e.g.: Bill of Landing).	Numeric	(0 - 3987)	280.471	400.909		