UNIVERSIDADE FEDERAL DO PARANÁ

JOÃO GABRIEL SANTIN BOTELHO

APPLICATIONS OF DATA-DRIVEN METHODS IN ENGINEERING: FROM MANUFACTURING SYSTEMS TO VEHICULAR CONNECTIVITY

CURITIBA

2025

JOÃO GABRIEL SANTIN BOTELHO

APPLICATIONS OF DATA-DRIVEN METHODS IN ENGINEERING: FROM MANUFACTURING SYSTEMS TO VEHICULAR CONNECTIVITY

Dissertação apresentada ao curso de Pós Graduação em Métodos Numéricos em Engenharia, Setor de Ciências Exatas, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Métodos Numéricos para Engenharia.

Orientador: Prof Dr. José Eduardo Pécora Junior
Coorientador: Prof Dr. Eduardo Alves Portela Santos

CURITIBA

2025

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Botelho, João Gabriel Santin
Applications of data-driven methods in engineering: from manufacturing systems to vehicular connectivity / João Gabriel Santin Botelho. – Curitiba, 2025.
1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Métodos Numéricos em Engenharia.

Orientador: José Eduardo Pécora Junior
Coorientador: Eduardo Alves Portela Santos

1. Sistemas inteligentes de veículos rodoviários. 2. Aprendizado do computador. 3. Processamento eletrônico de dados. 4. Teoria da predição. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Métodos Numéricos em Engenharia. III. Pécora Junior, José Eduardo. IV. Santos, Eduardo Alves Portela. V. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação MÉTODOS NUMÉRICOS EM ENGENHARIA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **JOÃO GABRIEL SANTIN BOTELHO**, intitulada: **APPLICATIONS OF DATA-DRIVEN METHODS IN ENGINEERING: FROM MANUFACTURING SYSTEMS TO VEHICULAR CONNECTIVITY**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 27 de Fevereiro de 2025.

Assinatura Eletrônica
02/04/2025 08:36:32.0
JOSÉ EDUARDO PÉCORA JUNIOR
Presidente da Banca Examinadora

Assinatura Eletrônica
02/04/2025 17:57:08.0
JOSÉ ROBERTO FREGA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
02/04/2025 12:34:18.0
ROBERTO ZANETTI FREIRE
Avaliador Externo (UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ)

Este trabalho é dedicado a todos que,
em alguma forma, me permitiram chegar até aonde estou.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais por me proporcionarem, desde muito cedo, acesso à educação de qualidade. E que, mesmo agora, continuam a me sempre me motivar. São eles, em grande parte, responsáveis por quem eu sou e onde consegui chegar.

Agradeço muitíssimo também ao meu orientador, professor Pécora, que me acompanhou e orientou durante os 2 anos do mestrado. O qual, mesmo com todos os seus compromissos e trabalhos, me auxiliou nessa jornada.

Igualmente agradeço ao meu coorientador, professor Portela, que me acompanhou durante o último ano do mestrado. O qual me deu a oportunidade de me tornar bolsista na Renault, o que deu luz a grande parte da pesquisa neste apresentado.

Agradeço também todos os professores do PPGMNE, os quais repassaram o conhecimento e embasamento necessário para a realização deste trabalho.

Não me esqueço de agradecer a Renault, a Fundação Araucária, e a todos os colegas de lá que me auxiliaram. Foi na Renault que tive a oportunidade de fazer grande parte da pesquisa neste apresentado.

E agradeço o professor Zanetti e o professor Frega que, como integrantes da banca, desempenharam um papel de extrema importância ao lerem e corrigirem esta dissertação.

De tanto ver triunfar as nulidades,
de tanto ver prosperar a desonra,
de tanto ver crescer a injustiça,
de tanto ver agigantarem-se os poderes nas mãos dos maus,
o homem chega a desanimar da virtude,
a rir-se da honra,
a ter vergonha de ser honesto.
(Rui Barbosa, 14 de dezembro de 1914)

RESUMO

Este trabalho tem como objetivo demonstrar as diferentes formas como a coleta e uso de dados podem gerar valor para diferentes áreas da engenharia. Apresentamos quatro artigos com aplicações de aprendizagem de máquina, mineração de processos, ciência e análise de dados em engenharia. Os artigos são divididos em duas áreas abrangentes: planejamento e controle de produção; e utilização de dados de veículos conectados. Os dois primeiros artigos focam no planejamento e controle de produções em manufaturas, com o primeiro sendo uma pesquisa inicial na previsão de tempo remanescente de ordens de produção e o segundo uma pesquisa mais robusta, a qual é continuação e melhora direta do primeiro. Nestes dois artigos, são apresentados modelos de predição de tempo remanescente de ordens de produção orientados ao produto, utilizando métodos de mineração de processo e aprendizagem de máquina. Os modelos foram testados em dados artificiais e em dados de uma manufatura real e apresentaram resultados interessantes. Os dois últimos artigos têm como foco a utilização de dados de veículos conectados para gerar valor em dois diferentes tópicos: eficiência energética e otimização no tamanho de baterias de veículos elétricos. No primeiro desses artigos, uma clusterização com base no contexto é apresentada como solução para tornar *rankings* de consumo de combustível mais justos, isto é, que comparem os motorístas com mínima influência externa. Neste artigo, tal método de clusterização é demonstrado com dados de veículos reais e também é demonstrada a influência do contexto no consumo de combustível. Utilizando essa clusterização, *rankings* justos são criados e outras aplicações são propostas. O segundo desses artigos explora, em parte, uma das aplicações propostas no artigo anterior para a clusterização com base no contexto. Neste artigo, são utilizadas técnicas de aprendizagem de máquina e ciência e análise de dados para otimizar o tamanho de baterias de veículos elétricos, considerando os perfis de viagem dos diferentes contextos e considerando diferentes hipóteses de recarregamento. Tamanhos ótimos de baterias são encontrados para diferentes perfis de motoristas.

Palavras-chaves: Veículos Conectados; Análise de Dados; Ciência de Dados; Aprendizagem de Máquina; Mineração de Processos; Predição de Tempo Remanescente.

ABSTRACT

This work aims to demonstrate the different ways in which the collection and use of data can generate value for different areas of engineering. We present four papers with applications of machine learning, process mining, data science and analysis in engineering. The papers are divided into two broad areas: production planning and control; and utilizing data from connected vehicles. The first two articles focus on production planning and control in manufacturing, with the first being initial research into predicting the remaining time of production orders and the second more robust research, which is a direct continuation and improvement of the first. These two articles present product-orientated models for predicting the remaining time of production orders using process mining and machine learning methods. The models were tested on artificial data and data from a real manufacturing plant and showed interesting results. The last two papers focus on using data from connected vehicles to generate value in two different topics: energy efficiency and optimizing the size of electric vehicles' batteries. In the first of these papers, context-based clustering is presented as a solution for making fuel consumption rankings fairer, i.e. comparing drivers with minimal external influence. In this paper, such a clustering method is demonstrated with real vehicles' data, and the influence of context on fuel consumption is also shown. Using this clustering, fair rankings are created, and further applications are proposed. The second of these papers partly explores one of the applications proposed in the previous paper for context-based clustering. In this paper, machine learning, data science and analysis techniques are used to optimize the size of electric vehicles' batteries, considering the travel profiles of different contexts and different recharging hypotheses. Optimal battery sizes are found for different driver profiles.

Key-words: Connected Vehicles; Data Analysis; Data Science; Machine Learning; Process Mining; Remaining Time Prediction.

SUMMARY

1	INTRODUCTION	10
2	REMAINING TIME PREDICTION IN MANUFACTURING SYSTEMS: AN APPROACH BASED ON ML AND PROCESS MINING	15
3	A PRODUCT-BASED HYBRID MODEL FOR REMAINING TIME PREDICTION OF PRODUCTION ORDERS: A PROCESS MINING AND MACHINE LEARNING APPROACH .	22
4	HOW CAN A CONTEXT-BASED CLUSTERING OF DRIVERS HELP INCREASE FUEL EFFICIENCY?	48
5	OPTIMIZING EV BATTERY SIZING WITH ICEV ENERGY CONSUMPTION AND CONTEXT-BASED CLUSTERING .	71
6	CONCLUSION	92
	REFERENCES	97

1 INTRODUCTION

Engineering is composed of many different knowledge areas. This comes from the definition of engineering, as it can be broadly defined as the use of natural science and mathematics to solve problems. The name engineering is a testament to its definition, coming from the Latin word *ingenium*, meaning “cleverness”, exactly what it takes to perform it.

For the constant problem-solving that is engineering, clever methods are constantly being envisioned, built, and tested. In the case of this work, we brought methods from mainly two fields of study: Machine Learning (ML) (Sarker, 2021) and Process Mining (PM) (Aalst, 2016). These methods can be inserted into the broad area of Data Science (Kelleher; Tierney, 2018) and Analysis (Kudyba, 2014), which uses different types of data to generate value, ranging from process optimization to the discovery of unknown behaviors. The field of ML can be defined as the use and development of statistical and mathematical algorithms that can learn from data and generalize to unseen data. Similarly, PM is a field that studies the processes that generate the data, extracting knowledge about its behavior.

In this work, we are going to demonstrate the usefulness of ML, PM, and Data Science and Analysis as a whole for two very different areas of engineering: production planning and control; and vehicular connectivity. The production planning and control area includes all the processes and methods involved in any part of a manufacturing process. Therefore, it can include a variety of problems, such as logistical planning of the production plant, control of the ongoing production, or analysis of the state of the production line. Vehicular connectivity is another broad area as it can include all that can be done with connected vehicles. However, as our focus here is vehicles with vehicle-to-cloud (V2C) connectivity, the applications are a little more constrained. They can include vehicle monitoring by either individual owners or fleet managers, analysis of vehicular usage by the manufacturer, or generation of training data for algorithms related to vehicular features.

The problem related to production planning and control we explore is the prediction of the remaining time of production orders in manufacturing systems. The remaining time of a production order is the amount of time until a production

order is finished given its current state. This information is of great importance for manufacturers and can be especially critical for process managers. In manufacturers, production orders are constantly being ordered, processed, or delivered. Thus, they require constant planning for production to run smoothly. For example, if a production manager knows the time in which a production order is going to be finished, it can schedule the logistics needed to deliver it to its client ahead of time and provide precise delivery dates. Furthermore, the remaining time information allows production managers to plan and prepare the next orders ahead of time, reducing idle time in the manufacturing.

The first two papers, in Chapters 2 and 3, present data-driven methods to predict the remaining time of production orders. To do such a task they make use of the logs of the processes that are normally generated by the production and kept by the manufacturers. These logs have a table-like structure and can be called event logs (Aalst, 2016) if they have basic information such as a production order identifier (called the case), the activity that was performed, and the time it occurred. From the event logs it is possible to extract the remaining time of each case and, with enough data, ML and PM models can be trained on it to predict the remaining time of new production orders. In our case, we consider manufacturers that produce different products on the production line. Thus, the models presented in the papers are product-oriented, e.g., each product has its own model, as different products can have different production times even if they pass throw the same activities.

In the first paper, five methods to predict the remaining time are presented and compared, which include two ML methods, a PM method, a hybrid method (Choueiri et al., 2020), and a baseline method. The hybrid method creates models that are a combination of many different models combined by optimizing a validation score. The baseline method is a statistical value of the remaining time of the training data. To compare and assess the models' performances they are tested on simulated logs. The simulated event logs are generated artificially by using preset production paths with different production times for each product and machine, and including different probabilities of rework in each machine. By generating logs for two different products with different probabilities of rework, we can analyze how the different methods behave in distinct situations.

In the second paper, we explore the same problem but with the addition of

two new ML methods and an improved baseline method. The baseline method is improved by transforming it into a simplified version of the PM method presented. Another addition in this paper is the usage of a different encoding. Encoding is the data processing technique used to make event log-type data understandable by ML methods. While in the first paper, a simple binary-type encoding was used, in this paper we present an encoding that accounts for the frequency of the activities. Additionally, the artificially generated logs are expanded, with new paths, machines, and products. Moreover, a deep analysis of a real manufacturer's log was done. We used this log to test the methods and compared the difference in performance of simulated logs with well-defined paths and real logs with unstructured paths.

The two first papers achieve their common objective of demonstrating the usefulness of the remaining time prediction in a manufacturing context. Their results also make clear the importance of product segregation when building the prediction models. The second paper, by having more models and test data, really shows the difference between the methods across different types of data.

Now we move from the production planning and control area to vehicular connectivity. The vehicular connectivity in this work is limited to V2C-type connectivity, e.g., vehicles with the capability of sending and receiving data from a server via the cellular network. This data can be miscellaneous, varying from simple information such as the vehicle's velocity to complex unstructured data such as images or videos. In the case of this work, the data is on the simpler side, consisting of information about the trip such as distance, time, speed, etc. Even though the information collected is not on the more complex side, there are a variety of applications that can use it. In our case, we explore two different topics: fuel efficiency improvement and size optimization of batteries for electric vehicles (EVs).

The importance of fuel efficiency in any context is clear, especially in recent years with the known influence of fossil fuels on global warming (Zecca; Chiari, 2010). Outside the environmental reasons, there also are economic reasons, as better fuel efficiency means lower expenses with fuel. The fuel efficiency can be improved by many different means, such as engineering more efficient motors and smarter transmissions or making the drivers themselves more efficient. In this work, we tackle this last, more indirect way, as the data provided by vehicular connectivity

allows us to understand how the vehicles are being driven. Our idea to indirectly improve fuel efficiency is to profit from a common human trace: competitiveness (Brankovic et al., 2018). To generate this sentiment between the drivers, a simple solution is to rank them based on the subject we want to improve, in the case of fuel efficiency, this means the construction of fuel efficiency rankings. If the drivers have access to their performance information and how they compare to other drivers, this can passively increase overall fuel efficiency as part of the drivers try to change how they drive to be better than the others. However, the constructions of these rankings are the real problem that needs to be solved as they must be as fair as possible. Given vehicles of the same model in similar conditions traveling the same route, what would mainly vary the fuel efficiency would be the different ways the drivers can drive the vehicles, precisely what ideal fuel efficiency rankings must classify. From the said characteristics that affect fuel efficiency, the vehicle's model is easily obtainable, leaving the problem we need to solve as the identification of where the vehicle was driven.

In the third paper in Chapter 4, a solution for the construction of fair fuel efficiency rankings is presented: a context-based clustering of the trips, built using data from millions of trips and thousands of vehicles from all around Europe. With context-based clustering it is possible to make rankings for each context, making the rankings fair by eliminating the external bias. To make this clustering the ML method k-means (MacQueen, 1967) is used. The k-means was chosen as the clustering method due to low computing time and good overall results, which were further improved using the k-means++ initiation. As our objective is context identification, context-related features such as the trips' distance, total time, average speed, and other speed-related features were used. The identified contexts can be directly related to different types of roads (Eppell et al., 2001), as they are strongly influenced by speed-related features. With the clustered trips, fair fuel efficiency rankings can be built, where the drivers' trips from different contexts are not compared. Furthermore, the constructed context-based clustering showed potential application in other areas outside making fuel efficiency rankings fair. One possible application in fleet management is demonstrated and applications in the improvement of recommendation systems and product development are suggested. Inside the product development area, one suggested application is related to EV battery sizing considering different trip profiles.

Choosing the right battery size for a new EV model is important for the same reason as fuel efficiency is important: global warming. Concerns with CO2 emissions have leveraged the production and sale of EVs all around the world in recent years (IEA, 2024). With this production growth, optimizing one of the most expensive parts of an EV (S&P, 2024) is even more important. In our case, the optimization of the battery size is done with one question as a north: which battery size would be ideal for drivers who are migrating from internal combustion engine vehicles (ICEVs) to EVs? This question is of utmost importance for manufacturers that want to sell vehicles for drivers with no EV experience who are used to ICEVs. To answer this question, our idea is to use data from real ICEVs, inferring their energy consumption if they were EVs and comparing how different battery sizes would perform in different contexts.

In the fourth and last paper in Chapter 5, ML, data science and analysis techniques are used to optimize the battery sizes using data from ICEVs. By using a function that converts average speed and temperature to energy consumption, we can analyze how energy is consumed and what battery sizes would allow for trips to be completed. Two hypotheses for vehicular recharging are compared, one where the vehicle is only charged at home, having just one full charge each day, and one where in-between trips fast charging was allowed for a limited amount of time. The context-based clustering from the third paper is used as a way to separate the different trip profiles, which influence energy usage and can lead to different battery sizes depending on the drivers' profile.

This work is divided into five parts, one for each paper and a conclusion: Chapter 2 with the paper "Remaining time prediction in manufacturing systems: an approach based on ML and process mining"; Chapter 3 with the paper "A product-based hybrid model for remaining time prediction of production orders: a process mining and machine learning approach"; Chapter 4 with the paper "How Can a Context-Based Clustering of Drivers Help Increase Fuel Efficiency?"; Chapter 5 with the paper "Optimizing EV Battery Sizing with ICEV Energy Consumption and Context-Based Clustering"; and the conclusion in Chapter 6. All four papers have individual "Related Works" sections with vast bibliographical reviews in each of their subjects.

2 REMAINING TIME PREDICTION IN MANUFACTURING SYSTEMS: AN APPROACH BASED ON ML AND PROCESS MINING

Given the data generated from logs of production orders in manufacturing systems, we wrote this paper to demonstrate the different ways such data could be used to generate value for the manufacturers.

Manufacturing systems deal with production orders that pass through specific processes to be produced. Such processes can have many steps and it can be difficult to know when the order is going to be finished in each step of its production. The research in this paper has the objective of demonstrating how data-driven methods can successfully predict the remaining time of production orders.

Using Process Mining (PM) techniques, the remaining time of previous production orders can be extracted from event logs. We present Machine Learning (ML) and PM methods that can be trained with this data and make predictions for future production orders. A hybrid method that creates optimized models from the output of other models is also proposed. All the methods are tested and compared to a baseline method on artificially generated logs. The logs are generated based on different preset paths and have a probability of activity rework, adding further complexity to the predictions.

This paper was presented at the 12th CIRP Global Web Conference (CIRPe 2024) in the area of Manufacturing Systems. It was published by the journal “Procedia CIRP” with DOI 10.1016/j.procir.2025.01.028 and can be accessed with this link.

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 132 (2025) 165–170

www.elsevier.com/locate/procedia

12th CIRP Global Web Conference (CIRPe 2024)

Remaining time prediction in manufacturing systems: an approach based on ML and process mining

João Gabriel Santin Botelho[a,1,*], Eduardo Alves Portela Santos[a], Alexandre Checoli Choueiri[a], José Eduardo Pécora Junior[a]

[a] *Universidade Federal do Paraná, Curitiba, Brazil*

* Corresponding author. *E-mail address:* joaobotelho@ufpr.br

Abstract

The remaining time prediction of production orders in the manufacturing domain is of major concern among production, planning, and control (PPC) managers. PPC managers must deal with significant uncertainty regarding the promise of delivering products to customers. Many techniques use data to predict the remaining time of production orders, such as neural networks, time series analysis, and non-parametric statistical models, among others. A powerful way to deal with these new machine-based data records is through process mining techniques, which can summarize and collect information about the underlying process based on event logs. This paper proposes a hybrid predictive model based on annotated transition-systems and machine learning models tailored to better predict ongoing production orders in industrial manufacturing environments. The linear combination of models is performed by optimizing a linear programming (LP) model that minimizes the combined absolute errors of predictions. We tested our new approach on artificially created logs. Results showed that our approach provides better accuracy measures than all the other tested methods for the test instances.

Keywords: Process Mining; Predictive process monitoring; Remaining time prediction; Manufacturing; Machine learning

1. Introduction

Recently, efforts have been concentrated on upgrading Production, Planning, and Control (PPC) systems following Industry 4.0 standards [1]. The advancements in hardware and software over recent years have enhanced the ability to generate and store large volumes of raw data on the shop floor [2]. This data proliferation has driven the development of innovative techniques for extracting valuable insights. In the realms of manufacturing and logistics, the emphasis on Industry 4.0, along with the concepts of smart manufacturing and smart supply chains, has underscored the need for machines to function as part of an integrated network and supply chains to operate seamlessly from end to end. This integration allows managers to leverage real-time data for more precise decision-making [3].

Within industrial contexts, process mining has emerged as a tool for performance evaluation, helping managers improve quality cycles, manage maintenance, and apply Six Sigma methodologies comprehensively, among other applications [4, 5]. The trend of using process mining to predict and monitor the remaining time in processes is gaining traction [6]. Predictive process monitoring encompasses techniques that utilize event logs to forecast the future state of business process executions [7].

As noted in [8], it is crucial to track the progress of production orders to reconcile differences between the planned schedule and the actual manufacturing process [9]. Predicting the remaining time for production orders enables managers to monitor deviations from the planned execution, facilitating real-time scheduling decisions. Ideally, the initial production plan would match the actual manufacturing process, but unexpected disturbances often cause deviations. Issues like machine tool breakdowns, the arrival of urgent orders, and large amounts of unqualified work-in-progress (WIP) items can significantly disrupt the original production schedule, causing major fluctuations in the remaining order time.

Predictive process monitoring is a research domain focused on exploring ongoing cases to predict future information. One

* Corresponding author. *E-mail address:* joaobotelho@ufpr.br

2212-8271
10.1016/j.procir.2025.01.028

of the earliest works in this area, proposed by [10], introduced a transition system annotated with timing information from ongoing cases. Later research enhanced this model by incorporating machine learning techniques such as Support Vector Machines (SVM) and Naive Bayes [11, 12]. Moreover, [12] included additional data, not just time, to enrich the transition system and the learning model. Other researchers [13, 14] proposed extending the annotated transition model with a context-based predictive clustering step, allowing for different predictors to characterize various contexts and scenarios.

The proposed hybrid method integrates two predictive approaches [7]: (1) an annotated transition system derived from event logs with operational timing information and (2) machine learning (ML) models. This hybrid approach combines predictions from both the annotated transition system and the ML models through a linear combination. The combination is optimized using a linear programming (LP) model that minimizes the combined absolute errors of the predictions.

The remainder of this paper is organized as follows: related works and process mining key concepts are presented in section 2. Section 3 presents the main ideas of the proposed framework for remaining-time predictions. In section 4, we present and analyse artificial logs. In section 5, discuss the results of the presented methods. Section 6 portrays the conclusion of the paper.

2. Foundations

In process mining terminology, an event is characterized by various attributes, e.g., an event has a time-stamp, a resource identifying the executor, associated costs, and so on [15]. Each event must be associated with a case. When all case events are in chronological order, we have a trace (a finite non-empty sequence of events, such that each event appears only once and time is non-decreasing). Note that it is possible to have various cases that follow the same trace, but each case is different. An event-log is a set of traces. In theory, any process with a time dimension could be stored as an event-log database, including manufacturing activities.

Table 1: An example of an event log.

Case	Act.	Start Time	Finish Time	Prod.	Qty
1	S	2023/01/16 10:00:00	2023/01/16 10:45:00	A	10
1	P1	2023/01/16 10:45:00	2023/01/16 12:00:00	A	10
1	P2	2023/01/16 12:00:00	2023/01/16 14:30:00	A	10
1	P3	2023/01/16 14:30:00	2023/01/16 16:00:00	A	10
1	E	2023/01/16 16:00:00	2023/01/16 16:30:00	A	10
2	S	2023/01/17 10:00:00	2023/01/17 10:30:00	A	6
2	P1	2023/01/17 10:30:00	2023/01/17 11:00:00	A	6
2	P3	2023/01/17 11:00:00	2023/01/17 12:15:00	A	6
2	P3	2023/01/17 12:15:00	2023/01/17 13:45:00	A	6
2	E	2023/01/17 13:45:00	2023/01/17 14:15:00	A	6

Table 1 presents an example of an event log related to a production system. The case id represents the production order, and the activity column (Act.) represents the activities (operations) performed by the machines. The two columns time represent the start and end of the activities. The last two columns, Prod. and Qty., are attributes related to the production order: the product being processed and the quantity, respectively.

3. Proposed Framework

Our model is composed of a linear combination of the predictions of three different models. One of those models is a transition system-based model (TSM), which relies heavily on process mining. The other two models are ML models, composed of a multiple linear regression model (MLR) and a random forest regression model (RF). The linear combination of all three models is done by optimizing an LP model, which minimizes the combined absolute errors of the predictions.

Our proposed framework collects data from a process as event logs. Those datasets are then treated, filtered, and transformed into forms each model can read. Then, those treated datasets are separated by product. TSM, MLR, and RF models are created for each product. Based on the testing predictions of the three models, the weights of the linear combination of the predictions for the hybrid models are decided by optimizing LP models. Finally, given an unfinished product with a partial trace, the hybrid model gives a predicted remaining time per product.

3.1. Transition System Based Model

A straightforward way to describe a transition system (TS) is that it represents all the paths in an event log. The TS has an initial state from which all the existing paths that are in the log branch. TSs are graphically represented as trees or graphs, in which states are represented by the nodes and the transition relations by the vertices.

We implemented an algorithm in Python to construct a TS from an event log. This algorithm builds a TS, creating all possible sub-traces considering the time order for each separated case. Consequently, the number of sub-traces produced equals the number of observations from the generating dataset since it allows repetition. This tolerance to repetitions will be essential in constructing the TSM, as it has annotations that can be different on equal sub-traces.

The transition system-based prediction model consists of attaching annotations [10], based on the target value, to all the different sub-traces in a TS and choosing a function that connects a partial trace to a prediction value, in this case, related to the remaining time.

Consider the following case example based on Table 1: running our algorithm for building TSs in this event log would give the sub-traces exposed in Figure 1. A good and straightforward way to calculate the possible remaining time, as one sub-trace can appear more than once, is to calculate its mean or median value. To account for processes that work in batches, we divide the remaining time by the quantity of products in the batch. In equation 1, we have an expression for the predicted remaining time per product of a sub-trace.

$$\text{Rem. Time}\,(Trace, n_{case}) = \frac{Trace_{case\_FT} - Trace_{act\_ST}}{n_{case}}. \quad (1)$$

In this remaining time per product equation (1), $Trace_{case\_FT}$ is the finish time of the whole case, e.g., for Case 1 in Table 1 it is "2023/01/16 16:30:00", $Trace_{act\_ST}$ is the start time of the activity from that case, and n_{case} is the number of products processed in that case.

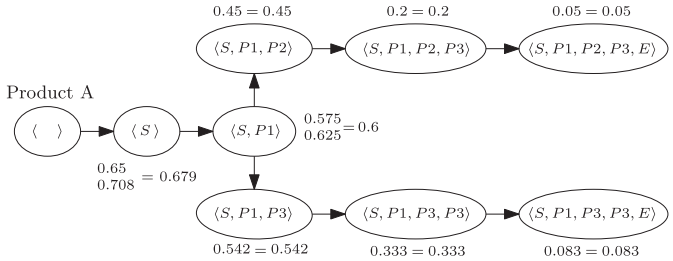

Fig. 1: The proposed annotated transition system for the event log in Table 1.

In Figure 1, we have the annotated transition system for the event log from Table 1, where for each sub-trace in the system, there is a mean remaining time per product.

As stated in [16], limitations of annotated transition system-based models appear when an input partial trace does not exist in the system. In those cases, a method for associating an unknown partial trace with a known one in the system is needed. Our chosen method for connecting partial traces to ones in the system is based on a similarity score.

Our score depends on the size of the partial trace or the size of the system sub-trace and the number of coinciding activities in the traces, which must be in the same position. To find an appropriate match to the analyzed partial trace in the annotated TS, the model searches all the system's sub-traces and gives each one a similarity score. The predicted remaining time per product of a partial trace is an average of all the annotations from the sub-traces with the best score. Algorithm 1 shows a pseudo-code for this connection function.

As seen in algorithm 1, the model cannot make a prediction if there is no match between an activity from the partial trace and one from any sub-trace in the system. This fact is an essential factor and disadvantage of TSMs. Unlike ML models, which make predictions based on features and can output predictions for whatever values its features have, transition system-based models, as presented here, need training data with a high degree of similarity to the data in which it will be used.

3.2. Machine Learning Regression Models

To diversify how a prediction can be made, we will present two ML methods that will be used together with the TSM to predict the remaining time per product in the hybrid model.

An event log is not an appropriate form to directly apply ML models, which usually need a set of variables from which it can derive its internal parameters' optimal values. Thus, some encoding is needed to extract information from an event log using an ML model.

Our encoding is from [16]. It is similar to the famous one-hot encoding, which gives binary values to categorical features to inform if the action or thing represented by that feature happened or not. The remaining time per product is used for the target variable. Applying this transformation to the log in Table 1, we obtain Table 2. This encoding does not account for activity repetition, which could be a problem for ML models.

Algorithm 1 An algorithm to associate a prediction value to a partial trace

Require: A transition system TS and a partial trace PT
Ensure: A prediction value $pred$
  $S_{max} \leftarrow 0$
  $match \leftarrow$ empty list
  for each $trace \in TS$ **do**
    $Size_{max}, Size_{min} \leftarrow \max(|PT|, |trace|), \min(|PT|, |trace|)$
    $S \leftarrow 0$
    for each $k \in (0 \leq k \leq Size_{min})$ **do**
      if $trace[k]$ is equal to $PT[k]$ **then**
        $S \leftarrow S + 1/Size_{max}$
      end if
    end for
    $match.add((S, trace))$
    if $S > S_{max}$ **then**
      $S_{max} \leftarrow S$
    end if
  end for
  $Mtraces \leftarrow$ all $trace$ in $match$ where S is equal to S_{max}
  $pred \leftarrow$ mean value from the annotations of each $trace$ in $Mtraces$

Table 2: Transformed event log from Table 1 using binary encoding.

S	P1	P2	P3	E	Rem. Time
1	0	0	0	0	0.65
1	1	0	0	0	0.575
1	1	1	0	0	0.45
1	1	1	1	0	0.2
1	1	1	1	1	0.05
1	0	0	0	0	0.7083
1	1	0	0	0	0.625
1	1	0	1	0	0.5417
1	1	0	1	0	0.3333
1	1	0	1	1	0.0833

Multiple Linear Regression: The MLR method estimates parameters that better describe some observations by applying a linear relation [17]. For their simplicity and the fact that many real applications have linear correlation, MLR models are the most commonly used data-fitting methods. This regression uses the relation $\boldsymbol{y} = \boldsymbol{X\theta} + \boldsymbol{\varepsilon}$. Given an observation vector $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$ with n observations and a matrix of variables $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ with p groups of variables, the MLR model tries to find the optimal vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^{n\times 1}$, which minimizes a vector of errors $\boldsymbol{\varepsilon} \in \mathbb{R}^{n\times 1}$.

The fair simplicity of the MLR method, while being an advantage, can also be seen as a disadvantage, as it has no hyper-parameters that could boost the method's performance.

Random Forest Regression: The RF method is a ML ensemble method [18]. An RF model is created by using the prediction of various decision trees, which are constructed from

random samples of the training dataset. The RF method is a way to reduce the variance of the overfitting-prone decision tree models. It does that by using the average predicted values of each decision tree as its predicted values.

The RF method can be considered a complex ML method based on the number of variations it can have. Some of its hyperparameters are the decision trees' depth, its maximum number of branches, its minimum number of leaves, its number of trees, the size of the tree training samples, or the variables in which a tree is trained. Hence, finding the optimal RF model for a dataset can be difficult. Nonetheless, all this complexity allows the construction of more robust and well-performing models.

3.3. Hybrid Model

Our Hybrid approach comes from [16]. It is a linear combination of the predictions of all the models, the TSM, MLR, and RF. The following expression gives the prediction of the hybrid model (HM):

$$\text{HM} = \alpha_1 \cdot TSM + \alpha_2 \cdot MLR + \alpha_3 \cdot RF. \tag{2}$$

The α_is are the weights of the combination; they follow the relation $\sum_{j=1}^{3} \alpha_j = 1$ and are non-negatives. Those weights are obtained by optimizing an LP model. The LP model finds coefficients that minimize the sum of the absolute errors from a test set.

Given the matrix M with the predictions of remaining time values, which has n lines, one for each sub-trace, and three columns, one for each model, and given the vector $\boldsymbol{o}$ containing the actual remaining time values of each sub-trace, the LP model is defined as:

$$\text{Min} \sum_{i=1}^{n} \varepsilon_i^+ + \varepsilon_i^- \tag{3}$$

s.t:

$$\sum_{j=1}^{3} \alpha_j \cdot M_{ij} + \varepsilon_i^+ - \varepsilon_i^- = o_i, \quad \forall i = 1, ..., n; \tag{4}$$

$$\sum_{j=1}^{3} \alpha_j = 1; \tag{5}$$

$$\alpha_j, \varepsilon_i^+, \varepsilon_i^- \geq 0, \quad \forall i = 1, ..., n, \forall j = 1, ..., 3. \tag{6}$$

The objective function in expression 3 minimizes the sum of the absolute predicted error, $|\varepsilon_i|$. The first set of constraints in equation 4 is an equality in which the weights and the errors must vary to make the weighted sum of the predictions, plus a positive or negative error, equal to the occurred value. The second constraint set in equation 5 is the already explained sum of the weights. The last constraint set in inequation 6 is the non-negativity constraint.

4. Case Study

In this section, we present, analyse, and test our methods on artificially created event logs. The artificial logs were created using the Python 3 programming language. They were generated considering the following parameters: there are two products, which differ in the machines they need; there are 11 different machines, which process time per product follow a normal distribution and differ depending on the product; the number of products of each batch varies from 10 to 100 uniformly; the number of products influences the total process time following a normal distribution; and there is a probability of process repetition.

Figure 2 shows the Petri net representation of the artificial logs. The silent transition τ illustrates the activity repetition process and is implicit in all transitions. In Table 3, we have the normal distribution of each machine's processing time per product for each product. The normal distribution for the variation of the total process time of the batch is $N(1, 0.05^2)$.

Table 3: Probability distribution of remaining time per product.

	Product	
Machine	1	2
M1	$N(0.3, 0.05^2)$	$N(0.25, 0.025^2)$
M2	$N(0.1375, 0.0175^2)$	-
M3	-	$N(0.225, 0.0125^2)$
M4	$N(0.25, 0.05^2)$	-
M5	$N(0.25, 0.025^2)$	-
M6	-	$N(0.275, 0.0375^2)$
M7	$N(0.2, 0.025^2)$	-
M8	$N(0.4, 0.025^2)$	-
M9	-	$N(0.325, 0.0125^2)$
M10	$N(0.35, 0.025^2)$	$N(0.175, 0.0125^2)$
M11	$N(0.15, 0.01^2)$	$N(0.1, 0.005^2)$

The probability of a case following a specific path was set as equal. The probability of process repetition refers to the chance of a product having to repeat the same process consecutively, as in the path $\langle M1, M1, M3, M6, M6, M9, M10, M11 \rangle$.

For our tests, we generated 5 logs, each with 680 cases. These logs differ in the probability of process repetition, with 0%, 5%, 10%, 15%, and 20% chance of process repetition. Because of this variation, they also differ in the number of events, with 4080, 4275, 4483, 4720, and 5112 events.

A total of 5 models were tested: the baseline model, where the predicted value is just the mean time per product of each product; the TSM; the MLR model; the RF model; and the HM. The training process was the same for all models: a 5-fold cross-validation with shuffling of the events. The shuffling minimizes the appearance of unique traces in the training or test sets.

The leading accuracy indicator chosen is the MAE, or mean absolute error, in equation 7, as it is the same, ignoring scale, as the one used in the LP model objective function 4. Another accuracy indicator used is the RMSE, or root mean square error, in equation 8, which emphasizes the outliers errors. The last one is the MAPE, or mean absolute percentage error, in equation 9, which is a percentage indicator of the MAE.

Table 4: Errors of the models for product 1 from the artificial logs.

Models	MAE					RMSE					MAPE				
Prob. Rep.	0%	5%	10%	15%	20%	0%	5%	10%	15%	20%	0%	5%	10%	15%	20%
Baseline	0.3854	0.4174	0.4424	0.4786	0.5219	0.4527	0.4970	0.5292	0.5756	0.6283	108.9106	113.1823	115.4856	120.0420	125.0817
TSM	0.0580	0.0925	0.1202	0.1576	0.2030	0.0831	0.1552	0.1943	0.2522	0.2946	6.5202	9.7907	12.6926	15.4323	19.8669
MLR	0.0701	0.1051	0.1304	0.1718	0.2143	0.0894	0.1555	**0.1882**	**0.2423**	**0.2865**	10.4190	13.5607	15.7941	18.5991	22.3014
RF	0.0584	0.0913	0.1159	0.1551	0.2025	**0.0828**	0.1532	0.1908	0.2522	0.2955	6.5404	**9.1470**	**11.0414**	**13.5457**	**17.1648**
HM	**0.0579**	**0.0909**	**0.1155**	**0.1542**	**0.1993**	0.0829	**0.1528**	0.1899	0.2489	0.2895	**6.5147**	9.1597	11.0840	13.8527	18.0079

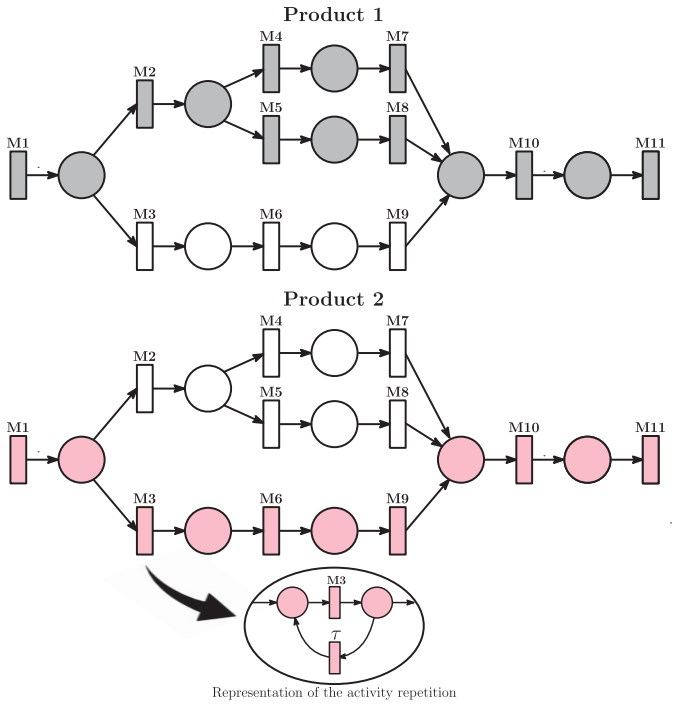

Fig. 2: Petri nets of the artificial logs process. Each color is a product path.

$$\text{MAE} = \frac{1}{n}\sum_{k=1}^{n}\left|\bar{y}_k - y_k\right| \tag{7}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(\bar{y}_k - y_k\right)^2} \tag{8}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{k=1}^{n}\left|\frac{\bar{y}_k - y_k}{\bar{y}_k}\right| \tag{9}$$

The models and logs implementations were done using Python 3. Libraries numpy, pandas, scikit-learn, pyomo, matplotlib, and seaborn were used.

We optimized the RF models' hyperparameters. The modified hyperparameters of the RF models were the number of trees and the maximum tree depth, and the model optimization was done by prioritizing the MAE score.

Table 4 has the mean MAE, RMSE, and MAPE from the cross-validation of each model for product 1 and each of the five repetition probabilities of the artificial logs. In Figure 3, we have graphs comparing the MAE and MAPE of all the models tested in all five artificial logs.

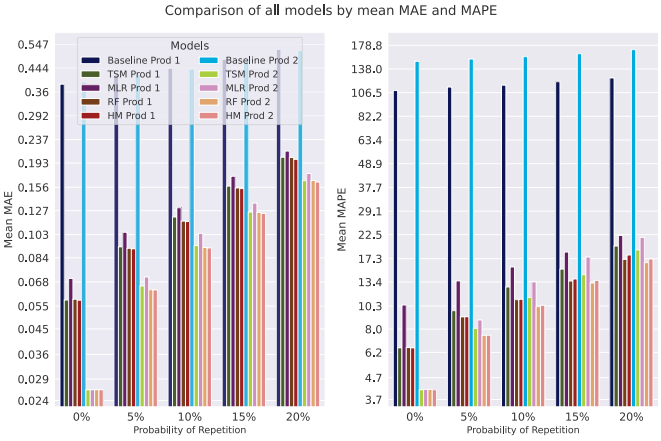

Fig. 3: Comparison of the MAE and MAPE for the two products.

As a 5-fold cross-validation was done, the hybrid models have coefficients for each fold. In Table 5, we have the mean coefficients of each hybrid model for each product and each artificial log.

Table 5: Mean coefficients of all hybrid models for the artificial logs tests.

Model	Prob. Rep.	Product 1			Product 2		
		TSM	MLR	RF	TSM	MLR	RF
HM	0%	0.845	0.016	0.140	0.787	0.069	0.144
HM	5%	0.257	0.029	0.714	0.064	0.054	0.882
HM	10%	0.113	0.034	0.853	0.204	0.035	0.762
HM	15%	0.250	0.053	0.697	0.380	0.057	0.563
HM	20%	0.439	0.060	0.501	0.387	0.038	0.575

5. Results and Discussions

Analysing the results of the artificial logs tests from Table 4 and Figure 3, it is noticeable that all models show better performance by a large margin than the baseline. However, as the probability of activity repetition increases, the performance of all models decreases. A reason for this behavior is the increase of possible activity paths that the activity repetition enables, thus increasing the variability in remaining time values, i.e., increasing the inherent randomness of the data.

For the MAE values of the artificial logs tests, the overall best model is the hybrid, which showed the best results in all instances. As the hybrid models are optimized based on the MAE, this supremacy on MAE values does not necessarily translate to better RMSE or MAPE values.

Analysing the RMSE, the MLR, the simplest model used, ignoring the baseline, showed the best performance. This good performance of the MLR model is not a surprise, given that the artificial logs follow well-behaved paths with normally distributed remaining time values. Also, better results on a less punitive metric on general models are expected, as the RMSE punishes more severely outliers than the MAE.

Looking at the MAPE values, the RF model is always the best model when there is activity repetition. Also, this model is the second-best model after the hybrids on MAE values.

When analysing the MAE values and the hybrid models' coefficients in Table 5, it is noticeable that the coefficients follow a particular logical pattern. For all instances, models that perform better have a higher coefficient value. As the coefficients follow this pattern, what we have in Table 5 is that, on average, the models that make up more of the hybrid models are mainly the TSM and RF models, with the MLR exerting little to no influence.

Overall, the artificial tests results showed that all models performed much better than the baseline, behaving very similarly and performing well on moderately well-distributed data and well-behaved paths.

6. Conclusion

This paper presented remaining time prediction methods based on process mining, machine learning, and a hybrid approach. The presented models are "product-oriented" and capable of coping with manufacture particularities, in which traces are represented as the activities already performed in the process, and a prediction of an incomplete trace is performed.

The framework presented deals with batch-type manufacturing. Furthermore, the presented TSM introduces a similarity score that copes with a transition system's "no trace" limitation. The two ML methods presented enable us to introduce the hybrid approach by combining them with the TSM.

We tested and validated the presented methods and approach using artificially created logs. Our tests of the models showed excellent results, with all models performing better than the baseline in all instances.

Considering the TSM, its ease of use appeals to manufacturing professionals. Since it does not require any specific process mining software, the learning curve to use it is significantly reduced, and there are no associated costs. Our prediction method, built on ML models, can provide excellent reliability to production and planning control managers. However, a challenge that arises is how to generate confidence in these models for PPC managers. Consequently, we are investigating the enrichment of our system to explain their reasoning processes and outputs (e.g., predictions) automatically. Also, we plan to perform future tests on real-world event logs and further investigate how the increase in the probability of repetition decreases the performance of our models.

Author Contributions

J.G.S. Botelho: Conceptualization, Methodology, Implementation, Writing, Visualization. E.A.P. Santos: Conceptualization, Methodology, Writing, Supervision. A.C. Choueiri and J.E.J. Pécora: Reviewing.

References

[1] H. Cañas, J. Mula, F. Campuzano-Bolar ñın, R. Poler, 2022. A conceptual framework for smart production planning and control in industry 4.0, Computers & Industrial Engineering 173.
[2] Y. Cheng, K. Chen, H. Sun, Y. Zhang, F. Tao, 2018. Data and knowledge mining with big data towards smart production, Journal of Industrial Information Integration 9, 1–13.
[3] C. L. Garay-Rondero, J. L. Martinez-Flores, N. R. Smith, S. O. C. Morales, A. Aldrette-Malacara, 2020. Digital supply chain model in industry 4.0, Journal of Manufacturing Technology Management 31, 887–933.
[4] C. dos Santos Garcia, A. Meincheim, E. R. F. Junior, M. R. Dallagassa, D. M. V. Sato, D. R. Carvalho, E. A. P. Santos, E. E. Scalabrin, 2019. Process mining techniques and applications–a systematic mapping study, Expert Systems with Applications 133, 260–295.
[5] R. Lorenz, J. Senoner, W. Sihn, T. Netland, 2021. Using process mining to improve productivity in make-to-stock manufacturing, International Journal of Production Research 59, 4869–4880.
[6] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, I. Teinemaa, 20219. Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring, ACM Transactions on Intelligent Systems and Technology (TIST) 10, 1–34.
[7] W. Rizzi, C. Di Francescomarino, F. M. Maggi, 2020. Explainability in predictive process monitoring: when understanding helps improving, in: International Conference on Business Process Management, Springer, 141–158.
[8] W. Fang, Y. Guo, W. Liao, K. Ramani, S. Huang, 2020. Big data driven jobs remaining time prediction in discrete manufacturing system: a deep learning-based approach, International Journal of Production Research 58, 2751–2766.
[9] J. C. Serrano-Ruiz, J. Mula, R. Poler, 2024. Job shop smart manufacturing scheduling by deep reinforcement learning, Journal of Industrial Information Integration.
[10] W. Aalst, van der, M. Schonenberg, M. Song, 2011. Time prediction based on process mining, Information Systems 36, 450–475.
[11] M. Polato, A. Sperduti, A. Burattin, M. de Leoni, 2014. Data-aware remaining time prediction of business process instances, IJCNN 2014, IEEE, 816–823.
[12] M. Polato, A. Sperduti, A. Burattin, M. d. Leoni, 2018. Time and activity sequence prediction of business process instances, Computing 100, 1005–1031.
[13] F. Folino, M. Guarascio, L. Pontieri, 2013. Discovering high-level performance models for ticket resolution processes: (short paper), OTM 2013 Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013. Proceedings, Springer, 275–282.
[14] F. Folino, M. Guarascio, L. Pontieri, 2014. Mining predictive process models out of low-level multidimensional logs, CAiSE 2014. Proceedings 26, Springer, 533–547.
[15] W. Aalst, van der, 2016. Data science in action, Springer.
[16] A. C. Choueiri, D. M. V. Sato, E. E. Scalabrin, E. A. P. Santos, 2020. An extended model for remaining time prediction in manufacturing systems using process mining, Journal of Manufacturing Systems 56, 188–201.
[17] M. Tranmer, J. Murphy, M. Elliot, M. Pampaka, 2001. Multiple linear regression (2nd edition), Cathie Marsh Institute Working Pa- per 2020-01.
[18] L. Breiman, 2001. Random forests, Kluwer Academic Publishers.

3 A PRODUCT-BASED HYBRID MODEL FOR REMAINING TIME PREDICTION OF PRODUCTION ORDERS: A PROCESS MINING AND MACHINE LEARNING APPROACH

The objective of this research remains the same as the previous paper, accurate prediction of the remaining time of production orders, but with further research, new models, and tests in different contexts. This paper has a much more robust literature review and a more in-depth explanation of the subject, techniques used and methods proposed.

The new models consist of two additional ML models and an improved baseline model. As the baseline model in the previous paper was too simple, the new proposed model has a little more complexity, being a simplified version of the presented process mining (PM) method.

The same logic as the previous paper to generate the artificial logs was used but with more products and expanded paths. An important addition to this paper is the use of event logs from real manufacturers. These real logs allow us to compare how different models perform in different contexts.

This paper was submitted to the journal “Engineering Applications of Artificial Intelligence” and, until the writing of this, is under review.

A product-based hybrid model for remaining time prediction of production orders: a process mining and machine learning approach

João Gabriel Santin Botelho[a,*], Eduardo Alves Portela Santos[a], José Eduardo Pécora Junior[a], Alexandre Checoli Choueiri[b]

[a]*Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE), Universidade Federal do Paraná, Evaristo F. Ferreira da Costa 408, Curitiba, Paraná, 81530-015, Brazil*
[b]*Departamento de Engenharia de Produção, Universidade Federal do Paraná, Curitiba, Paraná, 81530-015, Brazil*

Abstract

The remaining time prediction of production orders in the manufacturing domain is of major concern among production, planning, and control (PPC) managers. PPC managers must deal with significant uncertainty regarding the promise of delivering products to customers. Many techniques use data to predict the remaining time of production orders, such as neural networks, time series analysis, and non-parametric statistical models, among others. A powerful way to deal with these new machine-based data records is through process mining techniques, which can summarize and collect information about the underlying process based on event logs. This paper proposes a hybrid predictive model based on annotated transition-systems and machine learning models tailored to better predict ongoing production orders in industrial manufacturing environments. The linear combination of models is performed by optimizing a linear programming (LP) model that minimizes the combined absolute errors of predictions. We tested our new approach on artificial logs and a log from an actual manufacturer. Results showed that our approach provides better accuracy measures than all the tested methods for test instances.

Keywords: Process Mining, Predictive process monitoring, Remaining time prediction, Manufacturing, Machine learning

1. Introduction

According to [1], Manufacturing is one of the most challenging domains for business process management. There is an increasing need to reduce product launch and delivery times, optimize manufacturing operations and resources, reduce costs, and increase product quality. Extremely rigorous performance requirements and the demand for ultra-fast deliveries generate a search for continuous process improvement never seen before in the manufacturing industry. In this context, production, planning, and control (PPC) stands out as a company's strategic area where these requirements have to be achieved.

PPC systems are information systems (IS) designed to assist managers in decision-making [2]. These tools support all activities that define what, how much, and when to produce, purchase, and deliver efficiently to satisfy customer needs. Furthermore, they define the production sequence for each product. Traditional production scheduling defines which product will be produced when and where on a shop floor. However, several uncertainties can occur on the shop floor, affecting the initial production plan, such as stochastic processing times, machine breakdowns, lack of raw materials, and long times in machine setups, among others [3, 4]. In this context, production managers have to deal with significant uncertainty regarding the promise of delivering products to customers.

According to [5], usually financial criteria are considered for decision-making in production planning and control. With the advancement of Industry 4.0, which increases efficiency, value creation, and real-time optimization using technologies such as the Internet of Things (IoT), companies are seeking decision-making criteria focused on the

*Corresponding author
Email address: joaobotelho@ufpr.br (João Gabriel Santin Botelho)

customer experience. The idea is to improve customer service levels rather than just minimizing costs. Therefore, assuming that unforeseen events and uncertainties will occur, estimating more precise delivery dates for products to customers is extremely important.

Recently, efforts have been made to transform PPC systems towards Industry 4.0 [2]. Hardware and software developments in recent years have increased the ability to generate and store information, leading to a massive volume of raw data on the shop floor [4]. This data availability has sparked interest in developing new techniques capable of extracting increasingly helpful information from them. In manufacturing and logistics environments, this is reinforced by the pursuit of Industry 4.0 and the concepts of smart manufacturing and smart supply chain, in which machines work integrated as a collaborative network and the supply chain is integrated from end to end in all their operations, providing managers with the possibility of using online data to make more accurate decisions [6]. In this context, data analysis in the industry has grown extraordinarily [7, 8]. Furthermore, according to [7], recent advances in Big Data and Machine Learning (ML) have transformed traditional manufacturing towards a new digital transformation era. [9] point out the massive data collected in the manufacturing process has the characteristics of multi-dimensional, heterogeneous, and time series.

There are many techniques that use data to predict variables of interest in some processes, for example, neural networks, time series analysis, and non-parametric statistical models, among others [10]. A powerful way to deal with these new machine-based data records is through process mining (PM) techniques, which can summarize and collect information about the underlying process. The main concern in process mining is process discovery based on event logs [11]. With this information, discovery algorithms are run to extract a process model. More information can be analyzed with the model at hand; for example, compliance with the predefined model can be checked, bottleneck activities are more easily detected and addressed, and some future process extensions and inferences [12].

In the industrial context, process mining has been applied as a performance evaluation method, guiding managers in the quality improvement cycle, in maintenance management, in the integrated use of the Six Sigma methodology, among other applications [13, 14, 15]. The use of process mining to predict and monitor remaining times in processes is a current trend [16]. Predictive process monitoring is a family of techniques that uses event logs to predict the state of a business process execution [17]. For example, a predictive monitoring technique may seek to predict the remaining execution time of each ongoing case of a process [16], the next activity that will be executed in each case [18], or the outcome of a case in relation to a set of possible business outcomes [19].

According to [20], it becomes fundamental to monitor the progress of production orders to balance the difference between the original schedule and the actual manufacturing process [21]. In this way, the remaining time prediction of production orders allows the manager to monitor deviations in the execution of the actual production plan, thus enabling real-time decision-making related to scheduling. Ideally, the original production plan would be consistent with the actual manufacturing process. However, this is different from what usually happens due to unexpected disturbances in production. Breakdown of machine tools, the arrival of urgent orders, and large volumes of unqualified WIPs that disrupt the original production schedule will result in large fluctuations in the order's remaining time. Also, deviations in the time between the arrival of production orders, processing, and preparation time can affect the original schedule. In this case, these deviations change the production plan gradually and cause a slight fluctuation in the remaining time. However, since these deviations accumulate over time, rescheduling is likely unavoidable.

Despite numerous works in the predictive monitoring area, there are still several gaps with potential for applying new methods. This paper aims to address some of these gaps: high frequency of repetitive manufacturing tasks on machines, the same product with different production sequences, redundant machines, and high process variability (high number of traces or production order sequences). To achieve that, this paper analyzes two types of logs, one artificially made and another from a real production system with the mentioned characteristics. To address this complexity, this article proposes a hybrid method for predicting the remaining time of production orders based on integrating process mining with machine learning and optimization models. The work assumes that primary data will be available and be used for process mining and forecasting: the production order ID, manufacturing tasks (activities), machine IDs, and timestamps (start and end of activities). This dataset, called event log, is the raw material for process mining application [11].

The proposed hybrid method simultaneously encompasses two approaches used in prediction [17]: (1) annotated transition system discovered from the event log with operation timing information; (2) machine learning models. The hybrid method is implemented by the linear combination of predictions from annotated transition and machine learning models. The linear combination of models is performed by optimizing a linear programming (LP) model that

minimizes the combined absolute errors of predictions.

The proposed method aims to enhance the accuracy of predicting the remaining time of production orders and offers significant advantages for managers and decision-makers in manufacturing environments. By utilizing process mining techniques tailored to the specific characteristics of manufacturing processes, managers can achieve more precise estimates of when production orders will be completed. This improved accuracy enables better planning and allocation of resources, such as labor, materials, and equipment, reducing idle time and increasing overall operational efficiency.

The remainder of this paper is organized as follows: related works and process mining key concepts are presented in Section 2. Section 3 presents the main ideas of the proposed framework for remaining-time predictions. In Section 4, we present and analyze the artificial logs and the real industrial log. Section 5 shows and discusses the results of the methods applied to the logs. Section 6 portrays the conclusion of the paper and some remarks about the computational analysis of the logs.

2. Foundation

2.1. Preliminaries in Process Mining

In process mining terminology, an event is characterized by various attributes, e.g., an event has a timestamp, a resource identifying the executor, associated costs, and so on [11]. Each event must be associated with a case. When all case events are in chronological order, we have a trace (a finite non-empty sequence of events, such that each event appears only once and time is non-decreasing). Note that it is possible to have various cases that follow the same trace, but each case is different. An event log is a set of traces. In theory, any process with a time dimension could be stored as an event-log database, including manufacturing activities. Several works formalize the main elements of process mining. The reader is invited to see [11] and [22].

Case ID	Activity	Resource	Start Timestamp	Complete Timestamp	Span	Order Qty	Part Desc.	Worker ID
Case 1	Turning & Milling - Machine 4	Machine 4 - Turning & Milling	2012/01/30 06:59:00.000	2012/01/30 07:21:00.000	000:22	10	Cable Head	ID4167
Case 1	Turning & Milling - Machine 4	Machine 4 - Turning & Milling	2012/01/30 07:21:00.000	2012/01/30 10:58:00.000	003:37	10	Cable Head	ID4167
Case 1	Turning & Milling Q.C.	Quality Check 1	2012/01/31 13:20:00.000	2012/01/31 14:50:00.000	001:30	10	Cable Head	ID4163
Case 1	Laser Marking - Machine 7	Machine 7- Laser Marking	2012/02/01 08:18:00.000	2012/02/01 08:27:00.000	000:09	10	Cable Head	ID0998
Case 1	Lapping - Machine 1	Machine 1 - Lapping	2012/02/14 00:00:00.000	2012/02/14 01:15:00.000	000:00	10	Cable Head	ID4882
Case 1	Lapping - Machine 1	Machine 1 - Lapping	2012/02/14 00:00:00.000	2012/02/14 01:15:00.000	000:00	10	Cable Head	ID4882
Case 1	Lapping - Machine 1	Machine 1 - Lapping	2012/02/14 09:05:00.000	2012/02/14 10:20:00.000	000:00	10	Cable Head	ID4882
Case 1	Lapping - Machine 1	Machine 1 - Lapping	2012/02/14 09:05:00.000	2012/02/14 09:38:00.000	000:33	10	Cable Head	ID4882
Case 1	Round Grinding - Machine 3	Machine 3 - Round Grinding	2012/02/14 09:13:00.000	2012/02/14 13:37:00.000	004:24	10	Cable Head	ID4445
Case 1	Round Grinding - Machine 3	Machine 3 - Round Grinding	2012/02/14 13:37:00.000	2012/02/14 15:27:00.000	001:50	10	Cable Head	ID4445
Case 1	Final Inspection Q.C.	Quality Check 1	2012/02/16 06:59:00.000	2012/02/16 07:59:00.000	001:00	10	Cable Head	ID4493
Case 1	Final Inspection Q.C.	Quality Check 1	2012/02/16 12:11:00.000	2012/02/16 16:12:00.000	004:01	10	Cable Head	ID4493
Case 1	Final Inspection Q.C.	Quality Check 1	2012/02/16 12:43:00.000	2012/02/16 13:58:00.000	000:00	10	Cable Head	ID4493
Case 1	Packing	Packing	2012/02/17 00:00:00.000	2012/02/17 01:00:00.000	000:00	10	Cable Head	ID4820
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/17 07:01:00.000	2012/01/17 11:05:00.000	004:04	251	Spur Gear	ID3846
Case 10	Turning Q.C.	Quality Check 1	2012/01/17 11:00:00.000	2012/01/17 11:15:00.000	000:15	251	Spur Gear	ID4618
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/17 19:24:00.000	2012/01/17 20:01:00.000	000:37	251	Spur Gear	ID4132
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/17 20:01:00.000	2012/01/17 23:43:00.000	003:42	251	Spur Gear	ID4132
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/17 23:49:00.000	2012/01/18 06:32:00.000	006:43	251	Spur Gear	ID4794
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/18 06:59:00.000	2012/01/18 07:24:00.000	000:25	251	Spur Gear	ID3846
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/18 16:33:00.000	2012/01/18 17:55:00.000	001:22	251	Spur Gear	ID4132
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/18 17:57:00.000	2012/01/18 20:04:00.000	002:07	251	Spur Gear	ID4132
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/18 20:10:00.000	2012/01/19 06:29:00.000	010:19	251	Spur Gear	ID4794
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/19 16:12:00.000	2012/01/19 18:09:00.000	001:57	251	Spur Gear	ID4132
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/19 18:10:00.000	2012/01/19 20:08:00.000	001:58	251	Spur Gear	ID4132
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/19 20:09:00.000	2012/01/20 05:10:00.000	009:01	251	Spur Gear	ID4794
Case 10	Turning & Milling - Machine 9	Machine 9 - Turning & Milling	2012/01/22 02:42:00.000	2012/01/22 06:34:00.000	003:52	251	Spur Gear	ID4641

Table 1: An example of an industrial manufacturing event log.

Table 1 shows a fragment of an event log related to an industrial manufacturing system (available at: data.4tu.nl). The first column of the Table shows the case ID, representing the production order ID. In manufacturing, all activities performed on a given product can be stored in a Production Order (PO). Thus, in this event log, a PO is considered as a case for process mining application. The second column specifies the activity, in this case, an activity along with its performed machine, considering that one machine can perform more than one activity. The third column specifies the resource responsible for each activity. The fourth and fifth columns contain the activity's start and complete timestamps. The remaining columns specify the attributes of each case ID and activities. For example, the column

"Work Order Qty" represents the total of parts processed in an activity by a specific machine. The column "Part Desc" identifies the products processed by machines (in this segment, two products are processed: cable head and spur gear). When we consider a specific case, it is possible to identify a sequence of activities. For example, case 1 (or production order 1) has a sequence of 14 activities. In this example, there are some repetitions of activities.

In the industrial manufacturing domain, we use the terms activities and machines interchangeably for convenience. We may use (M_1, M_2, M_3) to denote a sequence of three manufacturing activities executed by machines M_1, M_2 and M_3, respectively. Also, a unique machine can execute n manufacturing activities, such as drilling, turning, and milling. Thus, we define M_{mk} as a set of m (m=positive integer) manufacturing machines and $k \in K$, where K is a finite set of manufacturing operations. Note that depending on the project objective, it is not necessary to identify the operation performed by a specific machine. Therefore, in the event log, the M_i label is sufficient for a process mining project.

2.2. *Related Works*

Predictive process monitoring is an area of research that explores ongoing cases to predict future information. These can be activities, remaining time, or case results [23, 17]. The basic idea of remaining time prediction is illustrated in Figure 1. According to [24], the predictive monitoring literature can be classified into the prediction and algorithm types dimensions. In the prediction type dimension, one can also identify a subcategory called predictions related to numeric predictions. This subcategory includes the prediction of the remaining time of an ongoing execution. Another dimension reported by [24] is the approach to building the prediction model. In this case, two approaches are identified: those based on explicit models, such as annotated transition systems, generally obtained from the event log, or those that use statistical or machine learning models, such as regression models, classification models, or neural networks. In this case, the event log extracts attributes that are input to the learning models.

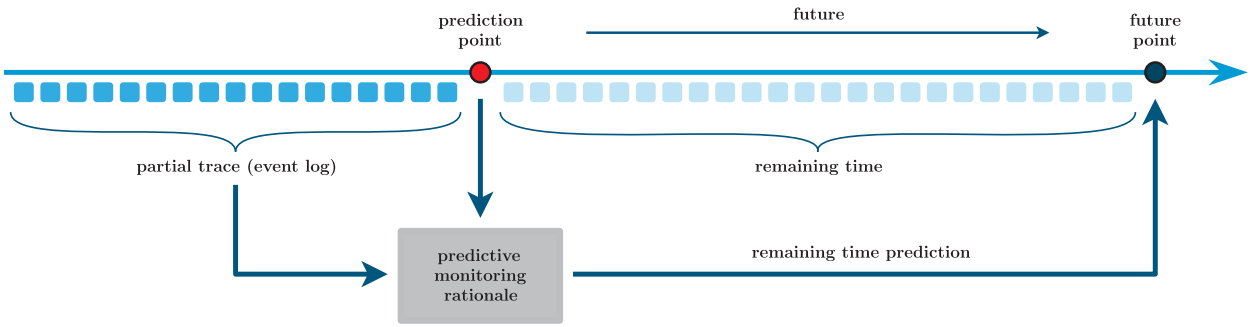

Figure 1: Exemplification of remaining time monitoring and predicting.

Considering the prediction of remaining time, one of the first works in the area is the one proposed by [25]. The authors propose a transition system annotated with timing information extracted from the event log. The model is then used to predict the remaining time for an ongoing case. Later work extended the annotated transition system model by integrating machine learning techniques, e.g., SVM and Naive Bayes [26, 27]. Additionally, [27] also uses data, in addition to time, to enrich the transition system and the learning model. [28, 29] also propose to extend the annotated transition model by combining a context-based predictive clustering step. The idea is that different predictors can characterize different contexts and scenarios.

[30] uses sequence trees model to predict remaining time and next future activity. The model allows the clusterization of traces with similar sequences of activities. A prediction model is then built for each node in the sequence tree. [31] use generally distributed transitions stochastic Petri nets (GDT-SPN) to predict the remaining execution time of a process instance. [32] extend previous work by exploring the time elapsed since the last event to obtain more accurate predictions. [33] use Hidden Markov Models (HMM) to predict remaining time. A comparative evaluation shows that HMM provides more accurate results than annotated transition systems and regression models. [34] introduce cross-case feature predictions to predict the completion time of an ongoing case. The proposed approaches leverage not only information related to the ongoing case but also the status of other cases (e.g., the number of concurrent cases) to make predictions.

[35] presents a framework capable of correlating process characteristics. They enrich the log with derived information and then discover correlations using decision trees. One of the proposed correlations is the prediction of completion times. Predicting future events in process mining is proposed in [36]. The authors use a sliding window model to transform the event-driven database into a predictive clustering problem, where the target variables are chosen as the characteristics of the next event in the case. [10] propose the same type of prediction. The authors use a long- and short-term neural network to predict the next event in the log. Along with the next event, the neural network can deal with several attributes of the event, one of them being the completion time.

A general model is proposed in [37], which aims to answer several questions about a process, in addition to estimates of completion time: estimates for discarding products in a machine, whether deviations from the regulatory process cause delays in cases carried out due to non-compliance. The approach combines elements of data and process mining techniques. [38] argues that most predictive methods for remaining time in business processes are considered "black boxes" as they predict a single scalar value without decomposing it into elementary components. The authors extract a process model from the log and replay the current case against it to see its current state. A flow analysis is then performed to predict the time remaining for completion. [16] researches business process forecasting methods. They present a review of the literature, as well as a cross-benchmark of 16 methods based on 16 real-life data sets.

Several works propose models for predicting remaining time in the industrial and manufacturing context. In this case, the term jobs remaining time (JRT) represents the remaining time required to complete the remaining jobs in the order, which is an instantaneous prediction under real-time conditions [20]. In manufacturing, several factors can affect the cycle time of a product [39, 40]: work-in-progress (WIP) levels, line-bottlenecks, rework rates, equipment downtime, mean time between failures/to repair, equipment load, product mix. These factors can cause significant variations in the original production plan and, consequently, in the job's remaining time.

Another aspect pointed out by [20] is the trend toward customization of complex products in the industry, which has led to a change in production mode from make-to-stock (MTS) to make-to-order (MTO). In this scenario, predicting the remaining production time is crucial to meet customer expectations. Works proposing methods for predicting promised time [41, 42], order completion time [43, 44], cycle time [45, 46, 47, 48], remaining time [40, 20] are very useful for supporting decision-making in production planning and control.

Digital twins (DT) have emerged as technologies for all-encompassing tools, including monitoring, simulation, optimization, and prediction. According to [49], advancements in machine learning, the Internet of Things (IoT), and big data have significantly improved DT features such as real-time monitoring and accurate forecasting. Other works deal with relevant manufacturing problems that affects remaining time of production orders, as bottlenecks and its cause analysis. For example, [50] propose an approach that utilises fusion-based clustering and hyperbolic neural network-based knowledge graph embedding for bottleneck identification and root cause analysis.

The studies referenced above lack a defined framework that facilitates the integration of predictive methods with an information system and the ever-expanding manufacturing databases. Process mining techniques effectively address this gap, as these techniques inherently rely on information systems to generate their logs. Utilizing process mining to gather and interpret data within an industrial context is a logical step, and extending this approach to make predictions is a natural progression. This paper presents an alternative to traditional cycle-time prediction methods, which leverages process mining and utilizes raw data as input.

By employing process mining, new data can be seamlessly collected and integrated into the model, enabling an online prediction system capable of adapting to changes on the production floor. Our model estimates the remaining time of a process instance in a manufacturing environment, effectively serving as a prediction for the remaining cycle time of a product. Also, we observed that only a few of the methods mentioned in this section address the online aspect of prediction, and those that do primarily focus on business processes. Although manufacturing processes can be broadly considered within the same framework as business processes, their unique characteristics can be leveraged to overcome some of the limitations of generic process mining algorithms, resulting in more accurate predictions.

Our paper emphasizes the importance of regularly updating the predictive model to reflect current system characteristics and adapt to changes in the manufacturing process, ensuring ongoing accuracy. Unlike previous models, this paper develops a unique transition system for each product, ensuring that predictions for one product are not influenced by others that may share the same machinery but have different cycle times. This specificity enhances the reliability of the predictions. In addition, the proposed model optimizes the weights of the transition system and machine learning methods using linear programming, contributing to the predictions' overall effectiveness.

3. Proposed Framework

This section presents and explains the proposed prediction model. Our model is composed of a linear combination of the predictions of five different models. One of those models is a transition system-based model (TSM), which relies heavily on process mining. The other four models are machine learning models, which are composed of a multiple linear regression model (MLR), a random forest regression model (RF), a support vector regression model (SVR), and a k-nearest neighbors regression model (KNN). The linear combination of all five models is done by optimizing an LP model, which minimizes the combined absolute errors of the predictions. Figure 2 presents a general overview of the framework.

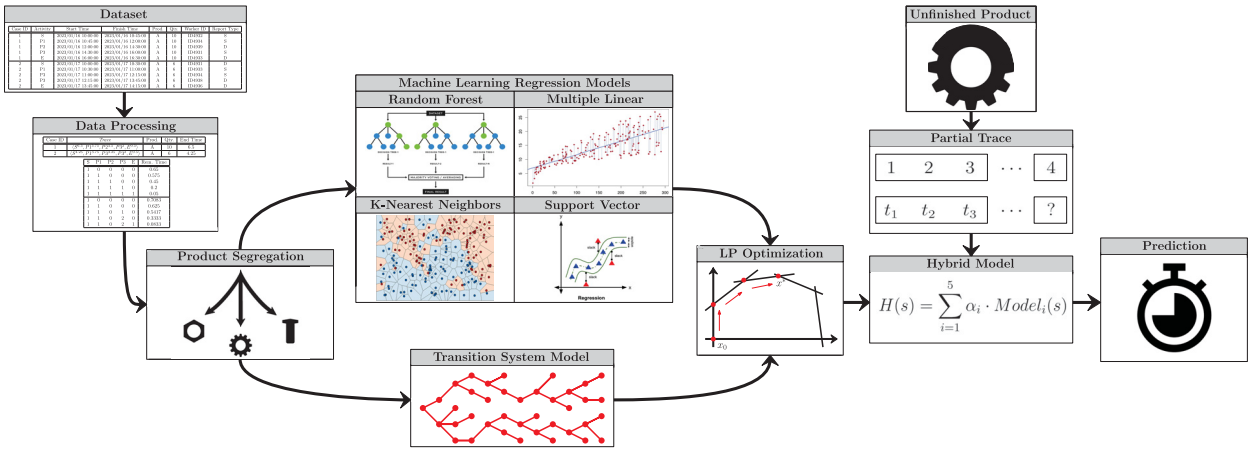

Figure 2: A general overview of the proposed framework.

Data from a process is collected in the form of event logs. Those datasets are then treated, filtered, and transformed into forms each model can read. Then, those treated datasets are separated by product. Models are then created based on each product dataset, i.e., each product has its own TSM, MLR, RF, SVR, and KNN. Based on the testing predictions of the five models, the weights of the linear combination of the predictions for the hybrid models are set by optimizing LP models. Finally, given an unfinished product with a partial trace, the hybrid model predicts its remaining time per product.

For the remainder of this section, we show how a transition system can be extracted from the event log and how to create the TSM from the transition system. Also, we explain how the TSM can predict the values of traces it has never seen, as this is of the utmost importance when cross-validation is performed. The machine learning models and the necessary database transformations are then explained. Here, as a means of comparison, two possible database transformations are presented, one that accounts for repetitions and one from [40] that does not. Finally, we explain the LP model, which optimizes the weights of each model's predictions for the hybrid model.

3.1. Transition System Based Model

A straightforward way to describe a transition system is that it represents all the paths in an event log. It starts with a mutual initial state from which all the existing paths in the log branch out.

More formally, a transition system can be modeled as a tuple (S, L, T), where S is the set of states (all possible states in the process), L is the set of the transition labels (events), and T, the labeled transition relation, is a subset of $S \times L \times S$ which describes how the system goes from one state to another. Transition systems are graphically represented as trees or graphs, in which the states are represented by the nodes and the labeled transition relations by the vertices. There are different ways to define states and transitions of a transition system. Here, the definitions from [25] will be used.

A state can be represented by a function l^{state} that produces some representation given a partial trace σ. Formally, $l^{state} \in \mathcal{T} \rightarrow \mathcal{R}$, where $\mathcal{T}$ is the set of possible traces and $\mathcal{R}$ is the set of possible representations (e.g., sequences, sets). As there is a need to label the states in the transition system, there is also a need to label the events, the transition

labels. Note that a concatenation of states can represent a path, so there must be a function to connect subsequent states or transitions. This transition has a label represented by the l^{event} function.

An event can be represented by a function l^{event} that produces some representation given an event e. Formally, $l^{event} \in \mathcal{E} \rightarrow \mathcal{R}$ where $\mathcal{E}$ is the set of possible events and $\mathcal{R}$ is the set of possible (event) representations (e.g., the corresponding activity name). Now that we have defined l^{state} and l^{event} functions, creating a transition system is possible. However, we must still define one last set function to expose the transition system-generating algorithm.

The function hd^k creates a sequence of a set's first k elements. For example, if we apply hd^3 to the set defined by the trace $\langle a, b, c, d, e, f \rangle$, we obtain the partial trace, or sub-trace, defined by the set $\langle a, b, c \rangle$. Algorithm 1 shows a pseudocode that we implemented in Python to create a transition system generating function based on an event log dataset in which each line contains a process and is separated by cases, and each case is ordered by time.

Algorithm 1 An algorithm to obtain all sub-traces in an event log

Require: A dataset DF
Ensure: A list of sub-traces ST
  $ST \leftarrow$ empty list
  for each case $c \in DF$ **do**
    for each $k \in (0 \leq k \leq |c|)$ **do**
      $trace \leftarrow$ empty list
      for each $j \in (0 \leq j \leq k)$ **do**
        $trace.add$(process j from case c)
      end for
      $ST.add(trace)$
    end for
  end for

This algorithm finds all the sub-traces in an event log by going through all observations, which are grouped by cases and ordered by start time. The sub-traces are obtained by listing all the activities/processes between the case's start and the sub-trace's end. Consequently, the number of sub-traces produced is equal to the number of observations from the generating dataset, as it allows repetition. This tolerance to repetitions will be essential in constructing the annotated transition system, as it has annotations that can be different for equal sub-traces.

After using Algorithm 1 and obtaining a list of all sub-traces in the log, the next step in the building of a transition system-based prediction model consists of attaching annotations [25], based on the target value, to all the different sub-traces and choosing a function that connects a partial trace to a prediction value. As the prediction target in this paper is the remaining time, a value related to it from each sub-trace will be the annotation of this sub-trace in the model.

Case ID	Activity	Start Time	Finish Time	Product	Qty
1	S	2023/01/16 10:00:00	2023/01/16 10:45:00	A	10
1	P1	2023/01/16 10:45:00	2023/01/16 12:00:00	A	10
1	P2	2023/01/16 12:00:00	2023/01/16 14:30:00	A	10
1	P3	2023/01/16 14:30:00	2023/01/16 16:00:00	A	10
1	E	2023/01/16 16:00:00	2023/01/16 16:30:00	A	10
2	S	2023/01/17 10:00:00	2023/01/17 10:30:00	A	6
2	P1	2023/01/17 10:30:00	2023/01/17 11:00:00	A	6
2	P3	2023/01/17 11:00:00	2023/01/17 12:15:00	A	6
2	P3	2023/01/17 12:15:00	2023/01/17 13:45:00	A	6
2	E	2023/01/17 13:45:00	2023/01/17 14:15:00	A	6

Table 2: An example of an event log.

Consider the following case example based on Table 2: running Algorithm 1 in this event log would give a list with the sub-traces exposed in Figure 3. Given a partial trace, we want to stipulate a good value for the remaining time.

A good and straightforward way to calculate the possible remaining time, given that one sub-trace can appear more than once, is to calculate its mean or median value. However, when accounting for processes that work in batches, a reasonable operation is to divide the remaining time by the quantity of products from that batch. In equation 1, we have an expression for the predicted remaining time per product of a sub-trace.

$$\text{Remaining Time}\,(Trace, n_{case}) = \frac{Trace_{case\_RT} - Trace_{act\_ST}}{n_{case}}. \tag{1}$$

In this remaining time per product equation (1), $Trace_{case\_RT}$ is the finish time of the whole case, e.g., for Case ID 1 in Table 2 it is "2023/01/16 16:30:00", $Trace_{act\_ST}$ is the start time of the activity from that case, and n_{case} is the number of products processed in that case.

It is possible to write a more compact notation for the event log now that the prediction of the remaining time per product does not need the date information. In Table 3, we have this compact form, where instead of the activities and the start and finish date, we have a Trace column, which has the trace of the completed case, and, in the superscript of each activity, there is the remaining time of the case when that activity happened.

Case ID	*Trace*	Product	Qty	End Time
1	$\langle S^{6.5}, P1^{5.75}, P2^{4.5}, P3^{2}, E^{0.5}\rangle$	A	10	6.5
2	$\langle S^{4.25}, P1^{3.75}, P3^{3.25}, P3^{2}, E^{0.5}\rangle$	A	6	4.25

Table 3: A compact form of the Table 2 log.

With a list of sub-traces and remaining times per product, it is possible to build an annotated transition system. Algorithm 2 does that by selecting all equal sub-traces and their remaining time per product. Then, as a system needs just one of the multiple equal sub-traces, it calculates the prediction time of that sub-trace as the mean (or median) value of the remaining time per product of all the equal sub-traces. Using Algorithm 2 with the sub-traces and remaining times per product of the example from Tables 2 and 3, we get the annotated transition system of Figure 3.

Algorithm 2 An algorithm to create an annotated transition system

Require: A indexed list of sub-traces ST and values V
Ensure: A annotated transition system ATS
  traces, *vals*, *traces_inds*, *list_vals* ← empty list
  for each $sb\_ind \in ST$ indexes **do**
    if $sb\_ind \notin traces\_inds$ **then**
      $traces.add(ST[sb\_ind])$
      $rep\_inds \leftarrow$ indexes of ST where it is equal to $ST[sb\_ind]$
      for each $rep\_ind \in rep\_inds$ **do**
        $traces\_inds.add(rep\_ind)$
        $list\_vals.add(V[rep\_ind])$
      end for
      $vals.add(mean(list\_vals))$
      $list\_vals \leftarrow$ empty list
    end if
  end for
  $ATS \leftarrow$ dictionary where the keys are *traces* and the values are *vals*

With the annotations in the transition system explained, we will now explain how the transition system model connects a partial trace to an existing sub-trace in the system. This connection is of primordial importance for the model, as its performance can change drastically depending on the connecting function.

As stated in [40], limitations of annotated transition system-based models appear when an input partial trace does not exist in the system. In those cases, a method for associating an unknown partial trace with a known one in the

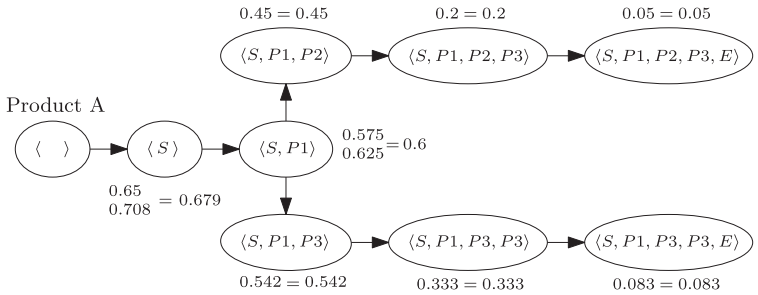

Figure 3: The proposed annotated transition system for the event log in Table 2.

system is needed. Our chosen method for connecting partial traces to ones in the system is based on a similarity score. A possible way to calculate a similarity score is using the Jaccard index, which is the intersection ratio by the union of two sets (sub-traces). However, the Jaccard index does not account for similarities in the sequence, e.g., relations between the activities. Therefore, we will introduce another way to calculate the similarity score of two sub-traces.

Our score depends on the size of the partial trace or the size of the system sub-trace and the number of coinciding activities in the traces, which must be in the same position. To find an appropriate match to the analyzed partial trace in the annotated transition system, the model searches all the sub-traces in the system and gives a similarity score to each one. The predicted remaining time per product given to the analyzed partial trace is an average of all the annotations from the sub-traces with the best score. Algorithm 3 shows a pseudocode for this connection function in the case of an unknown partial trace.

Algorithm 3 An algorithm to associate a prediction value to an unknown partial trace

Require: A transition system TS and a partial trace PT
Ensure: A prediction value *pred*

```
S_max ← 0
match ← empty list
for each trace ∈ TS do
    Size_max ← max(|PT|, |trace|)
    Size_min ← min(|PT|, |trace|)
    S ← 0
    if Size_max ≠ Size_min then                ▷ Optional penalty for different length sub-traces
        S ← -1/Size_max
    end if
    for each k ∈ (0 ≤ k ≤ Size_min) do
        if trace[k] is equal to PT[k] then
            S ← S + 1/Size_max
        end if
    end for
    match.add((S, trace))
    if S > S_max then
        S_max ← S
    end if
end for
Mtraces ← all trace in match where S is equal to S_max
pred ← mean value from the annotations of each trace in Mtraces
```

As seen in algorithm 3, the model can only make a decent prediction if there is a match between an activity from the partial trace and one from any sub-trace in the system. This fact is an essential factor and disadvantage of transition system models. Unlike machine learning models, which make predictions based on feature values and

can output predictions for whatever values its features have, transition system-based models, as presented here, need training data with a very high degree of similarity to the data in which it will be used.

3.2. Machine Learning Regression Models

To diversify how a prediction can be made, we will present four machine learning methods that will be used together with the transition system model to predict the remaining time per product in the hybrid model. The four used ML methods: multiple linear regression (MLR), random forest regression (RF), support vector regression (SVR), and k-nearest neighbors regression (KNN), were chosen mainly due to their popularity, simplicity of implementation, and performance. All these methods are widely used mainly in the data science and analysis area, being easy to implement due to Python's vast set of ML libraries and achieving great performances in their proper contexts. Another popular alternative ML method is artificial neural networks (ANN). ANNs were not used because finding the right architecture for each log and product would be much more complex than optimizing the models of the four methods used.

In addition to the motivation of using the ML methods to build the hybrid model, another motivation for their usage is to compare two event log encoding methods, one that does not account for activity repetition and one that does. An event log is not an appropriate form to directly apply an ML model. It usually needs a set of variables, the features, from which it can derive its internal parameters' optimal values. Thus, some encoding is needed to extract information from an event log using a machine-learning model. We will present two possible encodes. Those encodes transform each event log activity into a feature, e.g., in a column in the transformed dataset.

The first encoding is from [40]. It is similar to the famous one-hot encoding, which gives binary values to categorical features to inform if the action or thing represented by that feature happened or not. The remaining time per product is used for the target variable. Applying this transformation, which we will call binary, to the log in Table 2, we obtain Table 4. As it can be noticed in the transformed log, this encoding does not account for activity repetition. This lack of repetition information motivates the introduction of another encoding based on frequency.

The second encoding is very similar to the first, except if an activity happens more than once, the frequency of repetitions is used instead of the number 1 in the repetition and its subsequent lines. Table 5 has the result of applying this transformation to the log in Table 2.

S	P1	P2	P3	E	Rem. Time per Prod.
1	0	0	0	0	0.65
1	1	0	0	0	0.575
1	1	1	0	0	0.45
1	1	1	1	0	0.2
1	1	1	1	1	0.05
1	0	0	0	0	0.7083
1	1	0	0	0	0.625
1	1	0	1	0	0.5417
1	1	0	1	0	0.3333
1	1	0	1	1	0.0833

Table 4: Transformed event log from Table 2 using the binary encoding.

S	P1	P2	P3	E	Rem. Time per Prod.
1	0	0	0	0	0.65
1	1	0	0	0	0.575
1	1	1	0	0	0.45
1	1	1	1	0	0.2
1	1	1	1	1	0.05
1	0	0	0	0	0.7083
1	1	0	0	0	0.625
1	1	0	1	0	0.5417
1	1	0	2	0	0.3333
1	1	0	2	1	0.0833

Table 5: Transformed event log from Table 2 using the repetition encoding.

The lack of differentiation between lines 8 and 9 of Table 4 can be a problem for machine learning models. The only information the models have is this dataset, so there is no possible way for the models to understand that lines 8 and 9 come from different sub-traces that repeat an activity. So, the additional information that the frequency encoding provides could be helpful for more complex decision-based models, such as the random forest, which is based on decision trees.

Multiple Linear Regression: The MLR method estimates parameters that better describe some observations by applying a linear relation [51]. For their simplicity and the fact that many real applications have linear correlation, MLR models are the most commonly used data-fitting methods. This regression uses the relation $\boldsymbol{y} = \boldsymbol{X\theta} + \boldsymbol{\varepsilon}$. Given an observation vector $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$ with n observations and a matrix of variables $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ with p groups of variables, the MLR model tries to find the optimal vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^{n\times 1}$, which minimizes a vector of errors $\boldsymbol{\varepsilon} \in \mathbb{R}^{n\times 1}$.

Using Table 4 and Table 5 datasets as an example, the $\boldsymbol{y}$ vector corresponds to the "Rem. Time" column and the $\boldsymbol{X}$ matrix corresponds to the matrix formed by all the activities columns.

The fair simplicity of the MLR method, while being an advantage, can also be seen as a disadvantage. Unlike more complex machine learning methods, which have many variable hyperparameters that, if properly chosen, can seriously boost the method's performance, the MLR method has no hyperparameters.

Random Forest Regression: The RF method is a machine learning ensemble method that can be used for classification or regression [52]. It is called an ensemble method because of its construction. An RF model is created by using the prediction of various decision trees, which are constructed from random samples of the training dataset. As decision tree models are easily prone to overfitting, e.g., high variance, as the trees grow deeper, the RF method is a way to reduce the variance by using the average of a set of trees. The predicted value of an RF regression model, different from a classification model, is the average of the continuous value predicted in each decision tree.

The RF method can be considered a complex machine learning method when we look at the number of variations it can have. Apart from all the hyperparameters of the decision tree method, such as the depth of the tree, its maximum number of branches, or its minimum number of leaves on a branch, the RF method has its own hyperparameters, such as the number of trees, the size of the tree training samples, or the variables in which a tree is trained. Hence, finding the optimal RF model for a dataset can be fairly difficult, if not impossible. Nonetheless, all this complexity allows the construction of more robust and well-performing models.

Support Vector Regression: As the name implies, the SVR method is based on the Support Vector Machines (SVM) method. In the SVM, roughly speaking, an optimal hyperplane that separates the categorical data is searched by transforming the variables by applying kernel functions. However, in the case of the SVR, no data separation is needed; instead, data fitting is needed. In the SVR, an optimal hyperplane that better fits the training data is searched [53].

The SVR, while more complex than the MLR method, which has no hyperparameter, is less complex than the RF method, hyperparameters-wise. The primary hyperparameters of the SVR method are its kernel function, the kernel coefficient, and the regularization parameter.

K-Nearest Neighbors Regression: The KNN method is a classification method that classifies an observation based on the most common class among its k closest neighbors [54]. When applied to regression, instead of the most common class, the output is the average value of its k closest neighbors. An important technique that can improve the performance of a KNN model is using a weighted average when calculating the output of the k closest neighbors, and a standard scheme uses the inverse of the distance, $1/d$, as the weight.

When discussing method customization, the KNN is closer to the MLR method than the RF or SVR methods. While the KNN method still has hyperparameters, they are constrained. One is the number of neighbors k, which has to be a natural number and usually does not get very high; the other is the weight of the weighted average of the neighbors' values; and the third is the type of distance used.

3.3. Hybrid Model

The method introduced in this work builds upon the approach proposed by [40]. As proposed by [40], our approach employs a transition system built for each product type. However, our method enhances this by integrating a linear combination of the transition system with four different machine learning models: Multiple Linear Regression, Random Forest, Support Vector Regression, and K-Nearest Neighbors Regression. In contrast, [40] approach solely utilized the transition system alongside a linear regression model. Additionally, our method accounts for the repeatability of activities when transforming the event log into a data table for machine learning models, an essential consideration in production systems that needed to be addressed in the work of [40]. The following expression gives the prediction of the hybrid model (HM):

$$\boxed{\text{HM} = \alpha_1 \cdot TSM + \alpha_2 \cdot MLR + \alpha_3 \cdot RF + \alpha_4 \cdot SVR + \alpha_5 \cdot KNN.} \tag{2}$$

The α_is are the weights of the combination; they follow the relation $\sum_{j=1}^{5} \alpha_j = 1$ and are non-negatives. Those weights are obtained by optimizing a linear programming (LP) model. The logic of the LP model is to find the coefficients that minimize the sum of the absolute errors from a test set, e.g., create a model that is a union of the best-performing models.

Given the matrix *Models* with the predictions of remaining time values, which has n lines, one for each sub-trace, and five columns, one for each model, TSM, MLR, RF, SVR, and KNN, and given the vector $\boldsymbol{o}$ containing the actual remaining time values of each sub-trace, the LP model is defined as:

$$\text{Min} \sum_{i=1}^{n} \varepsilon_i^+ + \varepsilon_i^- \tag{3}$$

s.t:

$$\sum_{j=1}^{5} \alpha_j \cdot Models_{ij} + \varepsilon_i^+ - \varepsilon_i^- = o_i, \qquad \forall i = 1, ..., n; \tag{4}$$

$$\sum_{j=1}^{5} \alpha_j = 1; \tag{5}$$

$$\alpha_j, \varepsilon_i^+, \varepsilon_i^- \geq 0, \qquad \forall i = 1, ..., n, \forall j = 1, ..., 5. \tag{6}$$

The objective function in 3 minimizes the sum of the absolute predicted error, $|\varepsilon_i|$. The first set of constraints in 4 is an equality in which the weights and the errors must vary to make the weighted sum of the predictions, plus a positive or negative error, equal to the occurred value. The second constraint set in 5 is the already explained sum of the weights. The last constraint set in 6 is the classical non-negativity constraint of linear programming.

4. Case Study

This section presents, analyzes, and tests our methods on two types of logs, artificially created and from an actual manufacturer. In the first part, we focus on the artificial logs, explaining what they are and how they were generated. In the second part, we focus on the real manufacturers' log, where we present, study, and analyze it. In the last part, we present the error metrics used and the models' performance on all the presented logs.

4.1. Artificial Logs

The artificial logs were created using the Python 3 programming language. The artificial logs were generated considering the following parameters: there are three products, which differ in the machines they need; there are 13 different machines, which process time per product follow a normal distribution and differ depending on the product; the number of products of each batch varies from 10 to 100 uniformly; the number of products influence the total process time following a normal distribution; and there is a probability of process repetition.

Figure 4 shows the Petri net representation of the artificial log. In this figure, the activity repetition process is illustrated apart not to pollute the Petri net representation, as the representation of the repetition with the silent transition τ is the same for all transitions. In Table 6, we have the normal distribution of each machine's processing time per product for each product. The normal distribution for the variation of the total process time of the batch is $N(1, 0.05^2)$.

The probability of a case following a specific path was set as equal, so the number of cases of each product should be very similar in the artificial logs. The probability of process repetition mentioned before refers to the chance of a product having to repeat the same process consecutively, as in the path $\langle M1, M1, M3, M7, M7, M11, M13 \rangle$.

For our tests, we generated 5 logs, each with 1000 cases. These logs differ in the probability of process repetition, with 0%, 5%, 10%, 15%, and 20% chance of process repetition. Because of this variation in process repetition, the logs also differ in the number of events, with 5663, 5983, 6273, 6633, and 7146 events.

4.2. Real Industrial log

We have tested our approach on an actual industrial event log, which can be found in data.4tu.nl and has a total of 43 products and 31 activities. Table 7 has a fragment of the explored log. The first column of the Table shows the case ID in this log, representing the production order. The second column specifies the activity, in this case, an activity along with its performed machine. The third and fourth columns contain the activity's start and complete timestamps.

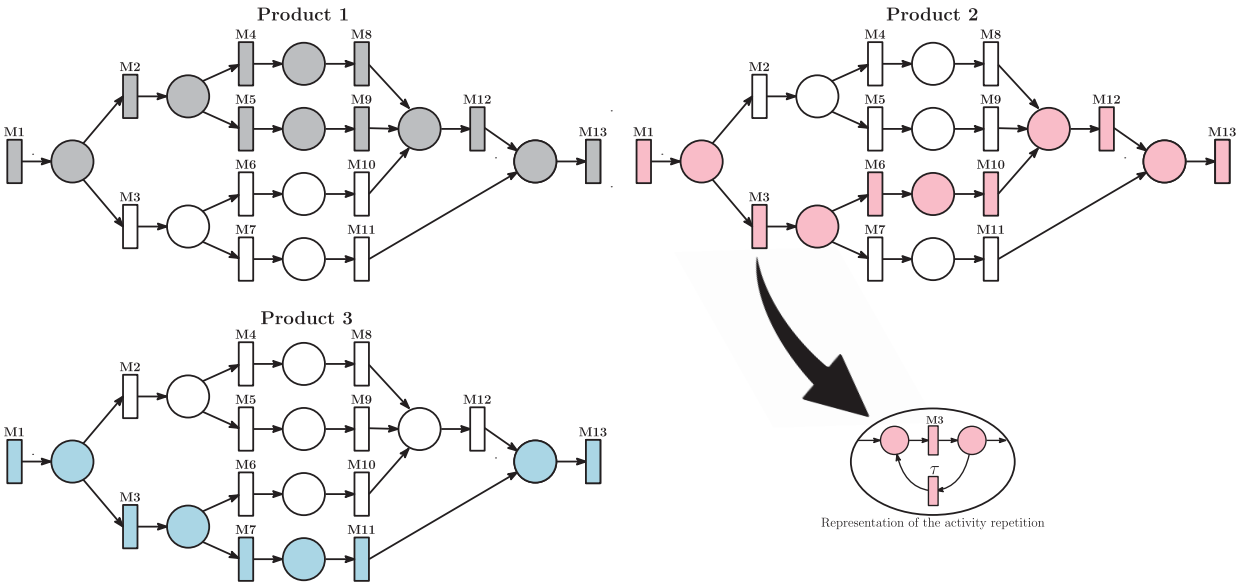

Figure 4: Petri nets of the artificial logs process, with each color representing a product path.

	Product		
Machine	1	2	3
M1	$N(0.3, 0.05^2)$	$N(0.25, 0.025^2)$	$N(0.2, 0.025^2)$
M2	$N(0.1375, 0.0175^2)$	-	-
M3	-	$N(0.225, 0.0125^2)$	$N(0.225, 0.0125^2)$
M4	$N(0.25, 0.05^2)$	-	-
M5	$N(0.25, 0.025^2)$	-	-
M6	-	$N(0.275, 0.0375^2)$	-
M7	-	-	$N(0.1175, 0.01625^2)$
M8	$N(0.2, 0.025^2)$	-	-
M9	$N(0.4, 0.025^2)$	-	-
M10	-	$N(0.325, 0.0125^2)$	-
M11	-	-	$N(0.2, 0.025^2)$
M12	$N(0.35, 0.025^2)$	$N(0.175, 0.0125^2)$	-
M13	$N(0.15, 0.01^2)$	$N(0.1, 0.005^2)$	$N(0.085, 0.0025^2)$

Table 6: Probability distribution of remaining time per product for each machine-product combination.

The number of products that are being processed resides in column 5. Lastly, the product being processed is indicated in the last column. Figure 5 shows a representation of the distribution of the machines and their respective processes on the production floor from the event log. The red arrows follow the trace of Case 1, shown in Table 7.

We notice a lack of events and extreme variability in this event log. Even though the log has 43 products, just three products are responsible for 56%, or 2586 of the 4543 events. Therefore, we chose to apply our methods to just these three most frequent products: Cable Head, with 1291 events; Ballnut, with 875 events; and Spur Gear, with 420 events.

Further study of the event log using DISCO software reveals the lack of variant repetition. For the three chosen products, the number of cases is equal to the number of variants, e.g., there is no process repetition, e.g., each batch was produced using a different sequence of activities, in this case, machines. Another fact to take note of in this log is the incredible number of activity repetitions, which, in this case where a batch of products is being manufactured,

Case ID	Resource	Start Timestamp	Complete Timestamp	Work Order Qty	Part Desc.
Case 1	Machine 4 - Turning & Milling	2012/01/29 23:24:00	2012/01/30 05:43:00	10	Cable Head
Case 1	Machine 4 - Turning & Milling	2012/01/30 05:44:00	2012/01/30 06:42:00	10	Cable Head
Case 1	Machine 4 - Turning & Milling	2012/01/30 06:59:00	2012/01/30 07:21:00	10	Cable Head
Case 1	Machine 4 - Turning & Milling	2012/01/30 07:21:00	2012/01/30 10:58:00	10	Cable Head
Case 1	Quality Check 1	2012/01/31 13:20:00	2012/01/31 14:50:00	10	Cable Head
Case 1	Machine 7- Laser Marking	2012/02/01 08:18:00	2012/02/01 08:27:00	10	Cable Head
Case 1	Machine 1 - Lapping	2012/02/14 00:00:00	2012/02/14 01:15:00	10	Cable Head
Case 1	Machine 1 - Lapping	2012/02/14 00:00:00	2012/02/14 01:15:00	10	Cable Head
Case 1	Machine 1 - Lapping	2012/02/14 09:05:00	2012/02/14 10:20:00	10	Cable Head
Case 1	Machine 1 - Lapping	2012/02/14 09:05:00	2012/02/14 09:38:00	10	Cable Head
Case 1	Machine 3 - Round Grinding	2012/02/14 09:13:00	2012/02/14 13:37:00	10	Cable Head
Case 1	Machine 3 - Round Grinding	2012/02/14 13:37:00	2012/02/14 15:27:00	10	Cable Head
Case 1	Quality Check 1	2012/02/16 06:59:00	2012/02/16 07:59:00	10	Cable Head
Case 1	Quality Check 1	2012/02/16 12:11:00	2012/02/16 16:12:00	10	Cable Head
Case 1	Quality Check 1	2012/02/16 12:43:00	2012/02/16 13:58:00	10	Cable Head
Case 1	Packing	2012/02/17 00:00:00	2012/02/17 01:00:00	10	Cable Head

Table 7: Case 1 of the event log.

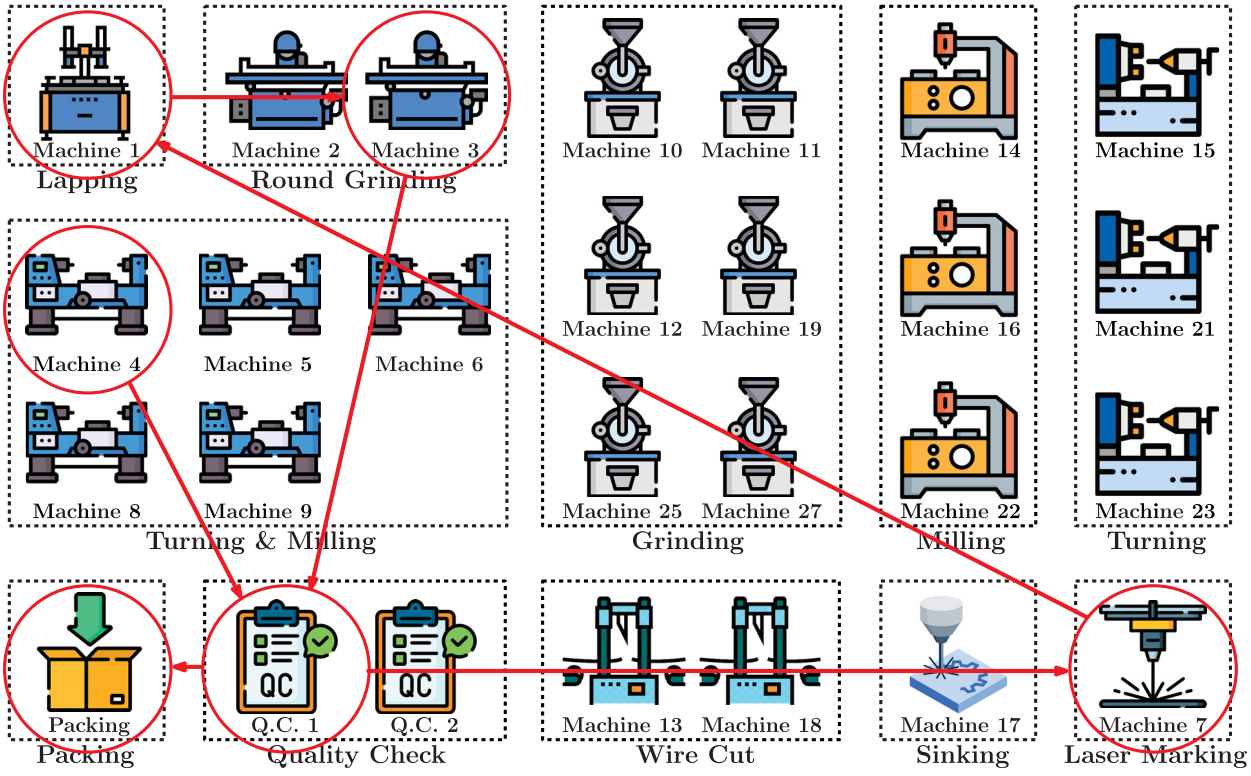

Figure 5: Layout of the industrial process from the event log, exemplifying the process of Case 1 from Table 7.

could be interpreted as the production of a whole batch, which was segmented because of production halts due to emergence stops or the use of the same machine for the production of other products. Frequency encoding can be a good alternative with almost no case without repetition.

Another important fact about this event log is the frequent concurrency of activities, as shown in lines 7 to 10, 14, and 15 of Table 7. This concurrency can be interpreted as the parallelization of the production line, where there is

more than one machine or worker doing the same process simultaneously. This kind of process behavior in event logs can negatively affect the prediction results of ML models as most ML approaches view event logs as merely sequential data rather than sequential manifestations of a concurrent system [22].

In Figure 6, we have a simplified process map of the event log extracted using DISCO. We emphasize that this map is a simplified version, where less frequent paths are not represented, because the entire map is unreadable as the process is unstructured, also called a Spaghetti process. This nonexistence of a defined process structure is true for all products with a reasonable number of cases, not just the process as a whole. All the presented information about the studied log demonstrates how it could be challenging to extract decent information from it.

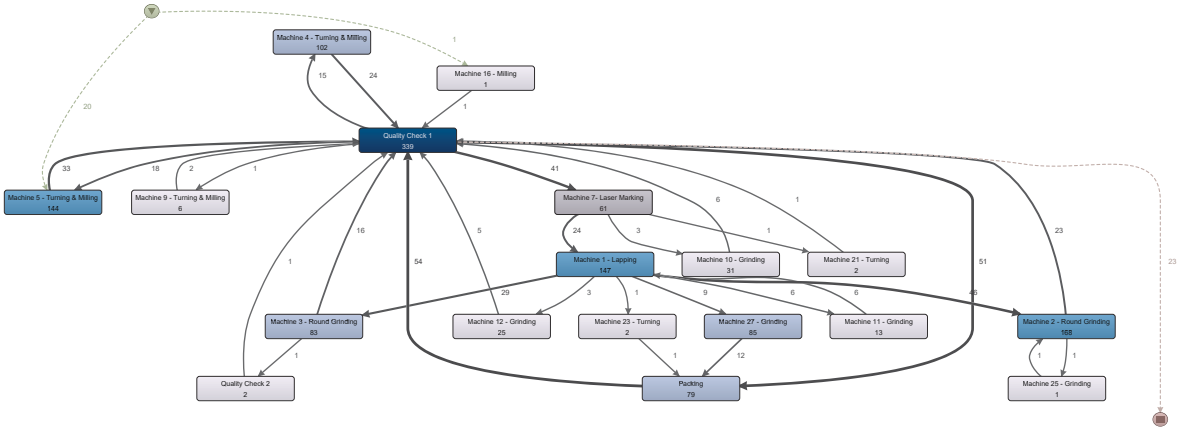

Figure 6: A representation of the process from the studied log created by DISCO.

4.3. Testing the models

For both the artificial and real logs, a total of 12 models were tested: the baseline model; the TSM; two models of each of the four ML methods, which vary on the used encoding, binary and repetition; and two models of the HM, which also vary on the used encoding for the ML models.

The baseline model is a very simplified version of the TSM. Instead of aggregating equal sub-traces, which can be seen as the whole picture of a case in an event, it just uses the last process performed in the sub-traces. As an example, instead of the tree created by the TSM in Figure 3, the baseline model would form just a list where the indexes are the activities/processes in the data and the values are the mean/median remaining time per product of every sub-trace where that activities/processes was the last performed. In the case of the data from Table 3 and taking activity P3 as an example, while in the TSM there are three different sub-traces and values which have P3 as the last activity, in the case of the baseline model, all these three different sub-traces would share the same value derived from their mean/median remaining time per product.

The training process was the same for all models: a 5-fold cross-validation [55] with shuffling of the events. Cross-validation is necessary to evaluate the models' capability of extrapolating from its training data and make predictions for unknown cases [56]. In this manner, we decided that a 5-fold, i.e., a data division of 80% for train and 20% for test, is enough to assess the models' quality. The shuffling is necessary because the logs are ordered by case and each case by time, so not doing a shuffle would mean that in the train-test division, whole groups of cases would be excluded from the training or the testing set.

The leading accuracy indicator chosen is the MAE, or mean absolute error, as it is the same, ignoring scale, as used in the LP model objective function 4. Another accuracy indicator used is the RMSE, or root mean square error, which emphasizes outliers errors. The last one is the MAPE, or mean absolute percentage error, a percentage indicator

of the MAE. In 7, we have the formulas for the used accuracy indicators.

$$\text{MAE} = \frac{1}{n}\sum_{k=1}^{n}\left|\overline{y}_k - y_k\right| \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(\overline{y}_k - y_k)^2} \qquad \text{MAPE} = \frac{100\%}{n}\sum_{k=1}^{n}\left|\frac{\overline{y}_k - y_k}{\overline{y}_k}\right| \tag{7}$$

where $\overline{y}_k$ is the actual value and y_k is the predicted value.

A disclaimer for the calculation of the MAPE: in the real log tests, it was calculated using both the mean and the median, as the computation by the mean is hugely affected by the fact that only a few smaller values can result in extremely large relative errors (dividing "almost zero" leads to "almost infinite"), which is a known problem of this error measure [57]. Calculating the MAPE using the median allows a better understanding of the models' performance for the whole log, as it ignores extreme relative errors.

The models' implementations, logs' dataset treatment and transformation were done using Python 3. Almost all implementation extensively used the numpy library. The pandas library was used for the data manipulation. All the machine learning algorithms are from the scikit-learn library. The pyomo library and the glpk solver were used for the LP model construction and optimization. Lastly, the error plots were made using the matplotlib library.

The hyperparameters' optimization of the ML models were done manually by varying the different hyperparameters values until a good local optima of each ML model, based on the test errors, were found. This search avoided spending too much time on just one model, as it could lead to biased results. As the MLR models have no hyperparameters, they were not modified. The modified hyperparameters of the RF models were the number of trees and the maximum tree depth, and the model optimization was done by prioritizing the MAE score. The kernels, their degrees and coefficients, and the regularization parameter were modified for the SVR models. Lastly, for the KNN models, the number of neighbors and the weight function were modified.

In the Tables and Figures with the tests results, we will be using the following notations: the "B" at the end of the models' names symbolizes the use of the binary encoding; the "R" at the end of the models' names symbolizes the use of the repeating encoding.

Tables 8, 9, and 10 have the mean MAE, RMSE, and MAPE from the cross-validation of each model for each of the three products: 1, 2, and 3, and each of the five repetition probabilities of the artificial logs. Tables 11, 12, and 13 have the mean MAE, RMSE, and MAPE from the cross-validation of each model for each of the three products: Cable Head (CH), Ballnut (BN), Spur Gear (SG). In Figure 7, we have graphs for the MAE and MAPE of the artificial logs experiments. In Figure 8, we have graphs for the MAE and MAPE of the real log's experiments.

As 5-fold cross-validation was done, the hybrid models have coefficients for each fold. In Tables 14 and 15, we have the mean coefficients of each hybrid model for each product of the artificial logs and the real log tests, respectively.

Models	MAE					RMSE					MAPE				
Prob. Rep.	0%	5%	10%	15%	20%	0%	5%	10%	15%	20%	0%	5%	10%	15%	20%
Baseline	0.0554	0.0794	**0.1169**	0.1538	0.2184	0.0803	**0.1365**	0.1960	0.2430	0.3317	6.1889	8.0272	**11.1376**	**13.3516**	18.2417
TSM	0.0555	0.0842	0.1221	0.1658	0.2349	0.0803	0.1436	0.1986	0.2523	0.3471	6.1980	9.1453	12.8635	16.2503	22.3857
MLR_B	0.0675	0.0917	0.1318	0.1700	0.2311	0.0869	0.1396	0.1932	0.2351	0.3218	10.0204	11.7712	15.6617	17.9113	23.9770
MLR_R	0.0675	0.1232	0.1809	0.2266	0.2933	0.0869	0.1688	0.2391	0.2868	0.3821	10.0204	22.4435	34.7992	40.1964	51.2728
RF_B	0.0558	0.0794	0.1170	**0.1537**	0.2180	0.0808	**0.1365**	0.1959	0.2425	0.3316	6.1881	8.0301	11.1508	13.3632	**18.1925**
RF_R	0.0558	0.0799	0.1190	0.1558	0.2198	0.0808	0.1370	0.1955	0.2420	0.3299	6.1881	8.1218	11.4135	13.7176	18.7902
SVR_B	0.0661	0.0940	0.1307	0.1672	0.2273	0.0854	0.1418	0.1959	0.2391	0.3286	13.0363	13.9129	17.1863	19.7737	24.4039
SVR_R	0.0661	0.0962	0.1334	0.1701	0.2253	0.0854	0.1418	**0.1927**	0.2391	0.3230	13.0363	16.3251	19.9748	21.3453	25.1160
KNN_B	0.0555	0.0817	0.1200	0.1692	0.2275	0.0803	0.1408	0.1967	**0.2338**	**0.3202**	6.1806	8.1833	11.7950	16.4591	21.6750
KNN_R	0.0555	0.0831	0.1267	0.1781	0.2491	0.0803	0.1408	0.1967	0.2415	0.3481	6.1806	8.7300	13.9371	19.2370	26.1582
HM_B	**0.0553**	**0.0793**	**0.1169**	**0.1537**	**0.2178**	**0.0801**	0.1366	0.1953	0.2418	0.3302	**6.1664**	**8.0126**	11.1970	13.4101	18.3180
HM_R	**0.0553**	0.0798	0.1183	0.1556	0.2194	**0.0801**	0.1368	0.1939	0.2415	0.3278	**6.1664**	8.1245	11.5899	13.7640	19.2555

Table 8: Errors of the models for product 1 from the artificial logs.

Models	MAE					RMSE					MAPE				
Prob. Rep.	0%	5%	10%	15%	20%	0%	5%	10%	15%	20%	0%	5%	10%	15%	20%
Baseline	0.0293	0.0665	0.0918	0.1304	0.1632	0.0417	0.1239	0.1520	0.2114	0.2573	4.4983	8.0131	10.4176	**14.0308**	16.3743
TSM	0.0293	0.0686	0.0923	0.1351	0.1707	0.0417	0.1264	0.1526	0.2138	0.2650	4.4983	8.6257	11.0923	15.9341	19.1106
MLR_B	0.0293	0.0731	0.1002	0.1408	0.1711	0.0417	**0.1188**	**0.1445**	**0.1999**	**0.2491**	4.5010	9.5722	13.2879	18.8824	20.8418
MLR_R	0.0293	0.1194	0.1466	0.2072	0.2506	0.0417	0.1606	0.1852	0.2635	0.3242	4.5010	34.2347	40.8524	56.6729	62.2815
RF_B	0.0293	0.0665	0.0919	0.1303	0.1630	0.0418	0.1240	0.1521	0.2108	0.2571	4.5001	**8.0041**	10.4283	14.0466	**16.3422**
RF_R	0.0293	0.0665	0.0906	0.1303	0.1629	0.0418	0.1238	0.1507	0.2103	0.2566	4.5001	8.0187	**10.4044**	14.0794	16.3784
SVR_B	0.0478	0.0910	0.1130	0.1452	0.1748	0.0571	0.1251	0.1500	0.2056	0.2559	19.4117	23.0801	25.2540	27.0915	28.1274
SVR_R	0.0478	0.0878	0.1109	0.1444	0.1746	0.0571	0.1244	0.1500	0.2046	0.2560	19.4117	19.6129	24.0299	26.3591	27.6952
KNN_B	0.0294	0.0721	0.1003	0.1401	0.1672	0.0419	0.1198	0.1451	0.2003	0.2529	4.5144	8.8619	12.5056	17.5382	19.1232
KNN_R	0.0294	0.0732	0.0998	0.1403	0.1752	0.0419	0.1224	0.1453	0.2042	0.2558	4.5144	9.2851	13.3714	18.4669	22.2621
HM_B	**0.0292**	**0.0664**	0.0912	**0.1301**	0.1627	**0.0416**	0.1236	0.1500	0.2097	0.2563	**4.4923**	8.0181	10.5976	14.1194	16.4965
HM_R	**0.0292**	**0.0664**	**0.0905**	0.1302	**0.1626**	**0.0416**	0.1231	0.1499	0.2098	0.2555	**4.4923**	8.0147	10.4250	14.1482	16.7252

Table 9: Errors of the models for product 2 from the artificial logs.

Models	MAE					RMSE					MAPE				
Prob. Rep.	0%	5%	10%	15%	20%	0%	5%	10%	15%	20%	0%	5%	10%	15%	20%
Baseline	0.0231	0.0445	0.0665	0.0758	0.1027	**0.0316**	0.0805	0.1159	0.1261	0.1605	5.6046	**8.7246**	**11.3866**	**13.1239**	16.7276
TSM	0.0231	0.0459	0.0678	0.0780	0.1054	**0.0316**	0.0823	0.1172	0.1278	0.1612	5.6046	9.3055	12.3233	14.3098	18.4082
MLR_B	**0.0230**	0.0480	0.0730	0.0823	0.1092	**0.0316**	**0.0781**	**0.1094**	0.1203	**0.1547**	5.5894	10.1874	14.1695	16.9123	21.9982
MLR_R	**0.0230**	0.0711	0.1088	0.1216	0.1516	**0.0316**	0.0972	0.1422	0.1607	0.1951	5.5894	27.7812	40.0128	44.1003	52.3384
RF_B	0.0231	0.0445	0.0665	0.0758	0.1025	0.0317	0.0805	0.1157	0.1260	0.1603	5.6032	**8.7246**	11.3923	13.1326	**16.7231**
RF_R	0.0231	0.0446	0.0665	0.0760	0.1026	0.0317	0.0807	0.1155	0.1261	0.1602	5.6032	8.7385	11.4050	13.2109	16.8009
SVR_B	0.0398	0.0740	0.0907	0.0945	0.1160	0.0502	0.0904	0.1159	0.1255	0.1593	25.2323	31.3303	31.8166	31.2830	32.7415
SVR_R	0.0398	0.0735	0.0902	0.0934	0.1156	0.0502	0.0915	0.1163	0.1249	0.1591	25.2323	31.0173	31.1032	30.2643	31.8700
KNN_B	0.0231	0.0457	0.0712	0.0808	0.1122	**0.0316**	0.0797	0.1101	**0.1202**	0.1715	5.6265	9.1742	13.0184	15.8258	20.3239
KNN_R	0.0231	0.0480	0.0738	0.0823	0.1140	**0.0316**	0.0803	0.1126	0.1225	0.1654	5.6265	10.0207	14.1451	16.4548	22.6303
HM_B	**0.0230**	**0.0444**	**0.0664**	**0.0757**	**0.1023**	**0.0316**	0.0801	0.1150	0.1255	0.1600	**5.5868**	8.7362	11.4301	13.1739	16.7454
HM_R	**0.0230**	0.0446	**0.0664**	0.0760	0.1025	**0.0316**	0.0803	0.1149	0.1258	0.1594	**5.5868**	8.7404	11.4630	13.2324	17.0276

Table 10: Errors of the models for product 3 from the artificial logs.

MAE	CH	BN	SG
Baseline	8.3423	5.8747	3.9360
TSM	4.3717	4.9450	3.0017
MLR_B	10.1812	7.0488	4.2198
MLR_R	10.2119	7.5439	4.4750
RF_B	6.2484	5.3294	3.3700
RF_R	5.6483	5.2264	3.4164
SVR_B	6.1168	5.2067	3.2750
SVR_R	5.7978	5.1382	3.0943
KNN_B	7.7062	5.4610	3.6392
KNN_R	5.8211	5.4791	4.0967
HM_B	**4.3476**	4.8385	**2.8121**
HM_R	4.3543	**4.8084**	2.8303

Table 11: MAE of all models for each of the tree products.

RMSE	CH	BN	SG
Baseline	21.4302	14.3207	7.1721
TSM	17.3217	13.8988	6.6488
MLR_B	19.9699	13.3708	6.2892
MLR_R	21.2111	13.8471	6.4195
RF_B	18.1089	13.4592	6.0951
RF_R	18.1519	**13.3185**	6.1731
SVR_B	18.7195	13.5407	6.2923
SVR_R	18.8092	13.5647	**5.7806**
KNN_B	19.5602	13.5082	6.4981
KNN_R	19.4075	14.2169	7.7105
HM_B	17.0915	13.4764	5.9150
HM_R	**17.0609**	13.4366	5.7933

Table 12: RMSE of all models for each of the tree products.

5. Results and Discussions

Analyzing the results of the artificial logs tests from Tables 8, 9 and 10, an interesting result is noticeable: all models, except the MLR with repetition encoding and the two SVRs, show very similar performances. Even more interesting is how the baseline model outperformed almost all models in all artificial log instances. The simplicity of the logs can explain this incredible performance of a really simple model. This will be clearer when discussing the real log results, where the baseline model performed very poorly.

Another noticeable aspect of the models' performance in the artificial logs is how none of the models were able to cope with the increased probabilities of repetition. A reason for this behavior is the increase of possible activity paths that the activity repetition enables, thus increasing the variability in remaining time values, i.e., increasing the inherent randomness of the data.

MAPE	CH		BN		SG	
Baseline	71.9816	(21942)	79.2313	(2047)	67.3916	(9258)
TSM	**2.5956**	(10405)	**23.9265**	(1248)	**17.4189**	(**1116**)
MLR_B	100.8389	(38777)	154.0191	(4760)	79.2256	(8515)
MLR_R	95.6097	(27538)	199.7470	(8771)	102.5075	(24976)
RF_B	30.4513	(18476)	55.0757	(1363)	58.2252	(1689)
RF_R	13.6041	(15813)	50.5332	(1814)	48.2315	(2044)
SVR_B	22.0140	(17368)	57.7457	(1242)	52.0886	(1752)
SVR_R	17.8158	(13309)	60.4407	(1541)	46.2241	(4519)
KNN_B	39.3435	(**6607**)	60.1598	(**1188**)	55.0322	(1760)
KNN_R	4.6462	(13878)	48.1100	(1550)	35.8785	(5791)
HM_B	3.8588	(11141)	35.0561	(1325)	28.5584	(1488)
HM_R	3.2231	(10886)	38.0854	(1372)	32.1206	(2179)

Table 13: MAPE of all models for each of the tree products. For each products there are two values, the one in the left is the calculated using the median and the one in the right is using the mean.

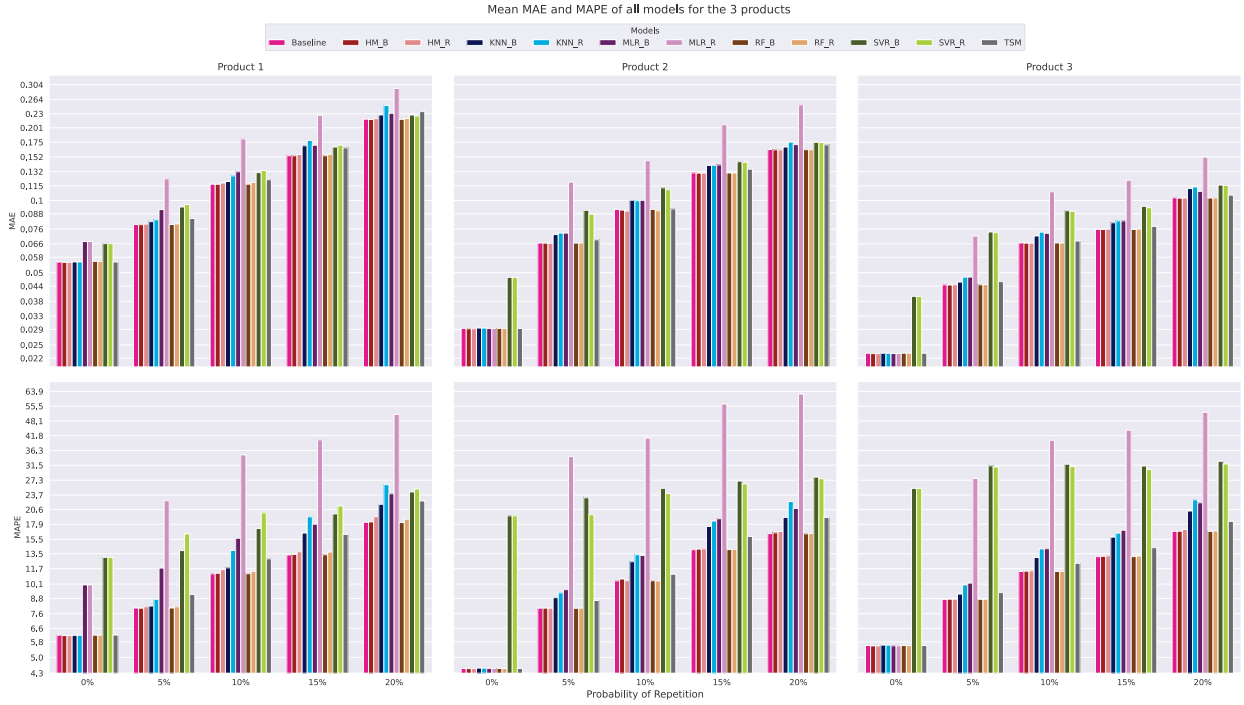

Figure 7: Mean MAE and MAPE of all models tested on the artificial logs.

For the MAE values of the artificial logs tests, the overall best model is the hybrid with binary encoding, which showed better or equal results in all instances, with just two exceptions in product 2 with 10% and 20% probability of repetition, where the hybrid with frequency encoding performed better. As the hybrid models are optimized based on the MAE, this supremacy on MAE values does not necessarily translate to better RMSE or MAPE values.

On an RMSE-focused analysis, the results are mixed: for product 1, the best model is the KNN with binary encoding; for products 2 and 3, the best model is, surprisingly, the simplest ML model, the MLR with binary encoding. A good performance of the MLR model is not a surprise, given that the artificial logs follow well-behaved paths with normally distributed remaining time values. Also, better results on a less punitive metric on general models are expected, as the RMSE punishes more severely outliers than the MAE. Looking at the MAPE values, the baseline model and the RF with binary encoding are consistently the best models when there is any activity repetition. The hybrid models showed a lower MAPE just in the cases without repetition. The MAPE values of the SVR models of product 3 express a very interesting and unique behavior. In addition to their odd behavior of not having their

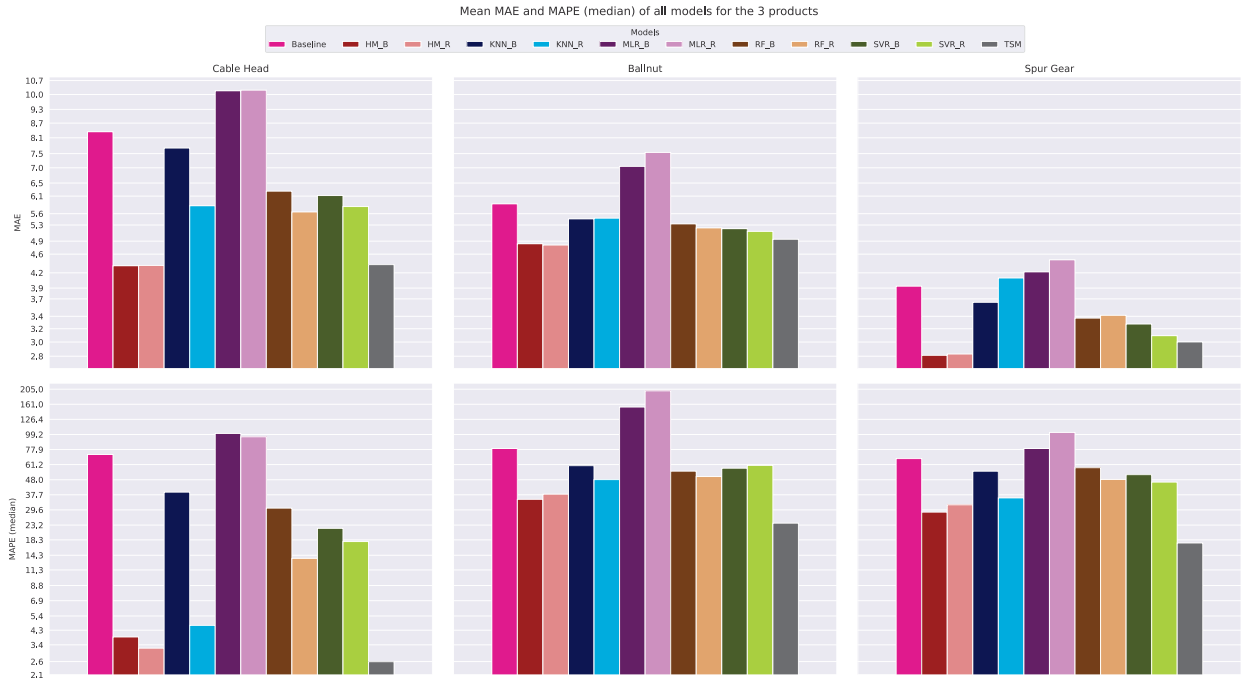

Figure 8: Mean MAE and MAPE of all models tested on the real log.

performance as affected by the increase in the probability of repetition as all the other models, they performed better in the 15% than in the 5% and 10% probability of repetition.

Overall, the artificial tests results showed that most models behaved very similarly and performed well on moderately well distributed-data and well-behaved paths. There are only two types of models that showed poor or odd performance: the MLR with frequency encoding and both SVR models. It is noticeable how the repetition encoding is not a good encoder for the MLR model, showing why the encoding is critical as it severely affected the performance of a method. As for the SVR models, they performed worse on the 0% probability of repetition instance than the rest. However, as the repetition probability increased, their scores got closer to the rest of the models. This indicates that the SVR can be a good choice when dealing with data with a lot of randomness.

Now, we shift our analysis to the real industrial log tests. From the results in Table 11, we can notice that MAE values-wise, almost all models for all the three products had a better performance than the baseline, with the exceptions being the MLR_B the MLR_R. These results demonstrate well the importance of choosing the suitable method for the data as, discordant with the artificial logs results where the baseline performed really well and the MLR_B did not perform poorly, for the messy data in the real log, these simpler methods cannot reach an acceptable performance level. When analyzing the MAE results, it is noticeable how the different encodings affected each method and product differently. Excluding the MLR models, as the same pattern from the artificial logs tests of the repetition encoding being worse happens again, for product CH, the models with the repetition encoding showed consistently better results. However, the results for products BN and SG are mixed.

Comparing the TSM with the ML models shows that it performed better than all the ML models. These results are not surprising, given our analysis of the event log. For an event log with no visible pattern, the ML models that usually search for observations with similar behaviors to construct their prediction models may struggle. Furthermore, as the TSM is based on an annotated transition system, its predictions are directly based on the observed values. Moreover, because the TSM makes its predictions based on the similarity score, its predictions are not based on pattern recognition but on trace similarity. In this event log where there are a lot of unique traces, if a trace is sufficiently similar to another, there is a high probability that they are related. Consequently, the prediction will most likely be more accurate.

Analyzing the results of the ML models using the two encodings, we can say that in the artificial logs, the models

Product 1	Prob. Rep.	TSM	MLR	RF	SVR	KNN
HM_B	0%	0.3993	0.0118	0.1328	0.0045	0.4517
HM_R		0.3993	0.0118	0.1328	0.0045	0.4517
HM_B	5%	0.0480	0.0281	0.8270	0.0000	0.0968
HM_R		0.0422	0.0103	0.8319	0.0155	0.1000
HM_B	10%	0.0518	0.0188	0.9116	0.0	0.0178
HM_R		0.2102	0.0095	0.7208	0.0012	0.0583
HM_B	15%	0.0000	0.0268	0.9649	0.0083	0.0000
HM_R		0.0601	0.0098	0.9183	0.0000	0.0119
HM_B	20%	0.0143	0.0262	0.9343	0.0000	0.0253
HM_R		0.0133	0.0050	0.8635	0.0969	0.0213
Product 2	Prob. Rep.	TSM	MLR	RF	SVR	KNN
HM_B	0%	0.2753	0.0000	0.3953	0.0071	0.3222
HM_R		0.2753	0.0000	0.3953	0.0071	0.3222
HM_B	5%	0.0244	0.0314	0.9367	0.0023	0.0052
HM_R		0.0246	0.0039	0.8981	0.0066	0.0669
HM_B	10%	0.4153	0.0047	0.5226	0.0004	0.0571
HM_R		0.0553	0.0036	0.9030	0.0000	0.0381
HM_B	15%	0.0440	0.0000	0.9086	0.0064	0.0410
HM_R		0.0340	0.0000	0.9392	0.02124	0.0056
HM_B	20%	0.0930	0.0257	0.8454	0.0024	0.0335
HM_R		0.0784	0.0015	0.8419	0.0369	0.0413
Product 3	Prob. Rep.	TSM	MLR	RF	SVR	KNN
HM_B	0%	0.0000	0.6918	0.1121	0.0046	0.1914
HM_R		0.0000	0.6918	0.1121	0.0046	0.1914
HM_B	5%	0.0601	0.0000	0.6805	0.0030	0.2564
HM_R		0.0558	0.0052	0.8740	0.0037	0.0613
HM_B	10%	0.0995	0.0071	0.8566	0.0021	0.0346
HM_R		0.0778	0.0030	0.8758	0.0039	0.0396
HM_B	15%	0.0183	0.0237	0.9429	0.0011	0.01413
HM_R		0.0162	0.0063	0.9605	0.0015	0.0155
HM_B	20%	0.0714	0.0323	0.8232	0.0047	0.0684
HM_R		0.0992	0.0071	0.8653	0.0042	0.0243

Table 14: Mean coefficients of all hybrid models for the artificial logs tests.

Cable Head	TSM	MLR	RF	SVR	KNN
HM_B	0.9586	0.0002	0.0143	0.0258	0.0010
HM_R	0.9447	0.0014	0.0295	0.0036	0.0208
Ballnut	TSM	MLR	RF	SVR	KNN
HM_B	0.6825	0.0134	0.0161	0.2165	0.0716
HM_R	0.5909	0.0022	0.0227	0.3213	0.0630
Spur Gear	TSM	MLR	RF	SVR	KNN
HM_B	0.6704	0.0044	0.0723	0.1587	0.0943
HM_R	0.5036	0.0260	0.0705	0.3999	0.0000

Table 15: Mean coefficients of all hybrid models.

with the binary encoding were better, while in the real log, the models with the repetition encoding performed better. The differences between the encodings in the artificial logs were very slight, excluding the MLR with repetition encoding, with the binary encoding being better by a small margin. The difference is also slight in the real log when we just count how many times each encoding was better than the other. However, contrary to the MLR models in the artificial logs and real log, the SVR models performed much better with the repetition encoding, and there was

an isolated case for the KNN where the binary encoding performed very poorly. From these results, the repetition information may be more suitable for more complex methods, such as the RF or the SVR, in contrast to simpler models, such as the MLR, which can not utilize the additional information or even the KNN.

It is noticeable when analyzing the MAE values and the hybrid models' coefficients in Tables 14 and 15 that the coefficients follow a particular logical pattern. For all instances, models that perform better have a higher coefficient value, i.e., have more influence on the hybrid models. As the coefficients follow this pattern, what we have in Table 14 is that, on average, the models that make up more of the hybrid models for the artificial logs are mainly the TSM and RF models, with the KNN models being less prominent but still having some influence. In table 15, we have a monopoly of the TSM for the CH hybrid model, while the BN and SG hybrid models are more diverse. The TSM still has the biggest coefficients for these last two products, but the SVR has a significant influence, and the RF and KNN also participate.

When comparing the differences between the hybrids and the other models, an interesting detail can be noticed: the bigger the difference between the best non-hybrid model and the other models, the smaller the difference between the hybrid and the best non-hybrid model. This is most noticeable when comparing the results from product CH, where the TSM has a much better MAE than the ML models but is just slightly worse than the hybrid models, with products BN and SG, where the TSM has a similar MAE as the ML models but is noticeably worst than the hybrid models.

These differences in MAE make the interpretation of hybrid models' coefficients values clearer. A pattern in both artificial and real logs results connecting these coefficients and the model's performance is that the better a model's MAE is, the bigger its coefficient is in the hybrid models. To demonstrate this pattern, we ordered the models by their MAE values, ascending order, and by their hybrid models' coefficients values, descending order, of each instance and log. Table 16 shows the mean absolute difference in the order of the MAE and hybrid models' coefficients values positions of each model for the artificial and real logs. In Table 16, no model had a mean difference in position bigger than one, which means that, on average, if a model has a specific position in MAE performance, its hybrid model coefficient will have this same position in coefficient value.

Log	TSM	MLR	RF	SVR	KNN
Artificial	0.7333	0.9333	0.2000	0.6667	0.7333
Real	0.0000	0.1667	0.5000	0.1666	0.8333

Table 16: Mean absolute difference of the position of the models in a MAE rank and a hybrid models' coefficient rank.

6. Conclusion

This paper presented some remaining time prediction methods based on process mining, machine learning, and a hybrid approach. As an annotated transition system, the transition system model was established based on process mining techniques. We introduced four machine learning methods and two encodings, which were used to present our hybrid approach that combines process mining with machine learning methods. The presented models are "product-oriented" and capable of coping with manufacture particularities, in which traces are represented as the activities already performed in the process, and a prediction of an incomplete trace is performed.

The time predictions made by the framework presented in this paper are remaining time per product, which deals with batch-type manufacturing where products can have similar production times but different batch sizes. Furthermore, the presented annotated transition system-based model (TSM) introduces a similarity score based on sequence and size similarity, which copes with the "no trace" limitation of a transition system, e.g., if there is no equal sub-trace in the system, it cannot establish a prediction value.

We introduced four machine learning methods: multiple linear regression, random forest regression, support vector regression, and k-nearest neighbors regression. Each of these methods is very different from the others, and the use of these methods as a way to introduce the hybrid approach allowed the impact of different encodings of event logs for machine learning methods to be shown. Two encodings were presented and compared: the binary encoding, which does not hold the activity repetition information, and the repetition encoding, which does.

We tested and validated the presented methods and approach in two different situations: on artificially created logs and on an actual data log from a real manufacturer. The artificial logs recreate the log of three products using their Petri net, the probability distributions of their activities' remaining time values, and their activity repetition probabilities. Consequently, the artificial logs are well-behaved and show a well-defined process structure. On the other hand, our apriori analysis of the real industrial log demonstrated some of the challenges that this log had, such as the lack of trace repetition, an incredible amount of variants, the occurrence of concurrent activities, and activity repetition.

Our tests of the models for the artificial logs showed surprising results, with almost all models performing well and the simple baseline model performing incredibly well. The fact that almost all models' performance equally decreased as the repetition probability increased demonstrated how difficult it is to cope with inherent randomness and how unpredictable reworking can seriously affect a production line. Overall, the best-performing method in these tests was the hybrid with binary encoding, but the hybrid with frequency encoding also showed similar results.

For the real log, in our leading indicator, the MAE, the model that performed better was the hybrid model with the repetition encoding. The differences between the hybrid models and the best non-hybrid model were slim when the hybridization was low and higher when the hybridization was high, demonstrating that the hybrid models perform better if the models that make it are well mixed.

Concerning the encoding comparison, the results were different for the two tests. According to the results of the artificial logs, the best encoding is the binary. However, the real log results show that the repetition encoding performed better overall, especially for the more complex and better-performing ML models: RF and SVR.

The estimates provided by the presented methods are satisfactory as quantified by the accuracy measures. A good remaining time estimate could be the difference between an accomplished and a non-accomplished customer expectation. Considering the annotated transition system-based method, its ease of use appeals to manufacturing professionals. Since it does not require any specific process mining software, the learning curve to use it is greatly reduced, and there are no associated costs. As for the machine learning methods, they are less automated for use than the TSM, as they require hyperparameter tuning and, as shown, have a performance that varies a lot depending on the encoding used. The hybrid method is as appealing as the annotated transition system-based method when there are at least two models and a linear programming solver.

Given the clear difference that different encodings can have in these ML methods' performance, future research could be associated with improving the ML methods for remaining time prediction in a process log. A more profound analysis of the process structure and the ML method can help to develop a better encoding that favors the process structure and method logic.

Our prediction method, built on machine learning models (ML), can provide greater reliability to production and planning control managers. However, a challenge that arises is how to generate confidence in these models for PPC managers. In this context, [58] proposes to apply explainable AI methods to create trustworthy AI-based manufacturing systems. Consequently, we are investigating the enrichment of our system to explain their reasoning processes and outputs (e.g., predictions) automatically.

Moreover, accurate production order completion times predictions allow managers to provide more reliable delivery timelines to customers, improving customer satisfaction and trust. This capability also aids in identifying potential delays earlier in the production process, enabling proactive measures to mitigate risks, such as adjusting schedules or reallocating resources. In highly competitive markets, where meeting deadlines is crucial, the ability to forecast production timelines with greater certainty can become a key differentiator, enhancing the company's reputation and potentially leading to increased market share.

Building on the promising results of our current approach, several avenues for future research have been identified to further enhance the accuracy and applicability of remaining time predictions in production orders. Firstly, future work will explore integrating additional machine learning models into our hybrid approach. By incorporating a broader range of models, we aim to improve the robustness and precision of our predictions, adapting to varying production environments and complexities.

Another important direction for future research involves including failure and maintenance stop events in the event log. By accounting for these critical factors, we can develop a more comprehensive model that reflects real-world disruptions, thereby increasing the reliability of the predictions. Additionally, integrating our approach with scheduling systems represents a crucial step towards not only keeping production orders within their due dates but also minimizing delays. This integration will allow for more dynamic and responsive scheduling, aligning production timelines more closely with operational realities.

Finally, future work will also consider including additional attributes in the event log, such as product color, family, and raw materials. By expanding the range of attributes analyzed, we can capture a more nuanced understanding of how different factors influence production times, leading to even more accurate and context-sensitive predictions. These investigations will contribute to the continuous improvement of production order management, providing valuable tools for optimizing manufacturing processes.

References

[1] S. Rinderle-Ma, J. Mangler, Process automation and process mining in manufacturing, in: International conference on business process management, Springer, 2021, pp. 3–14.

[2] H. Cañas, J. Mula, F. Campuzano-Bolarín, R. Poler, A conceptual framework for smart production planning and control in industry 4.0, Computers & Industrial Engineering 173 (2022) 108659.

[3] Z. Müller-Zhang, T. Kuhn, P. O. Antonino, Towards live decision-making for service-based production: Integrated process planning and scheduling with digital twins and deep-q-learning, Computers in Industry 149 (2023) 103933.

[4] Y. Cheng, K. Chen, H. Sun, Y. Zhang, F. Tao, Data and knowledge mining with big data towards smart production, Journal of Industrial Information Integration 9 (2018) 1–13.

[5] F. F. Alves, T. H. Nogueira, M. G. Ravetti, Learning algorithms to deal with failures in production planning, Computers & Industrial Engineering 169 (2022) 108231.

[6] C. L. Garay-Rondero, J. L. Martinez-Flores, N. R. Smith, S. O. C. Morales, A. Aldrette-Malacara, Digital supply chain model in industry 4.0, Journal of Manufacturing Technology Management 31 (2020) 887–933.

[7] R. Rai, M. K. Tiwari, D. Ivanov, A. Dolgui, Machine learning in manufacturing and industry 4.0 applications, 2021.

[8] D. Luo, S. Thevenin, A. Dolgui, A state-of-the-art on production planning in industry 4.0, International Journal of Production Research 61 (2023) 6602–6632.

[9] M. Liu, X. Li, J. Li, Y. Liu, B. Zhou, J. Bao, A knowledge graph-based data representation approach for iiot-enabled cognitive manufacturing, Advanced Engineering Informatics 51 (2022) 101515.

[10] N. Tax, I. Verenich, M. La Rosa, M. Dumas, Predictive business process monitoring with lstm neural networks, in: Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings 29, Springer, 2017, pp. 477–492.

[11] W. Van Der Aalst, W. van der Aalst, Data science in action, Springer, 2016.

[12] W. Van der Aalst, A. Adriansyah, B. Van Dongen, Replaying history on process models for conformance checking and performance analysis, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2 (2012) 182–192.

[13] C. dos Santos Garcia, A. Meincheim, E. R. F. Junior, M. R. Dallagassa, D. M. V. Sato, D. R. Carvalho, E. A. P. Santos, E. E. Scalabrin, Process mining techniques and applications–a systematic mapping study, Expert Systems with Applications 133 (2019) 260–295.

[14] A. Corallo, M. Lazoi, F. Striani, Process mining and industrial applications: A systematic literature review, Knowledge and Process Management 27 (2020) 225–233.

[15] R. Lorenz, J. Senoner, W. Sihn, T. Netland, Using process mining to improve productivity in make-to-stock manufacturing, International Journal of Production Research 59 (2021) 4869–4880.

[16] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, I. Teinemaa, Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (2019) 1–34.

[17] W. Rizzi, C. Di Francescomarino, F. M. Maggi, Explainability in predictive process monitoring: when understanding helps improving, in: International Conference on Business Process Management, Springer, 2020, pp. 141–158.

[18] J. Evermann, J.-R. Rehse, P. Fettke, A deep learning approach for predicting process behaviour at runtime, in: Business Process Management Workshops: BPM 2016 International Workshops, Rio de Janeiro, Brazil, September 19, 2016, Revised Papers 14, Springer, 2017, pp. 327–338.

[19] I. Teinemaa, M. Dumas, M. L. Rosa, F. M. Maggi, Outcome-oriented predictive process monitoring: Review and benchmark, ACM Transactions on Knowledge Discovery from Data (TKDD) 13 (2019) 1–57.

[20] W. Fang, Y. Guo, W. Liao, K. Ramani, S. Huang, Big data driven jobs remaining time prediction in discrete manufacturing system: a deep learning-based approach, International Journal of Production Research 58 (2020) 2751–2766.

[21] J. C. Serrano-Ruiz, J. Mula, R. Poler, Job shop smart manufacturing scheduling by deep reinforcement learning, Journal of Industrial Information Integration (2024) 100582.

[22] P. Ceravolo, S. B. Junior, E. Damiani, W. M. van der Aalst, Tailoring machine learning for process mining, ArXiv abs/2306.10341 (2023). URL: `https://api.semanticscholar.org/CorpusID:259203764`.

[23] C. Di Francescomarino, C. Ghidini, Predictive process monitoring, Process Mining Handbook. LNBIP 448 (2022) 320–346.

[24] C. Di Francescomarino, C. Ghidini, F. M. Maggi, F. Milani, Predictive process monitoring methods: Which one suits me best?, in: International conference on business process management, Springer, 2018, pp. 462–479.

[25] W. Aalst, van der, M. Schonenberg, M. Song, Time prediction based on process mining, Information Systems 36 (2011) 450–475. doi:`10.1016/j.is.2010.09.001`.

[26] M. Polato, A. Sperduti, A. Burattin, M. de Leoni, Data-aware remaining time prediction of business process instances, in: 2014 International Joint Conference on Neural Networks (IJCNN), IEEE, 2014, pp. 816–823.

[27] M. Polato, A. Sperduti, A. Burattin, M. d. Leoni, Time and activity sequence prediction of business process instances, Computing 100 (2018) 1005–1031.

[28] F. Folino, M. Guarascio, L. Pontieri, Discovering high-level performance models for ticket resolution processes: (short paper), in: On the Move to Meaningful Internet Systems: OTM 2013 Conferences: Confederated International Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings, Springer, 2013, pp. 275–282.

[29] F. Folino, M. Guarascio, L. Pontieri, Mining predictive process models out of low-level multidimensional logs, in: Advanced Information Systems Engineering: 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings 26, Springer, 2014, pp. 533–547.

[30] M. Ceci, P. F. Lanotte, F. Fumarola, D. P. Cavallo, D. Malerba, Completion time and next activity prediction of processes using sequential pattern mining, in: Discovery Science: 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings 17, Springer, 2014, pp. 49–61.

[31] A. Rogge-Solti, M. Weske, Prediction of remaining service execution time using stochastic petri nets with arbitrary firing delays, in: Service-

Oriented Computing: 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings 11, Springer, 2013, pp. 389–403.
[32] A. Rogge-Solti, M. Weske, Prediction of business process durations using non-markovian stochastic petri nets, Information Systems 54 (2015) 1–14.
[33] S. Pandey, S. Nepal, S. Chen, A test-bed for the evaluation of business process prediction techniques, in: 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), IEEE, 2011, pp. 382–391.
[34] A. Senderovich, C. Di Francescomarino, C. Ghidini, K. Jorbina, F. M. Maggi, Intra and inter-case features in predictive process monitoring: A tale of two dimensions, in: Business Process Management: 15th International Conference, BPM 2017, Barcelona, Spain, September 10–15, 2017, Proceedings 15, Springer, 2017, pp. 306–323.
[35] M. De Leoni, W. M. Van der Aalst, M. Dees, A general framework for correlating business process characteristics, in: International Conference on Business Process Management, Springer, 2014, pp. 250–266.
[36] S. Pravilovic, A. Appice, D. Malerba, Process mining to forecast the future of running cases, in: New Frontiers in Mining Complex Patterns: Second International Workshop, NFMCP 2013, Held in Conjunction with ECML-PKDD 2013, Prague, Czech Republic, September 27, 2013, Revised Selected Papers 2, Springer, 2014, pp. 67–81.
[37] M. De Leoni, W. M. Van Der Aalst, M. Dees, A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs, Information Systems 56 (2016) 235–257.
[38] I. Verenich, M. Dumas, M. La Rosa, H. Nguyen, Predicting process performance: A white-box approach based on process models, Journal of Software: Evolution and Process 31 (2019) e2170.
[39] Y. Meidan, B. Lerner, G. Rabinowitz, M. Hassoun, Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining, IEEE transactions on semiconductor manufacturing 24 (2011) 237–248.
[40] A. C. Choueiri, D. M. V. Sato, E. E. Scalabrin, E. A. P. Santos, An extended model for remaining time prediction in manufacturing systems using process mining, Journal of Manufacturing Systems 56 (2020) 188–201. doi:https://doi.org/10.1016/j.jmsy.2020.06.003.
[41] S. Nguyen, M. Zhang, M. Johnston, K. C. Tan, Genetic programming for evolving due-date assignment models in job shop environments, Evolutionary computation 22 (2014) 105–138.
[42] M. Li, L. Yao, J. Yang, Z. Wang, Due date assignment and dynamic scheduling of one-of-a-kind assembly production with uncertain processing time, International Journal of Computer Integrated Manufacturing 28 (2015) 616–627.
[43] C. Wang, P. Jiang, Deep neural networks based order completion time prediction by using real-time job shop rfid data, Journal of Intelligent Manufacturing 30 (2019) 1303–1318.
[44] M. Gansterer, Aggregate planning and forecasting in make-to-order production systems, International journal of production economics 170 (2015) 521–528.
[45] T. Chen, Y.-C. Wang, et al., An iterative procedure for optimizing the performance of the fuzzy-neural job cycle time estimation approach in a wafer fabrication factory, Mathematical Problems in Engineering 2013 (2013).
[46] J. Wang, J. Zhang, Big data analytics for forecasting cycle time in semiconductor wafer fabrication system, International Journal of Production Research 54 (2016) 7231–7244.
[47] E. Ruschel, E. d. F. R. Loures, E. A. P. Santos, Performance analysis and time prediction in manufacturing systems, Computers & Industrial Engineering 151 (2021) 106972.
[48] R. J. Kurscheidt, E. A. Santos, E. de FR Loures, J. E. Pecora Jr, J. M. Cestari, A methodology for discovering bayesian networks based on process mining, in: IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE), 2015, p. 2322.
[49] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, A. Calinescu, Digital twins: State of the art theory and practice, challenges, and open research questions, Journal of Industrial Information Integration (2022) 100383.
[50] J. Tang, Y. Liu, K. yi Lin, L. Li, Process bottlenecks identification and its root cause analysis using fusion-based clustering and knowledge graph, Advanced Engineering Informatics 55 (2023) 101862. URL: https://www.sciencedirect.com/science/article/pii/S1474034622003202. doi:https://doi.org/10.1016/j.aei.2022.101862.
[51] M. Tranmer, J. Murphy, M. Elliot, M. Pampaka, Multiple linear regression (2nd edition), Cathie Marsh Institute Working Paper 2020-01 (2001). doi:https://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/working-papers/2020/2020-1-multiple-linear-regression.pdf.
[52] L. Breiman, Random forests, Kluwer Academic Publishers (2001). doi:https://doi.org/10.1023/A:1010933404324.
[53] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, V. N. Vapnik, Support vector regression machines, in: NIPS, 1, 1996, p. 1. URL: https://api.semanticscholar.org/CorpusID:743542.
[54] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1967) 21–27. doi:10.1109/TIT.1967.1053964.
[55] P. Refaeilzadeh, L. Tang, H. Liu, Cross-Validation, Springer US, Boston, MA, 2009, pp. 532–538. doi:10.1007/978-0-387-39940-9_565.
[56] M. Stone, Cross-Validatory Choice and Assessment of Statistical Predictions, Journal of the Royal Statistical Society: Series B (Methodological) 36 (2018) 111–133. doi:10.1111/j.2517-6161.1974.tb00994.x.
[57] J. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: Empirical comparisons, International Journal of Forecasting 8 (1992) 69–80. doi:https://doi.org/10.1016/0169-2070(92)90008-W.
[58] C. V. Goldman, M. Baltaxe, D. Chakraborty, J. Arinez, C. E. Diaz, Interpreting learning models in manufacturing processes: Towards explainable ai methods to improve trust in classifier predictions, Journal of Industrial Information Integration 33 (2023) 100439.

4 HOW CAN A CONTEXT-BASED CLUSTERING OF DRIVERS HELP INCREASE FUEL EFFICIENCY?

Connected vehicles with vehicle-to-cloud (V2C) connections are vehicles that have the capability of sending and receiving information from a server. This information can be miscellaneous and so can be their applications. We wrote this paper to demonstrate how this data can generate value for both the vehicle's manufacturer and drivers.

An important aspect of a vehicle, especially for the drivers, is the vehicle's fuel efficiency. Improvements in fuel efficiency values mean lower cost and environmental impact. In this paper, we propose an indirect and passive way to increase fuel efficiency by using data from connected vehicles' trips.

Humans are naturally competitive. Our proposition in this paper is to use this competitiveness by relating it to fuel efficiency. For this, fuel efficiency rankings of the drivers are proposed. However, given the fact that the drivers can be in different environments on each trip, something that affects fuel efficiency, it would not be fair to insert all trips in the same ranking.

To solve this problem, a context-based clustering is presented. This clustering can identify and separate trips into different environments based on context-related values such as distance, time, and speed. With this information it is possible to create individual rankings for each context, making fair fuel efficiency comparisons.

Even though the presented context-based clustering was projected to build fair fuel efficiency rankings, we also present other possible applications. One is particularly interesting, as it served as a spark for the research of the next paper.

This paper was submitted to the journal "Transportation Research Part D: Transport and Environment" and, until the writing of this, is under review.

How Can a Context-Based Clustering of Drivers Help Increase Fuel Efficiency?

João Gabriel Santin Botelho[a,*], Eduardo Alves Portela Santos[a], José Eduardo Pécora Junior[a], Rodrigo Balani[b], Janaine Rodrigues[b]

[a] *Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE), Universidade Federal do Paraná, Evaristo F. Ferreira da Costa 408, Curitiba, Paraná, 81530-015, Brazil*
[b] *Renault do Brasil, Avenida Renault 1300, São José dos Pinhais, Paraná, 83070-900, Brazil*

Abstract

The automotive industry is evolving, and with the popularization of big data it is natural that the production of connected vehicles is growing. Among the many ways to extract value from vehicles' data is the identification of the drivers' environment, e.g., in which context they can be inserted, which influences fuel efficiency. In this paper, we present a study on the clustering of vehicles, based on their context and without using highly sensitive GPS data, to built unbiased fuel efficiency rankings using data from vehicles from all over Europe. We propose a framework for the clustering of the data before a fuel efficiency ranking may be applied. K-means models were tested and trained with historical data to cluster new unseen data and validated using available fuel consumption data. Also, we show how our method can be used for analyzing fleet behavior, improving recommendation systems, and assisting in product development.

Keywords: Connected vehicles, Driving context, Fuel efficiency, Clustering, k-means

1. Introduction

More than ever, the industry is shifting towards data-driven solutions and adapting to new technologies [1]. Nowadays, collecting and using vast amounts of data is already a reality for most medium to big-size companies. This growing accumulation of vastly different types of data is driven by the ever-growing possibilities in the generation of value that the advancements in data processing and machine learning techniques are creating [2].

In the automotive industry, data collection is becoming a common trend [3] as more connected vehicles enter the market each year [4]. A connected vehicle, precisely a vehicle with V2C (vehicle to cloud) connection [5], is a cellular-enabled vehicle that connects to a central server, enabling the receiving and sending of data using the mobile network [6]. The increase in the development and production of connected vehicles is a direct consequence of the many benefits it can bring to the drivers [7] and the many ways it can generate value for the vehicle's manufacturer [8].

Easy monitoring of the vehicle's conditions, receiving reports of the latest trips, controlling many of the vehicle's features, and vehicle software updates with OTA (over-the-air) updates. These are some benefits a V2C connection enables, being easily accessible to the driver through a smartphone app [5]. When well implemented, these features often mean a later gain to the manufacturer, as they increase customer satisfaction. A shorter-term gain a manufacturer has with connected vehicles is the value the data collected from it can generate. Aiming to decrease production costs and increase sales, a big part of vehicle development is identifying features that can be excluded and should be included in new models. Gathering and analyzing connected vehicles' data can easily replace most surveys performed to identify those features [9]. Furthermore, this data is often more reliable than the answers from survey participants as they contain the ground truth of how the vehicles are being used. Moreover, with the increased amount and quality

*Corresponding author
Email address: joaobotelho@ufpr.br (João Gabriel Santin Botelho)

of data comes the question of what is important enough to be the focus of analyses. One of those topics is definitively fuel consumption, given its importance inside and outside of the automotive industry [10].

When our focus is fuel consumption, the objective immediately turns to increasing fuel efficiency and, consequently, emitting less CO_2 by driven kilometer. One way of achieving that is by comparing drivers. The competitiveness born by the comparison of the drivers' fuel efficiencies and the gamification of the act of driving can be an efficient way to improve fuel efficiencies passively [11, 12, 13, 14] and, consequently, decrease overall CO_2 emissions. This comparison could be made with fuel efficiency rankings between drivers, constructed to avoid inserting any bias. To build unbiased fuel efficiency rankings, we must consider that the fuel efficiency of a vehicle is highly affected by different factors [15] such as vehicle model, type of fuel, drivers' behavior, road quality, vehicle maintenance, and many others. However, in this paper, the focus is the identification of the context, type of road in which the vehicle was driven, a factor that also seriously impacts fuel efficiency, and without using highly sensitive positional data such as GPS data, it is not as easily identifiable as the vehicle's model or type of fuel. Independent of the drivers' behavior and how economical or wasteful they are, vehicles on highways show very different fuel efficiencies than vehicles in city centers. Thus, removing most environment-induced biases, such as speed, space, and time limitations, is paramount before a fuel efficiency comparison can be made.

From this fact, the primary motivation of this research arises: how can an unbiased fuel efficiency ranking be built? We tackle this problem by clustering the drivers based on contextual data, i.e., data mainly describing where the vehicle has been driven without using highly sensitive positional information. By clustering the drivers according to their contexts, the biases that were different for the whole group get nearly the same for individuals from the same cluster, therefore removing a bias by creating groups where everyone has the same bias. Thus, comparisons between individuals with the same bias can be considered fair, given that the same environment-induced limitations affect all drivers; a fuel efficiency ranking for each context would be fair. Four clustering methods were tested and considered: the $k-$means++, the hierarchical agglomerative, the DBSCAN, and the spectral. Ultimately, the $k-$means++ was chosen due to its capabilities of clustering a massive volume of data and data unknown by the clustering models.

The data used in this research was generated by real connected vehicles with V2C connections. The type of data received and stored in the data lake is tabular, where each observation, row, has the aggregated data from one trip, and a trip is defined by the turn on and off of the vehicle. Therefore, we will be clustering the trips sent by the vehicles, and with a group of trips, it is possible to identify the drivers' use patterns and the context in which the vehicles are mainly driven. By being included in a group with the same context, i.e., bias, the drivers can be fairly ranked within these groups based solely on fuel efficiency.

The main contribution of the context-based clustering we will expose in this study is the construction of fair fuel efficiency rankings. However, other significant uses of the method are presented here. Some of the other meaningful applications of the proposed method we will show in this paper include using this clustering to analyze the drivers' behavior and in product development. We demonstrate how our method can be used in a macro drivers' behavior analysis by analyzing a whole fleet to identify the drivers' patterns. Also, we present a possibility of application in a micro driver's behavior analysis with the analysis of each vehicle to improve individual recommendation systems. The possible applications of the method in product development can be broad. However, we discuss two concrete examples: the segregation of interest groups in electric vehicles (EVs) battery size analysis and the discovery of features that should be removed or added in new vehicles.

The remainder of this paper is organized as follows: Section 2 discusses previous studies related to the topics presented in this paper. Section 3 presents our framework for clustering the trips, explaining data treatment, the used clustering methods and metrics, and the models selection and validation methodology. Section 4 validates the framework and the chosen features using available fuel consumption data, showing the results of applying it to real data. Section 5 presents the results obtained and some real-world applications. Lastly, Section 6 presents the study's conclusions and possible follow-up research related to the studied topic and reached results.

2. Related Works

In the field of the applications of vehicle connectivity, [5] presents and discusses various examples of different frameworks. This paper does not only focus on V2C connection but also on V2N (vehicle to network) and V2I (vehicle to infrastructure). It presents many applications of the different types of vehicle connectivity, ranging from as

simple as controlling vehicle features using a smartphone app to as complex as optimizing battery degradation cycles of electric buses [16].

Connectivity to create better methods is shown in [17]. In this paper, a V2C connection is used in conjunction with a method to manage electric vehicles' battery temperature in low-temperature environments. Their battery thermal management strategy is based on a non-linear model that utilizes predictions of the vehicle's future speed based on its path given via a V2C connection. Another example of performance gain from using a V2C connection is in [18]. In it, an energy management strategy (EMS) for hybrid buses is proposed, which uses driving pattern recognition and V2C connectivity. For the driving pattern recognition, the *k*-means algorithm is applied to separate 7208 power profiles into 16 clusters, and for each cluster, a heuristic rule is extracted. The driving pattern recognition algorithm is executed online with traffic information provided through V2C connectivity to select an off-line-trained optimal rule.

Although connectivity undoubtedly has many benefits, the collection of data can be a sensitive subject. In [19] this issue is tackled by the application of Human-Data Interaction (HDI) framework, conducting semi-structured interviews with 15 drivers. These interviews revealed issues related to the drivers understanding of car data and their control over it. This study brings light to the importance of data privacy on connected vehicles and how manufactures could improve their costumers awareness of the data they are sharing and what benefits their data can generate. The study in [20] explores the benefits that vehicle connectivity, the sending and receiving of data by a vehicle, could have in society. Three scenarios are presented and analysed in this study: Mobility for All; Mobility in Transition; and Fragmented Mobility. They conclude that the most significant societal benefits of connected vehicles come from its interactions with automation and electrification and that any push towards CV represent a low-regrets options.

A remarkably complete and thoughtful paper on clustering algorithms for vehicular data is [21]. This study reviews and summarizes 20 clustering techniques and various clustering metrics. This survey paper separates the clustering methods into groups according to their working principles. The authors explain each method, their advantages and disadvantages, and the scenarios where they are most advantageous. This paper was very helpful in selecting the clustering algorithms used in our research.

An extensive study on the value of vehicular data clustering can be found in [22]. In this study, the author writes an extensive introduction to data analysis and the evolution of data usage in the automotive industry context. Although the study focuses on trucks' historical data, its objective is similar to ours: the demonstrations of how the clustering of vehicular data can be a powerful tool for data analysis and how it can help manufacturers improve the development of their products. Another study on clustering can be found in [23], where there is a deep analysis of high-dimensional data based on driving styles. This study tests dimension reduction techniques coupled with clustering algorithms to identify the best combination of these techniques for clustering different driving styles. Another study on using vehicle data to cluster drivers can also be found in [24], where the authors categorize the drivers' behavior, focusing on drivers' aggressiveness. Although the features and data structure are not the same as in our study, as more detailed telematic data is used, the clustering is also performed using the *k*-means algorithm with *k*-means++ initiation.

Another interesting work on clustering related to vehicular data is [25]. With the objective of understanding road crashes, this paper investigates the optimal number of driving profiles with the most important characteristics to differentiate drivers. Two algorithms are used, the *k*-means and the OPTICS, and using drivers behaviour characteristics such as number of speeding, headway and harsh events per 100 km, three driving profiles were discovered, dividing the drivers into less risky, modest, and aggressive. Another paper that uses the *k*-means to identify driving styles is [26]. This study has the objective of using Basic Safety Messages (BSMs) generated by connected vehicles to quantify at each instant driving behavior and classify driving styles in different spatial contexts. Similar to the previous paper, the driving styles are divided into three groups: aggressive; normal; and calm. They show that the spatial contexts vary the proportions of each group, demonstrating that the environment affects the drivers' behavior.

A paper that partially uses clustering for fuel consumption reduction is in [27]. This paper uses a spectral clustering method to cluster fuel consumption groups based on speed and acceleration. Combined with environmental and behavioral data, this clustering information is then used to train a deep-learning network to predict fuel consumption levels. In [28] the impact of driving behavior in fuel consumption is analysed. This study focus on the driving styles in work-zones and curves and demonstrates that aggressive driving generates an increase on fuel consumption compared to normal driving. Another paper related to fuel, or energy as this case is for electric vehicles (EVs), is [29]. In this study different styles of charging a EV are identified via a clustering analysis over the data of 994 respondents across Australia. They were able to divide the drivers into five charging styles, which demonstrated that a uniform approach to EV-related policies is not appropriate, both by governments or as market strategies by manufactures.

3. Proposed Framework

In this section, we expose our proposed framework and explain each of its steps, beginning with the vehicles sending data to the cloud and ending with the construction of the fair fuel efficiency rankings or the other possible applications of our clustering method.

In Figure 1, we have a scheme of the proposed framework. The framework shows the big picture, from the vehicle to the clustered data applications, but our method is inserted in three major steps:

1. It starts in the data lake with the extraction of the historical data of the vehicles' trips and continues with the subsequent treatment, filtering, features selection and generation, and transformation of this data;
2. With the treated historical data, clustering models are tested, and the best models are trained with all the treated historical data and kept for use;
3. With the trained clustering models and the trained data rescaling information, new unseen data can be extracted from the data lake and be clustered by the models.

With the clustered data, many applications are possible; Figure 1 lists just some of them. The main motivation for this work is the construction of fair fuel efficiency rankings. Nonetheless, the presented method could also be used for other purposes, such as analyzing a fleet's behavior, improving recommendation systems by adding the clusters' information, and helping with product development by finding interest groups in the clustered data.

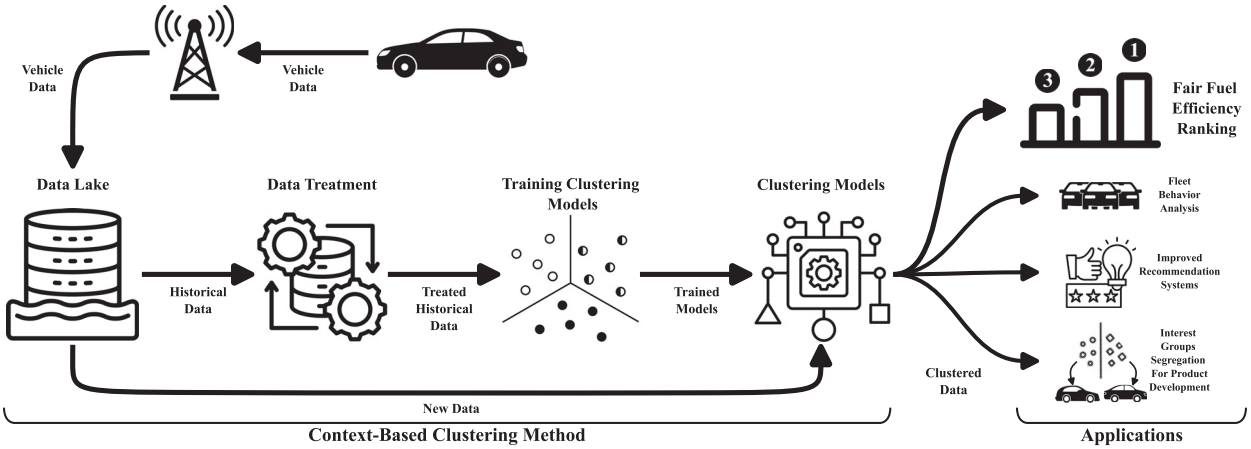

Figure 1: Graphical representation of the framework.

3.1. Data Treatment

In this study, we use data from connected vehicles from all over Europe of a French automotive company, from which the drivers permitted their data to be collected by V2C connection. Table 1 is an example of how this data is structured. It is possible to notice that the data collected is organized in tables, where each row contains information regarding one trip of a vehicle. The collected information is organized in different columns and can be divided into trip-related and vehicle-related. Trip-related information are the ones that change depending on the trip, like the time of start and end of the trip, distance traveled, duration the vehicle stayed in a specific velocity or RPM interval, average acceleration, brake pressure and external temperature, fuel consumption, and many more. Alternatively, vehicle-related information are inherent to the vehicle, like the hashed vehicle identification number (VIN), model, motor, transmission, power, battery size, features of the vehicle, and many more.

Before treating the data, we start by selecting the columns that will be used for the clustering. In this feature selection, we consider that we have data from different vehicle models, so we need features that would enable a previous separation of different vehicle models. We also have to consider that the objective of the clustering is context identification, so features that describe where the vehicle has been driven are needed. Another critical point to consider

is the reliability of the columns, i.e., their percentage of null values. If the trip has null values on used features, the clustering models cannot insert it into a cluster. By taking into account all these objectives, we chose the following columns:

- VIN hashed, to be able to identify the different trips that were created by different vehicles;
- Model, transmission, power, and battery size, to, before clustering by the context, manually separate different types of vehicles;
- Distance traveled;
- Start and end time of the trips, to be able to calculate the trips' total time and, with the distance traveled information, calculate the average speed;
- Duration the vehicle stayed in specific velocity intervals.

The duration the vehicle stayed in specific velocity intervals will be called "Speed i", where "i" is a number, and the bigger it is, the higher the speed interval. Due to their unmanageable percentage of null values, many useful features were not included. From these features, the average acceleration and brake pressure features were the most significant losses due to value nullity.

VIN hashed	Start Time	Finish Time	Distance (km)	Time (s) at 0 km/h	···	Model	Transmission	Power	Bat. Size
76973	01/01/2023 12:00:00	01/01/2023 12:30:00	10	60	···	A	M	P1	
92928	01/01/2023 12:20:00	01/01/2023 13:50:00	111	150	···	B	E	P1	B1
85349	01/01/2023 12:30:00	01/01/2023 12:50:00	8	50	···	A	M	P1	
36348	01/01/2023 12:35:00	01/01/2023 12:40:00	2	30	···	A	A	P2	
54526	01/01/2023 12:37:00	01/01/2023 12:47:00	6	30	···	A	A	P2	
76759	01/01/2023 12:40:00	01/01/2023 13:05:00	12	120	···	B	E	P1	B1
23915	01/01/2023 12:42:00	01/01/2023 13:22:00	24	230	···	B	E	P1	B1
31261	01/01/2023 12:47:00	01/01/2023 13:02:00	8	60	···	A	A	P2	
46851	01/01/2023 12:50:00	01/01/2023 13:15:00	7	130	···	A	M	P1	

Table 1: Table exemplifying the connected vehicles' dataset.

The treatment of the data starts by extracting it from the data lake. After the extraction, we start by eliminating the problematic rows. We eliminate observations with some anomalies, such as the existence of null values, start time greater than or equal to the end time, odometer value at the start greater than or equal to the odometer value at the end, and many more. Another group of rows that is eliminated is the ones where the distance traveled is smaller than 100 meters, as by testing, we noticed that most of the irregular trips belong to this group. Trips that differ too much on the duration calculated by delta time and time in velocity intervals are also eliminated. Lastly, we eliminate the observations in which the distance traveled is greater than the maximum possible distance given the values of duration in velocity intervals, e.g., if a vehicle stayed for 1 hour in the interval 0 to 100 km/h, its maximum traveled distance should be 100 km.

After excluding problematic data, we create two new columns: total trip time and average speed. Total trip time is the delta, in seconds, between the start and end time of the trip, and average speed, in km/h, is the distance traveled, in kilometers, divided by the total trip time, in hours. Another modification we make to the data is the transformation of the values from the columns of duration in specific speed intervals. Originally, the measurement unit of these columns is seconds. However, as each column represents the duration in a different continuous interval, and the end of each interval is the beginning of another, it is possible to transform the values from seconds to percentages by dividing it by the total time of the trip.

Knowing that one of the objectives of clustering algorithms is to find groups that are not easily found manually and that a huge factor that affects fuel consumption is the vehicle's characteristics, which are easily separable in our

data, we do a manual clustering based on the vehicles' inherent characteristics. In this manual clustering, we cluster the trips based on the vehicle model, transmission, power, and battery size information, forming different groups.

Lastly, having in mind that this data will be used for the training of clustering models and that the algorithm that will be used is the *k*-means [30], a method based on distances, a rescaling of the data is important as to not bias the model towards features with big values. We rescale each group by feature, making them range from 0 to 1, where 0 is the minimum value, and 1 is the maximum value of that feature. The rescaling is performed for all features which originally ranged outside the [0, 1] interval; thus, the "Speed i" columns are not rescaled. Also, the original minimum and maximum values of each rescaled feature for each group are kept for use in the rescaling of the new data, which will be clustered.

Using Table 1 as an example of a dataset to be clustered, Table 2 is an example of this dataset after the creation of the new columns, trip time and average speed, and selection of the columns that would be used in the clustering. Table 3 is an example of the data that would be fed to the clustering algorithms, i.e., after applying all the data treatment, separation, and transformation to the data from Table 2.

VIN hashed	Distance (km)	Trip Time (s)	Avg. Speed (km/h)	Speed 0 (s)	···	Speed 7 (s)	Model	Transmission	Power (kW)	Bat. Size (kWh)
76973	10	1800	20	60	···	0	A	M	P1	
92928	111	5400	74	150	···	300	B	E	P1	B1
85349	8	1200	24	50	···	0	A	M	P1	
36348	2	300	24	30	···	0	A	A	P2	
54526	6	600	36	30	···	0	A	A	P2	
76759	12	1500	28.8	120	···	0	B	E	P1	B1
23915	24	2400	36	230	···	0	B	E	P1	B1
31261	8	900	32	60	···	0	A	A	P2	
46851	7	1500	16.8	130	···	0	A	M	P1	

Table 2: Table exemplify the data selected to be used for the context-based clustering.

Group	Distance (km)	Trip Time (s)	Avg. Speed (km/h)	Speed 0 (s)	···	Speed 7 (s)
A_M_P1	1.0000	1.0000	0.4444	0.0333	···	0.0000
	0.3333	0.0000	1.0000	0.0417	···	0.0000
	0.0000	0.5000	0.0000	0.0867	···	0.0000
A_A_P2	0.0000	0.0000	0.0000	0.1000	···	0.0000
	0.6667	0.5000	1.0000	0.0500	···	0.0000
	1.0000	1.0000	0.6667	0.0667	···	0.0000
B_E_P1_B1	1.0000	1.0000	1.0000	0.0278	···	0.0556
	0.0000	0.0000	0.0000	0.0800	···	0.0000
	0.1386	0.2308	0.1593	0.0958	···	0.0000

Table 3: Table exemplify the data feed to the clustering algorithms.

3.2. Clustering

In this subsection, we explain the main reasons for using the *k*-means clustering method in detriment to the other tested methods: hierarchical agglomerative, the density-based spatial clustering of applications with noise (DBSCANS), and the spectral. Furthermore, we will explain how the treated historical data is used to test, select, and train the *k*-means models and why we used the *k*-means++ initiation. We will present the methods and metrics used to select the best parameters for training the final models.

3.2.1. Clustering Algorithm Selection

There are many different clustering algorithms. These algorithms can vary from how they aggregate observations to how they define clusters. The four clustering methods we tested belong to four classes: Partition, Hierarchical, Density-Based, and Modern clustering [21].

The Partition type method we will consider is the *k*-means [30]. Partition clustering methods are the most common and popular methods of clustering. Their working principle generally consists of using centroids and their relative distance to the observations to indicate to which cluster an observation belongs. The *k*-means basic working principles make it very suitable for large-scale data, as its low time complexities allow for fast training and use of the models.

The Hierarchical type method we will be considering is the hierarchical agglomerative. Hierarchical clustering methods work by creating trees that link their nodes and clusters based on different types of Linkages. These Linkages are the different ways to calculate the similarity between the clusters. Due to their hierarchical working principles, with the methods of this class, it is possible to define the optimal clustering they produce by analyzing their resulting trees, which keep all the clusters in different hierarchies. However, coupled with this advantage comes their main disadvantages: high time complexity and memory usage. The agglomerative, specifically, has a time complexity of $O(n^3)$ and a memory complexity of $O(n^2)$ [31], which makes it unsuitable for large-scale data.

The Density-Based type method we will be considering is the DBSCANS [32]. Density-Based clustering methods are on the more unique end of the clustering methods spectrum, as they are not based on distances but on density instead. They work by aggregating data with similar densities and, in the DBSCANS case, by identifying observations in the core, border, or out of the clusters' area. The main advantage of this class is its capability to cluster arbitrary shapes and identify outliers. One of their characteristic that could be an advantage or a disadvantage is the fact that the number of clusters is defined by the model, depending on its parameters values. Their time complexity and memory usage are lower than those of the hierarchical agglomerative, but they are still not as suitable for large-scale data as the *k*-means.

The Modern type method we will consider is the spectral [33]. Modern clustering methods are also on the unique end of the clustering methods spectrum, varying a lot within the class. They do not have a common working principle, but they all derive from the idea of mixing deep learning with clustering algorithms. The spectral clustering method uses graph theory, regarding sample points as vertices and the similarity between the data as edges. It can deal well with high dimensionality and arbitrary shape data. However, it has a very high time and memory complexity, making it unsuitable for large-scale data.

The data we have to cluster comprises millions of observations and 11 columns. Thus, we have to deal with a lot of data with not as high dimensionality. Another important aspect of our clustering needs is the clustering of unknown observations, as new data arrives daily. Considering the characteristics of our data and exposed methods, our only reliable choice is the *k*-means. The hierarchical agglomerative and spectral memory usage makes clustering our data using them impossible, and the DBSCANS can not manage to cluster our data promptly. Furthermore, the *k*-means is the only method that allows for easy clustering of new data. In contrast, if the other three methods were used, additional classification models would be required to be trained on the clustered data to be able to cluster by classification the new data.

3.2.2. K-means Method

Given that the *k*-means is our chosen method, we will further explain some of its characteristics. The *k*-means is by a far margin the most popular clustering algorithm. While not an exact method, as it is a heuristic for the minimization of within-cluster variance problem, which is an NP-hard problem, and is a little unstable due to its randomness on the initiation of the clustering, which can be diminished by the *k*-means++ initiation, the *k*-means is much faster than other clustering algorithms due to its simplicity. Also, although it is simple, it always converges to a local optima and often yields satisfactory results depending on the cluster's shape.

The *k*-means is an iterative method, needing just one iteration to have a solution but some to reach a convergent solution. By following Figure 2, it is possible to understand how the method works. It starts with *k*, the number of clusters, random points as initial centroids. These centroids work as references for building the clusters, where each point is assigned to its closest centroid in each iteration. At the end of each iteration, the new clusters' centroids are calculated, and the process repeats until a stop criterion is satisfied, which could be if the change from one iteration to another in the positions of the centroids or in the composition of the clusters is less than a tolerance value.

The *k*-means++ initiation method was created to cover the *k*-means flaw of sometimes converging to bad solutions, which usually is a consequence of poor initial centroids choice. This initiation method is an improved method for obtaining initial centroids. It randomly chooses a point from the data as the first centroid. Then, for each data point, the distance between it and its nearest centroid is calculated. The next centroid is chosen following a weighted probability distribution with the square of the previously calculated distances as the weights. This process is repeated until *k* centroids have been obtained. Figure 3 demonstrates how the method works. The size of the different points represents the probability of being chosen as new centroids.

Even though this method of generating initial centroids takes more time than choosing at random, the *k*-means with it is proven to converge much faster and to better local optima [34], ultimately making it faster and better than with the normal initiation.

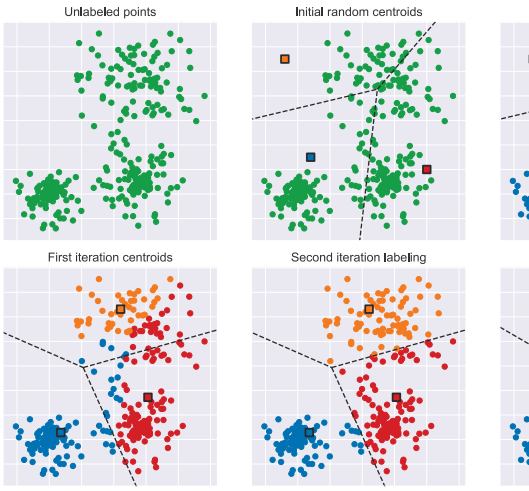

Figure 2: Illustration of how the *k*-means algorithm works on two dimensional data. The dashed lines represent the boundaries created by the centroids.

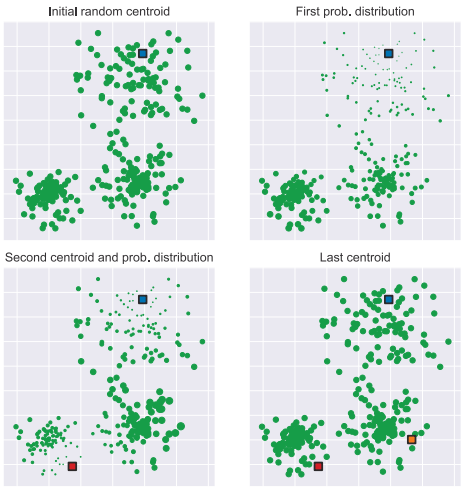

Figure 3: Illustration of how the *k*-means++ algorithm works on two dimensional data.

3.2.3. Model Selection Method

With the chosen clustering algorithm exposed, we will present the selected metrics to assist in choosing the number of clusters of the *k*-means and measure the clusters' fitness. The chosen metrics are the silhouette score [35] and the Davies-Bouldin index [36], as both measure the fitness of the data points to their cluster compared to the other clusters, not needing label values, which do not exist in our case.

The silhouette score measures how similar a point is to its cluster compared to the others. This metric takes two measures for each point: the mean distance to its cluster and the smallest mean distance to any other cluster. The silhouette score of one data point is obtained by subtracting the latest from the first and dividing it by the maximum between them. This method yields a score that ranges from -1 to 1, where a high value indicates high proximity/similarity with its cluster and lower to the other clusters, and lower values indicate the opposite. In our case, we used the average silhouette score from all clustered points as the fitness measurement.

Unlike the silhouette score, the Davies-Bouldin index (DBI) measures the clustering as a whole. It measures the separation between the clusters and variation within the clusters. It works by measuring the average distance between the points and the centroids in a cluster and measuring the distance between all the centroids. Using these values, it compares the clusters by pairs, adding the intra-cluster mean distances and dividing by the distance between the centroids. The DBI is the average of the maximums of this pair-wise comparison. Consequently, the DBI is a non-negative number, where a low value indicates high separation between clusters and low variation within clusters.

To ensure that our final models are optimal for the clustering of each data group, we will introduce a method for searching for the best number of clusters *k* for each *k*-means model. This search is based on a train and test division to ensure no over-fitting, as the chosen models will be used to cluster new observations.

Our searching method tests k varying from 2 to 15 using a 10-fold cross-validation with five variations of initiation seeds for each fold. The 10-fold cross-validation is our countermeasure against over-fitting. The division of each data group into 10 samples: 9 for training and 1 for testing and the repetition of this process of varying the train and test samples ensures that the models do not get biased by the training data. The seed variation is a countermeasure for when the k-means converge to bad solutions. Figure 6 and the following explains how the method works:

1. For each group of historical data to be clustered, divide it into 10 samples of the same size without repetition. These are the samples that will be used in the cross-validation, taking turns being data for test and training;
2. For k varying from 2 to 15, do the following:
   2.1. Each fold is defined by 1 sample for the fitness test, different from the other folds, and the remaining 9 samples for the training of the models;
   2.2. For each of the 10 folds, train and test five k-means models using different initiation seeds and measure the fitness of their test sample;
3. For each group, find the k which yields, on average, over the 10-folds and the five initiation seeds, the maximum silhouette, and the minimum DBI.

The described method tests 700 k-means models for each group.

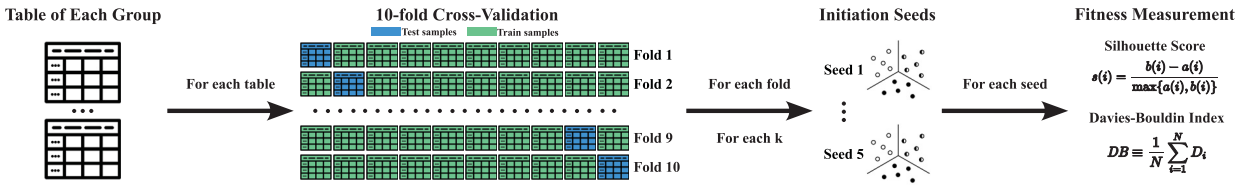

Figure 4: Graphical representation of the method for searching the optimal number of clusters k.

After the best k is found, the final k-means model for each group is created by using all historical data for training and varying the initiation seed ten times to increase the probability of finding the best models. These models are then saved to be used in conjunction with their rescaling data to cluster new data that arrives daily in the data lake.

4. Case Study

This section will present the previously shown framework applied to real-world data. We will show the clustering of real data collected from vehicles of a French automotive company scattered all over Europe and what exactly is the context in which the vehicles are clustered. Also, we will use available consumption data to demonstrate the validity of the chosen features. In this validation, we will demonstrate the influence of the context on fuel efficiency and how raking without clustering based on context would be biased and unfair. This unfairness comes from comparing drivers' fuel efficiencies without considering that their efficiencies are also affected by their contexts, not just their driving styles.

4.1. Vehicles' Data & Clusters' Nomenclature

The data used to find and train the best models were from trips made from June 2022 to December 2023. The data from this period yielded more than 12 million trips, which were reduced to 10 million after applying the data treatment presented in Section 3.1. Because around 7 million of these 10 million trips were from just one vehicle group, a sampling of 2 million trips was performed in this anomalously big group, eliminating 5 million trips. In the end, the data used had a total of 5 million trips divided into eight groups. In those groups, there were two vehicle models, one hybrid and one electric, which we will call "A" and "B", respectively. The hybrid model has five variants, varying in hybridization level, transmission, and power. The electric has three variants, varying in power and battery size. From here on out, to facilitate mentioning the vehicle groups, we will call them "model_power_transmission_battery-size". For example, the group of model "A" with power "P1" and transmission "A" is called "A_P1_A".

The base of our clustering is context, as the features that define them are based on distance, speed, and trip duration. Something that is also defined by these features is the existing different types of roads. Thus, the names we gave to the clusters are from the road hierarchy, specifically, the hierarchy also defined by [37]. What is represented by the clusters is the type of road the vehicle was on that trip, or at least was in for most of the trip, as it is highly improbable that a vehicle was driven on only one type of road. Our hierarchy divides the roads into four main categories: highway, arterial, collector, and local. This division is mainly based on three axes: speed limit, traffic flow, and accessibility to property. Figure 5 illustrates how each of the four main categories is placed in the space of these three axes, and complementary to this figure, these categories would be described as follows:

- Highway: very high speed limits and traffic flow but very low accessibility to property. They are built as connections between cities and detours from cities' centers;
- Arterial: high speed limits and traffic flow but low accessibility to property. They are built to enable fast transit inside the cities and as connections between the cities and the highways;
- Collector: moderate traffic flow, speed limits, and accessibility to property. They are built to connect local roads to more busy roads, working as distribution type of roads;
- Local: high accessibility to property but low speed limits and traffic flow. They are built to create easy access to properties.

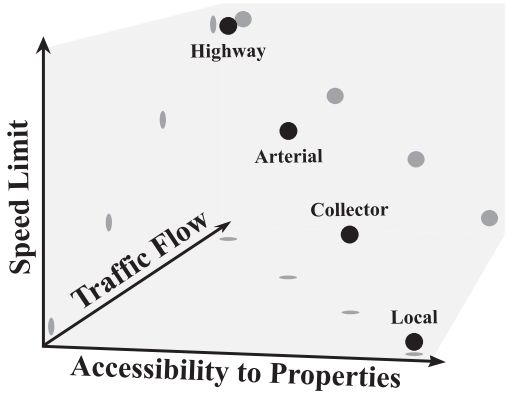

Figure 5: Illustration of the three axes in the road's hierarchy.

Complementary to these four primary categories, we established subcategories for clusters that could be classified in the same hierarchy. The main subcategories are: local-center and local-outskirts, which are subdivisions of local roads. By clustering the data, this subdivision gets clearer, as seen in Figures 7 and 8. In these figures, both local-center and local-outskirts have similar, very low average speeds but slightly different distances and total times. However, the main feature differentiating them is the "Speed 0" feature, which is the percentage of the trips' duration in which the vehicles were stationary. From this, we called the cluster stationary for the most time local-center, as in city centers, there are more traffic lights and traffic in general. In contrast, we call local-outskirts the one which does not stay stationary as much, as in the outskirts, traffic lights are more uncommon and traffic is lower.

4.2. Clustering Models

By following the optimization method for the k-means models illustrated by Figure 4, we obtained the results shown in Figure 6. This Figure presents the results from only two groups, but the same process was performed for all eight groups. Using this type of graph, we were able to choose the best number of clusters k based on maximum silhouette score and minimum Davies-Bouldin Index (DBI). In these two specific groups, it is possible to notice, for the silhouette score, a prominent peak around $k = 5$ for the "A" model and $k = 6$ for the "B" model. For the DBI, the reverse is true, as a big valley is noticeable for the same ks, which indicates that the optimal number of clusters is 5 for the "A" and 6 for the "B" model. Using this method for each group, we created one k-means clustering model

for each group. Interestingly, $k = 5$ was the optimum number of clusters for the five groups of the "A" model, and $k = 6$ was the optimum number of clusters for the three groups of the "B" model. As model "A" is hybrid and "B" is electric, this coincidence in an optimal number of clusters indicates that the a priori separation of those groups was a good decision.

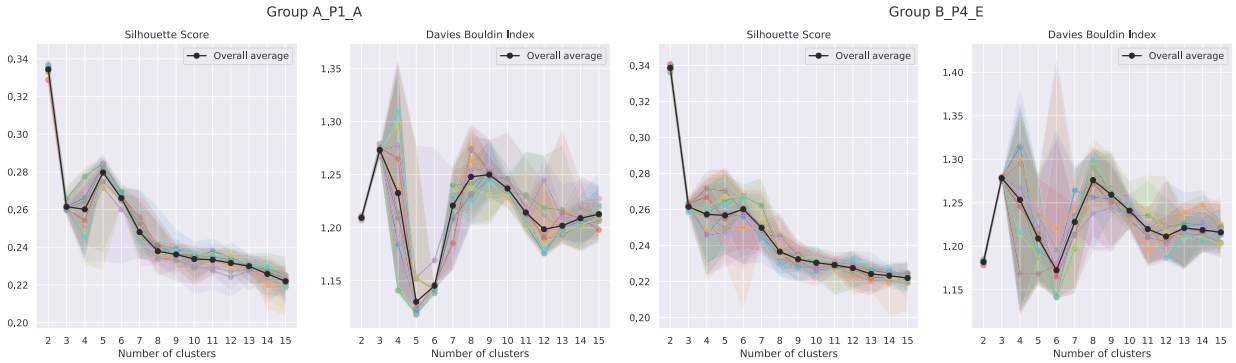

Figure 6: Results of the optimization method for the k-means models applied to two different groups.

Using the number of clusters we found, we were able to train the final k-means models. Figure 7 is an example of how the clustering models separate the data. In Figure 7, we use the three main features, distance, average speed, and total time, to visualize the separations created by the model of the group "A_P1_A". The cluster's segregation is well defined in all dispersion graphs, with amalgamation between clusters occurring just in boundary regions. The distance and total time variables are in the log scale because most trips are agglomerated in a small distance and time range, making the visualization in a linear scale much more difficult.

The average speed separation in the clusters is very linear, as shown in Figure's 7 dispersion graphs where the average speed is in the y-axis. This linearity is also expressed by the influence of the speed on the other variables, which can be seen in the distance by the total time graph in Figure 7, where diagonal lines separate the clusters very well. These lines reveal the existing positive linear relation between trip distance and duration: smaller trips are shorter, and longer trips are lengthier. In graphs where both axes are in the logarithm scale, changes in the slope, a, of a linear function $f(x) = ax + b$ translates more to a shift in the x-axis than to the expected change in line angle. Thus, the strip-shaped clusters in this graph confirm the direct influence of speed on trip distance and duration. The lines formed by the clusters show how, in contexts with low speeds, a slight change in distance translates to a big change in trip time, bigger a, while in contexts with high speeds, the same change in distance translates to a smaller change in trip time, smaller a.

It is noticeable how the separation of the average speed feature is more well-defined than the other two features. This fact is probably due to the use of the duration in speed interval features, which we called "Speed i". Figure 8 shows well how these speed-related features affected the model's decision. The combination of these two figures, 7 and 8, make our nomenclature choice for the clusters much more apprehensible. They demonstrate how, for example, the trips in the "Highway" cluster were clearly driven in a context where very elevated speeds can be achieved, and trips are generally longer and lengthier. This example holds for the other cluster names, roads categories, which can be easily distinguishable by looking at the average speed values in Figure 7 and "Speed i" values in Figure 8.

In statistics, correlation is any statistical relation between variables. In the case of Figure 8, the correlations in the heat map are Pearson correlation coefficients [38]. These correlations indicate the degree of linear relation between the features, with values close to -1 indicating strong negative linear relation, values close to 0 indicating no linear relation, and values close to 1 indicating strong positive linear relation.

The k-means is a purely distance-based method, meaning there is no rationale behind its clustering decisions, just distances to centroids. This fact makes it difficult to rationally justify why a particular trip was clustered in a specific cluster. The correlations in Figure 8 can help with that and, at least, demonstrate the logic behind the models' decisions. From these correlations, it is possible to understand the degree of importance the models give to each feature when classifying the trips in each cluster. If the correlation is high, most trips in that cluster have a high value for that feature, and the opposite is also true. However, if the correlation is zero, the feature is unimportant to that

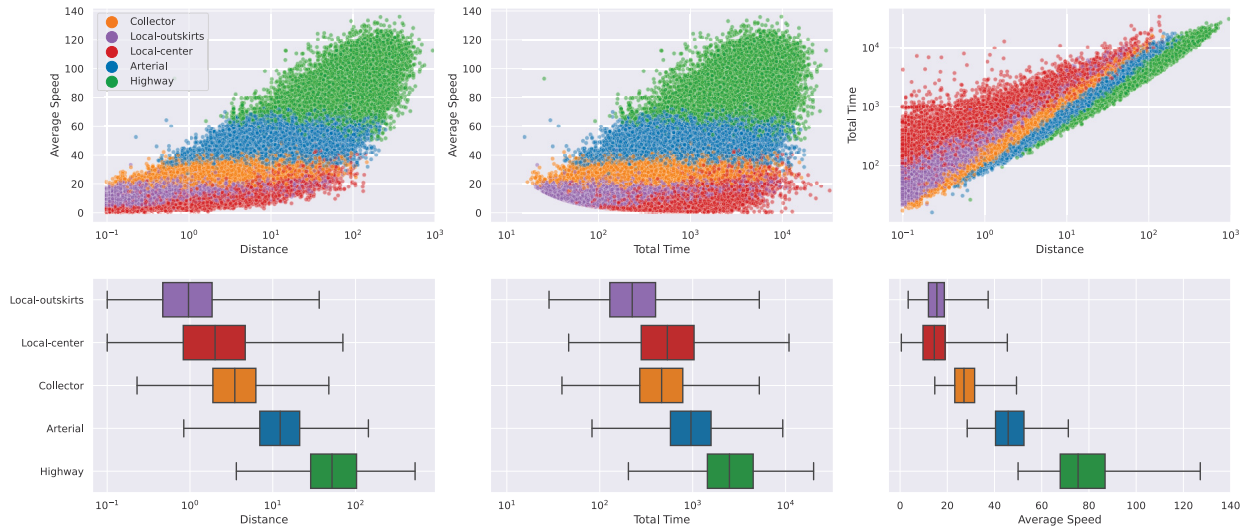

Figure 7: Visualization of the clustering of group's "A_P1_A" vehicles and the cluster's box plot distributions for the main features.

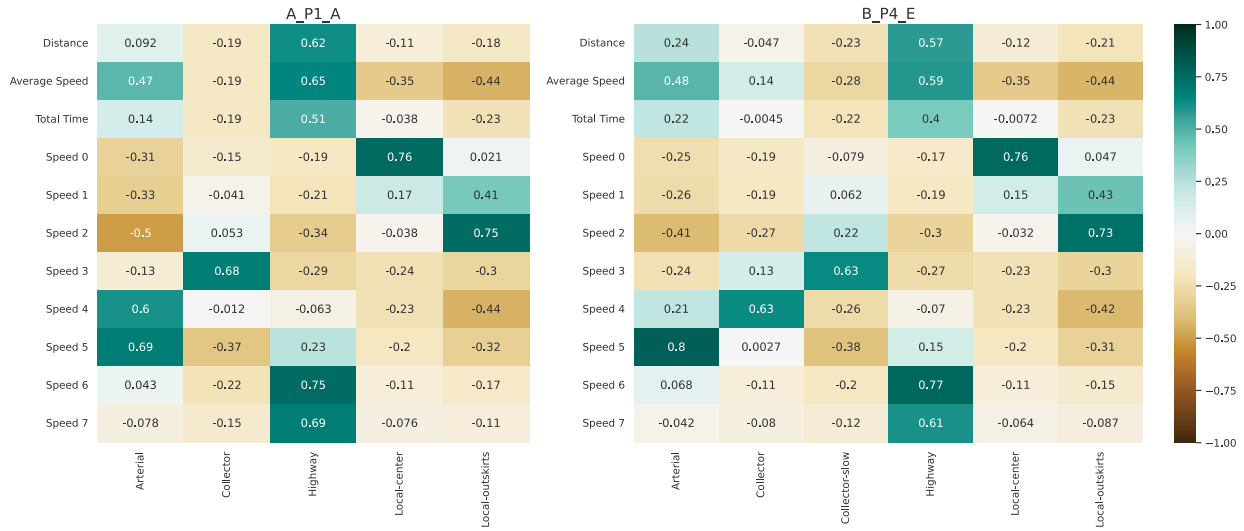

Figure 8: Heat map of the correlation between the features and the clusters for two groups of vehicles.

cluster. For example, a trip with high values in the features distance, average speed, total time, and "Speed"s 5 to 7, with low values in the features "Speed"s 0 to 3, and whatever value in feature "Speed 4" would most probably be clustered as "Highway".

In Figure 7, the "Local-center" and "Local-outskirts" clusters are the less well-defined ones, making it difficult to understand why the model separated them. However, their differences become crystal clear with the additional correlation information in Figure 8. The abnormally high correlation between "Local-center" and "Speed 0" indicates that in most of the "Local-center" trips, the vehicles are stationary for a long time. Meanwhile, the correlation is almost non-existent between the "Local-outskirts" and the "Speed 0", meaning there is a similar number of trips with high and low stationary time clustered as "Local-outskirts". This means that even if their average speeds are similar, their speed profiles are very different.

4.3. Clustering Validation

The main objective of our proposed context-based clustering is to enable the construction of unbiased fuel efficiency rankings. This proposition comes from the assumption that vehicles in different contexts also have different fuel efficiencies. In this subsection, we will empirically demonstrate that context plays a significant role in fuel efficiency fluctuation using our clustering models and available fuel consumption data from actual vehicles.

The data we had did not have a proper efficiency column. Therefore, we calculated the efficiency of each trip, in kilometers per liter, by dividing its total distance by the amount of fuel consumed. Also, just three vehicle groups had fuel consumption values: groups "A_P1_M", "A_P1_A" and "A_P3_CVT". The treatment for this data was the same as the one described in subsection 3.1.

A valuable technique to understand how much a feature impacts another is the calculation of the Pearson correlation between them. The usefulness of this correlation analysis was already demonstrated in subsection 4.2 when we used it to understand the clustering models' decisions in Figure 8. Moreover, besides its use for models' explainability, it is also essential in many feature selection methods for machine learning, as in the Correlation-based Feature Selection (CFS) [39]. In Figure 9, we have the correlations between the features used in the clustering models and the efficiencies of three vehicle groups. The correlations in these graphs corroborate our initial hypothesis that the context greatly influences fuel efficiency. Figure 9 clearly distinguishes how each feature affects the efficiency. For example, the features that have the most significant positive influence on fuel efficiency across all three models are "Average Speed" and "Speed"s 4 and 5, and the ones with the most significant negative influence are "Speed"s 0 to 2. In contrast, "Speed"s 6 and 7 have a small positive influence. This indicates that medium to high speeds contribute to elevated fuel efficiency while being stationary or at low speeds contributes to poor fuel efficiency and that very high speeds do not necessarily lead to good fuel efficiency. By analyzing these correlations, it is clear how high speeds and long trips typically indicate high efficiency and low speeds and short trips indicate low efficiency, which already indicates the existence of, at the very least, two distinct groups.

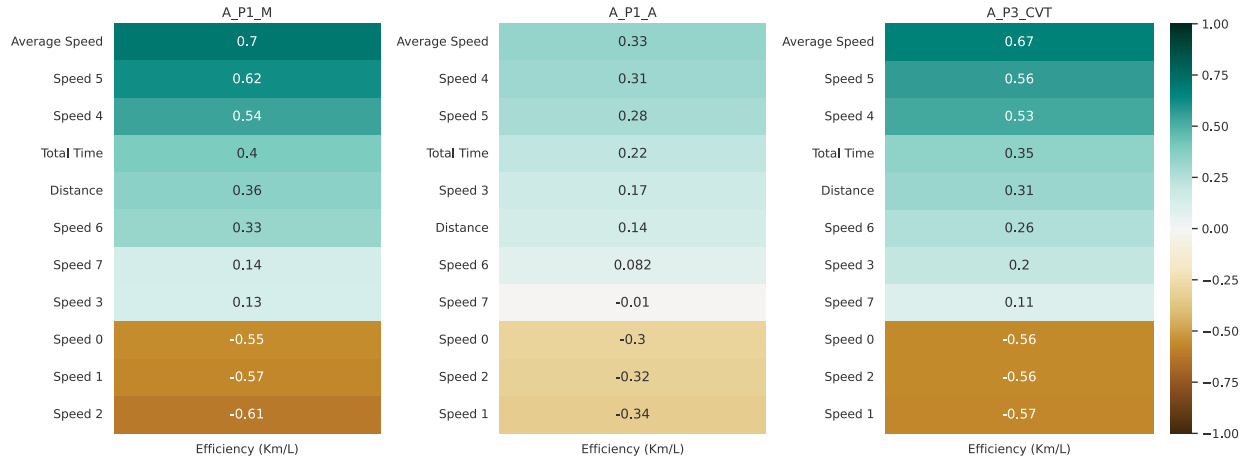

Figure 9: Heat map of the correlation between the used features and the efficiency for three groups of vehicles.

The correlation between features and efficiency is already a good indicator. However, we can have an even better understanding by analyzing the efficiency of each cluster created by the clustering models. Figure 10 has graphs showing how the clusters separate the efficiency data. The dispersion plots give us a better understanding of how the average speed, distance, and total trip time relate to fuel efficiency. All three of the dispersion graphs from Figure 10 show a crescent relation between the x-axis feature and the efficiency, which was already indicated by the positive correlations in Figure 9. Another noticeable feature of these dispersion plots is how the different clusters' regions span through different efficiency ranges, especially in the distance and total time graphs. In these two graphs, it is noticeable how the clusters are less well separated than in the average speed graph, but the segregation is still evident.

Another interesting insight the dispersion graphs from Figure 10 bring is the variation of the fuel efficiency values in each cluster. As seen in all three graphs from this figure, the difference in efficiency variance is highly related to

the average speed, distance, and duration. Trips with low speeds, encompassing both "Local" clusters, have fairly consistent low efficiencies, while trips with high speeds, encompassing the "Highway" cluster, have very consistent high efficiencies. Also, the two clusters with medium speeds have the highest efficiency variance. Another aspect related to efficiency variance that needs to be addressed is the possible errors in measuring the trips' fuel consumption. Some efficiency values shown in Figure 10 are impossible given the vehicles. Values above 35 Km/L are significantly over the analyzed vehicles capabilities, and values under 5 Km/L should not occur unless in unusual situations where the vehicle stays on and stationary for a long time, which would mean a very high trip duration and low distance and average speed, what rarely happens.

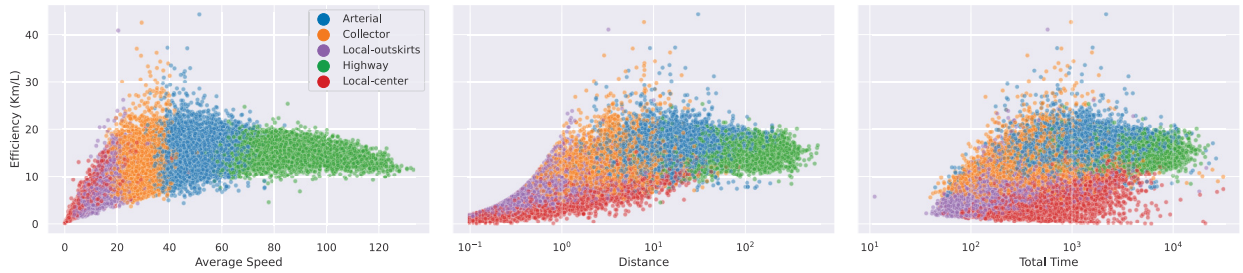

Figure 10: Visualization of the efficiency clustering in relation to average speed, distance and total trip time for the "A_P1_M" group.

The validity, necessity, and importance of the proposed context-based clustering for fuel efficiency rankings are really demonstrated by the box plots in Figure 11. The difference in the distribution of the efficiency data between the clusters for all three groups is evident in these graphs. For example, this difference is more prominent in the clusters "Highway" and "Local-outskirts" from the group "A_P1_M". In this case, it is clear how a fuel efficiency comparison between drivers from these two clusters would be extremely unfair. All drivers from the "Highway" cluster show a better efficiency than 75% of the drivers from the "Local-outskirts" cluster. If we were to compare the average driver from these two clusters, there would be a 7 km/L gap between them, a gap which is not due to the drivers' ability, what fuel efficiency rankings usually try to measure, but by their context, which is out of their control. Once we can cluster based on this uncontrollable characteristic that is the environment, we end this unfairness by building ranks within each cluster, which will compare the ability of each driver with minimum bias from their context.

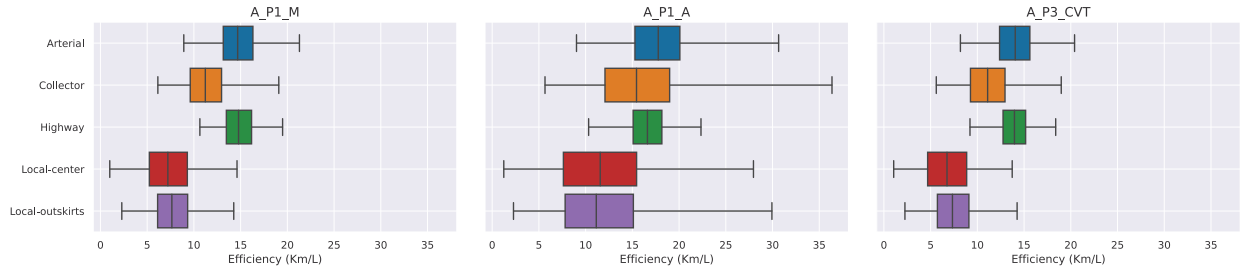

Figure 11: Efficiency box plot distribution of each cluster for three groups of vehicles.

Another important aspect of our clustering method that the box plots from Figure 11 corroborate with is the difference between the fuel efficiency of different groups of vehicles. While groups "A_P1_M" and "A_P3_CVT" are very similar efficiency-wise, group "A_P1_A" is much different, having a much higher overall fuel efficiency. This difference comes from the fact that the vehicles from group "A_P1_A" are full-hybrid, while the ones from the other two groups are mild-hybrid.

5. Results and Discussions

In this section, we will focus on the last part of the framework from Figure 1, where we present the applications and value of the presented context-based clustering. We will discuss how the proposed clustering method can be used

to build different types of unbiased fuel efficiency rankings and how it can be used to gain knowledge on the drivers' behaviors on fleet analysis.

Our initial proposed method to rank the drivers' fuel efficiency using the context-based clustering is done by choosing a periodic time interval and calculating the drivers' trips average efficiencies from different clusters over that time interval. Tables 4 and 5 show the position in weekly and monthly fuel efficiency rankings, one for each cluster, of some drivers from the vehicles from group "A_P1_M". These cluster rankings show the position of the drivers in relation to other drivers that had trips in that cluster and also show the percentage of drivers better than them in terms of fuel efficiency. For example, looking at vehicle #68, we can see that it had no trips clustered as "Arterial" and "Highway" in the week of the weekly rankings, so it is not in the rankings of that week for those clusters, but it is in the other clusters rankings. In the other rankings, it is possible to notice that this vehicle has very low efficiency in its "Collector" and "Local-center" trips, consequently being worse than 76.4% and 73.0% of the drivers from these clusters, respectively. However, in the "Local-outskirts" cluster ranking, it performed better, surpassing 59.5% of the drivers.

Context	**Arterial**		**Collector**		**Highway**		**Local-center**		**Local-outskirts**	
Vehicle	Position	Effi.	Position	Effi.	Position	Effi.	Position	Effi.	Position	Effi.
#129	45 | 54.3%	14.37	57 | 62.9%	10.88	15 | 51.9%	14.72	31 | 40.5%	7.90	4 | 3.8%	10.90
#68			69 | 76.4%	10.12			55 | 73.0%	6.40	33 | 40.5%	7.94
#122	50 | 60.5%	14.02	87 | 96.6%	7.76	20 | 70.4%	13.69	50 | 66.2%	6.77	54 | 67.1%	6.91
#10			41 | 44.9%	11.44	16 | 55.6%	14.42	14 | 17.6%	9.98	12 | 13.9%	9.54

Table 4: Weekly rankings of each cluster of some vehicles of the "A_P1_M" group.

An essential aspect of this fuel efficiency ranking method is the time interval. This is very noticeable by comparing the rankings from Tables 4 and 5. Looking again at vehicle #68 it is possible to see that even if in one week of the month it had no trips clustered as "Arterial" and "Highway", in the whole month it had. Moreover, more interesting is that it performed exceptionally well, specifically in the "Highway" cluster monthly ranking, having the best efficiency of all drivers in that ranking for that month.

Context	**Arterial**		**Collector**		**Highway**		**Local-center**		**Local-outskirts**	
Vehicle	Position	Effi.	Position	Effi.	Position	Effi.	Position	Effi.	Position	Effi.
#129	66 | 65.0%	14.52	86 | 84.2%	10.87	24 | 42.6%	15.18	68 | 66.3%	7.72	14 | 12.9%	10.21
#68	43 | 42.0%	15.33	78 | 76.2%	11.06	1 | 0.0%	18.36	80 | 78.2%	6.67	56 | 54.5%	8.27
#122	65 | 64.0%	14.61	37 | 35.6%	12.33	38 | 68.5%	14.16	47 | 45.5%	9.28	53 | 51.5%	8.50
#10			50 | 48.5%	11.91	31 | 55.6%	14.70	43 | 41.6%	9.37	26 | 24.8%	9.82

Table 5: Monthly rankings of each cluster of some vehicles of the "A_P1_M" group.

The main objective of a ranking is to compare and, from this comparison, make the drivers more eager to perform better [11, 12, 13, 14]. The benefits of a fuel efficiency ranking come from that, as drivers strive to be better positioned in a ranking, less CO_2 is emitted as they collectively increase their fuel efficiencies. For this to be possible, a driver has to be able to know how it is performing. The graphs from Figure 12 show a possible layout for a driver to follow its fuel efficiency and ranking position in different clusters and throughout different weeks. With this type of visualization, the driver can monitor its performance over time and analyze where it can improve. An example of how helpful graphs like Figure 12 can be, comes from noticing how different is the performance of vehicle #167 in different clusters. While its performance is overall mediocre in the "Collector" trips, it is terrible in the "Highway" trips. This fact, made clear by the graphs, could motivate the driver to investigate the reasons for this and improve his driving.

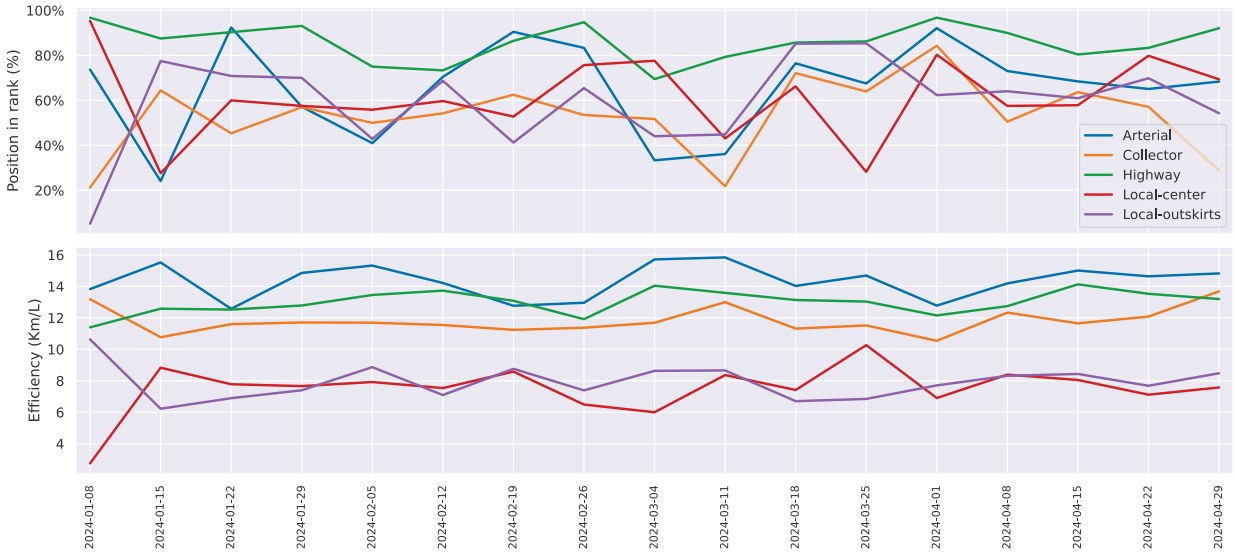

Figure 12: Position in weekly rankings and average efficiency of vehicle #167 throughout four months using the first ranking method.

Instead of including the driver in one ranking for each cluster it had trips clustered as, another possible way of ranking the drivers is by including them in just one cluster in each time period. However, a way to classify the drivers is needed for this to be possible, as the observations we initially clustered are the trips, not the drivers. The classification of a driver based on its trips' clusters can be made in different ways depending on the end goal. This driver's classification has at least two parameters: time interval and viewpoint. The time interval is the span of time in which a classification is made, for example, the driver's context in a day, week, or month. The viewpoint is related to the application of the classification, which is based on what the classification will be used for. For example, it could be based on the clustered trips' frequency, distance, or duration.

Figure 13 exemplifies this driver's classification method by using a weekly time aggregation and choosing the week's cluster by three different viewpoints: absolute frequency, weighted frequency based on distance, and weighted frequency based on time duration. In the frequency-based classification, just the number of occurrences of each cluster in a period is accounted for, with the classification of a period being the cluster of the most trips in that period. This classification method could be attractive for applications related to short trips. On the other hand, the distance and time-based ones could be better for longer and lengthier trips. In those cases, the weight of each trip is the distance or total time of that trip, making longer trips more important in the classification. This increased importance of longer trips makes sense when the analysis is focused on fuel consumption. By using this method, possible unbiased fuel efficiency rankings can be created by comparing the fuel efficiency of drivers from the same groups, clusters, and time periods.

From the classifications in Figure 13, it is possible to see why the viewpoint is essential, as the weekly classification changes drastically depending on how the trips' clusters are accounted for. This can be clearly seen in the first week of the analyzed period, the date "2024-01-02", where we have three different classifications for that week depending on the viewpoint. These different classifications are strongly related to the viewpoint used. The frequency viewpoint classified that week as "Local-outskirts", which makes sense since trips in this cluster are short and can be quickly done many times in a single day. The distance viewpoint classified that week as "Highway", which is expected given that trips in this cluster have very high distances. The time viewpoint classified that week as "Arterial", which also makes sense, as the high frequency combined with the high total time of the trips in this cluster resulted in a very high accumulated time.

A possibility of fuel efficiency ranking is partially displayed in Table 6. In Figure 14, we have a graph of this ranking comparing the fuel efficiency of all vehicles in each cluster. This ranking was built using data from the vehicles in the "A_P1_M" group. The vehicles were clustered with a weekly aggregation and with a distance-based viewpoint. The ranking shown in Table 6 and Figure 14 is just from one of the weeks in the analyzed period. The

efficiency numbers in the ranking are the weighted averages of the efficiencies on that week, where the weights are the distance of the trips. The results in this ranking are very similar to the ones from the box plots of Figure 11. Again, as in Figure 11, the most prominent difference in the ranking is between the vehicles from cluster "Local-outskirts" and the ones from cluster "Highway", where the worst vehicle from cluster "Highway" still has a better weighted average efficiency than the best from "Local-outskirts". Even though vehicle #25 is in first place in the ranking of its cluster, "Local-center", it still would be in last place if it was in the "Highway" cluster rank. This ranking example reinforces what we demonstrated in subsection 4.3, that a ranking without the context-based clustering segregation would be unfair.

Arterial			**Collector**			**Highway**			**Local-center**			**Local-outskirts**		
Rank	Vehicle	Effi.	Rank	Vehicle	Effi.	Rank	Vehicle	Effi.	Rank	Vehicle	Effi.	Rank	Vehicle	Effi.
1	#111	17.60	1	#77	13.04	1	#61	16.35	1	#25	11.71	1	#68	11.28
2	#31	16.93	2	#137	13.00	2	#165	15.71	2	#13	11.50	2	#5	8.22
3	#48	16.88	3	#85	12.85	3	#118	15.66	3	#151	8.97	3	#107	8.15
48	#28	11.15	18	#83	9.43	16	#167	12.57	4	#157	7.23	4	#67	8.09
49	#17	10.79	19	#141	9.33	17	#160	12.48	5	#30	5.87	5	#60	4.71
50	#143	9.28	20	#42	9.31	18	#74	12.08	6	#112	4.96	6	#125	1.14

Table 6: Rankings of one of the weeks of a weekly analysis from vehicles of the "A_P1_M" group.

Figure 15 follows the same idea as Figure 12. However, this time, we have just one position and fuel efficiency per period, as in this type of fuel efficiency ranking, a driver is included in just one cluster in each time period, different from the first one presented. In Figure 15, the colors of each point represent the ranking the driver was included in that week. Unlike the visualization of the first type of ranking presented, this one is much simpler, as it is impossible to know how a driver performed in different contexts. This simplicity, while being good due to its easy readability for the driver, can also be a disadvantage, as the information on the driver's performance in each context could be

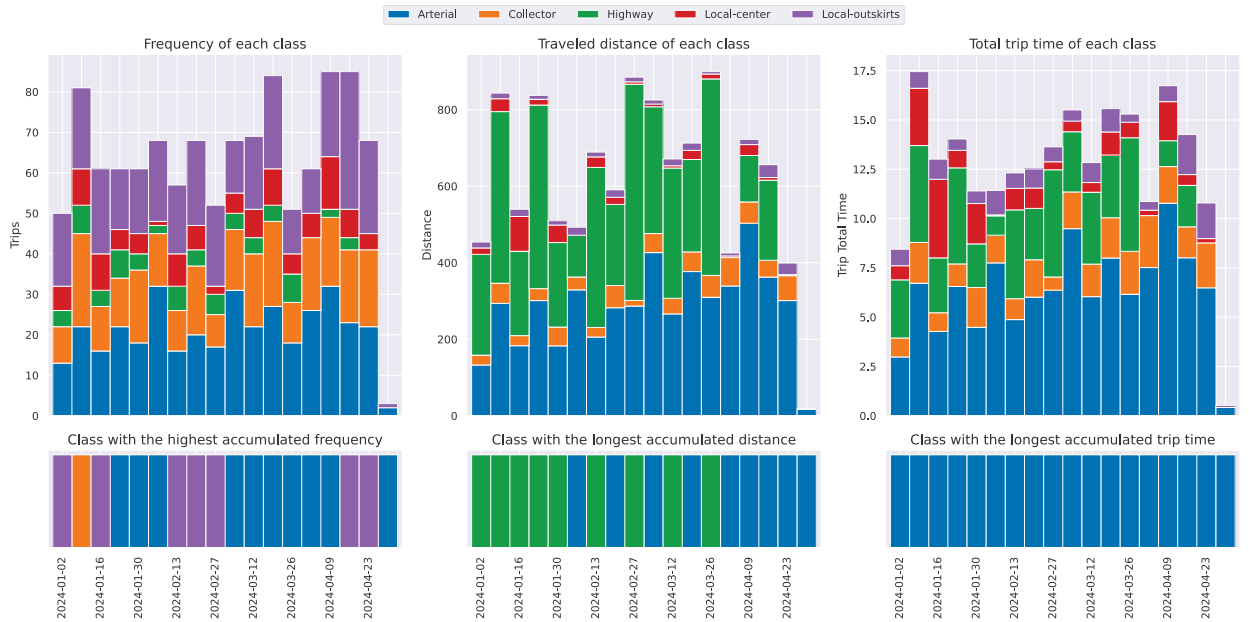

Figure 13: Classification of a driver from group "A_P2_M". Aggregated by week and using different viewpoints.

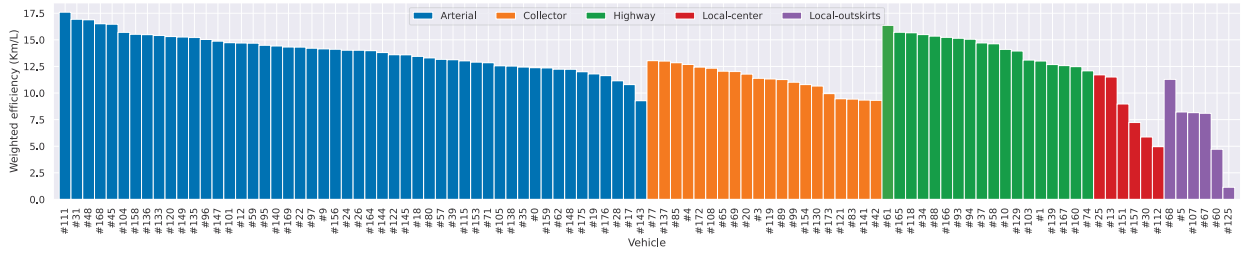

Figure 14: Visualization of the rankings of one of the weeks of a weekly analysis from vehicles of the "A_P1_M" group.

beneficial for a driver to know in what contexts it needs to improve its efficiency.

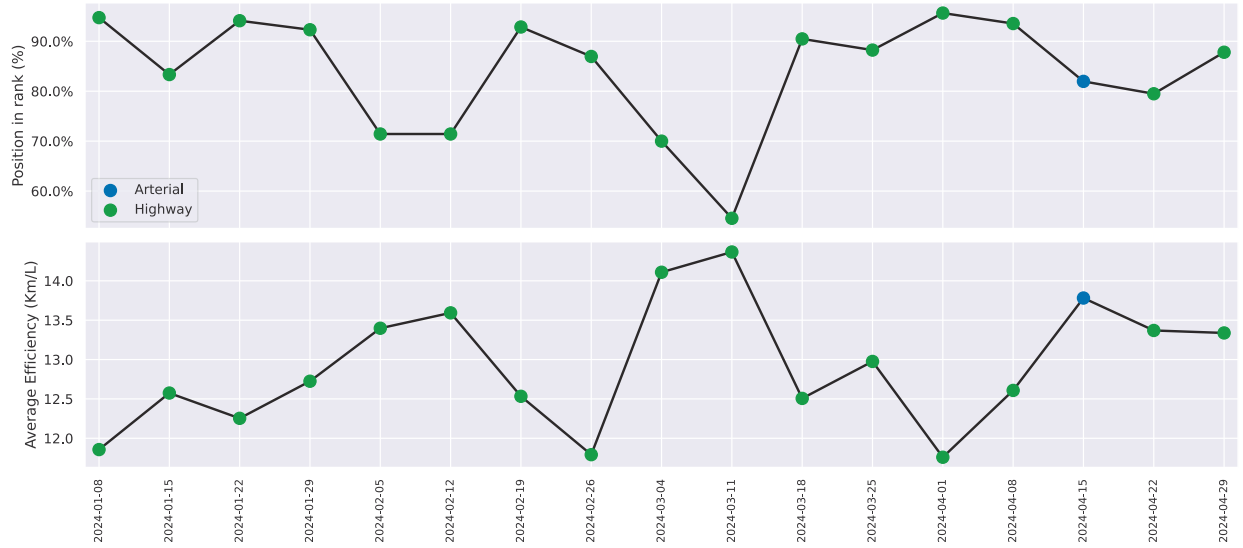

Figure 15: Position in the weekly ranks it was included and average efficiencies of vehicle #167 throughout four months in the second ranking method.

Additionally to the fair fuel efficiency rankings, another possible application of our clustering is analyzing drivers' behaviors. As one example of this application, we will dive into the fleet management area by analyzing the change over time in the clusters' ratio of a group of vehicles. Fleet management is a valuable area that can not be overlooked by various sectors: vehicle manufacturers, rental, insurance, and delivery companies. All of them can gain something by better understanding their fleets. Figure 16 is an example of how a fleet of vehicles can be monitored and analyzed. In this figure, vehicles from the "B_P4_E" group are used, and their drivers were classified using the weekly classification with the distance viewpoint. For each week, it shows the percentage of drivers classified in each cluster and how this percentage changes over time.

Figure 16 displays a fascinating overall behavior of the drivers, mainly with the changes in the ratios of the clusters "Highway" and "Collector". It is very intriguing how these two clusters seem to have an almost "sum zero" relation, i.e., when the rations of "Highway" increase, the ones of "Collector" decrease, and the inverse is also true. It is also very noticeable how cluster "Highway" ratios increase around the north hemisphere during summer time, and it is even more noteworthy how spikes in the line of ratio change appear at Christmas time. This behavior of the ratios from the cluster "Highway" reinforces the common knowledge that most long trips happen around summer and holiday times. Also, it gave us the surprising information that Christmas is by far the most long trip intensive holiday. By using our context-based clustering to analyze fleet behavior, we were not just able to confirm common knowledge that summer has more long trips but also detect a not-as-obvious behavior related to Christmas in Europe and the intriguing relation between the drivers classified in the "Highway" and "Collector" clusters.

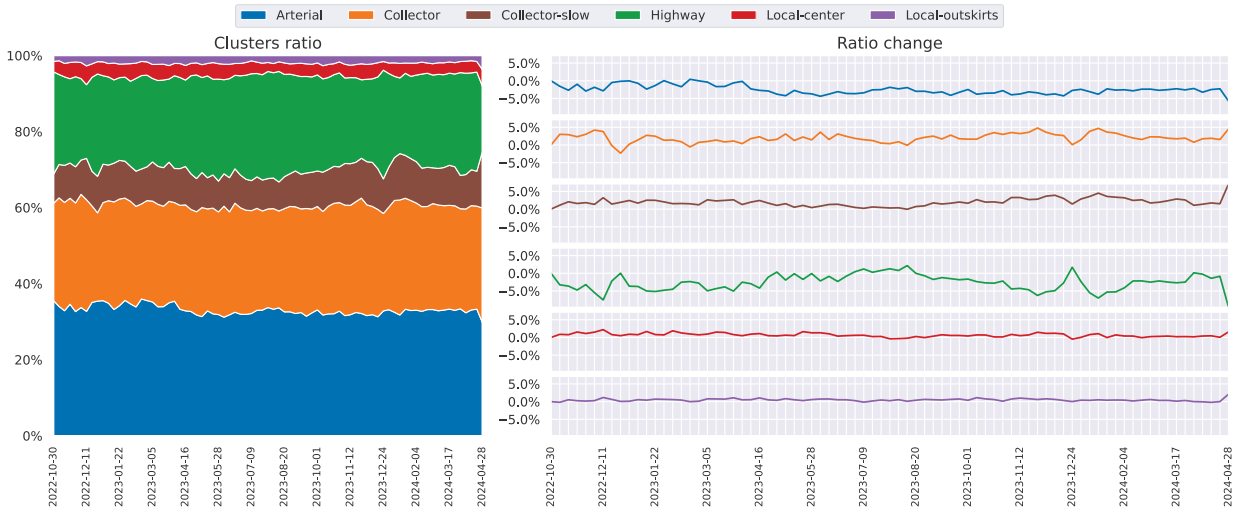

Figure 16: Ratio of the clusters and its change over time for the vehicles from the "B_P4_E" group.

6. Conclusion

The recent increase in connected vehicles brought a surge of vehicle-related data. This data, in turn, increased the possibilities of how connected vehicle information can be utilized to increase our understanding of the drivers and the vehicles. One of the many possible ways to use this data is to identify how and where the vehicles have been driven and, in doing so, apply this knowledge to desired applications. The context-based clustering presented in this paper works as a tool for this purpose, developed primarily to enable the construction of unbiased fuel efficiency rankings of the drivers without having to use highly sensitive positional data to identify the context. These rankings are constructed to passively decrease CO_2 emissions by using the drivers' competitiveness to increase their fuel efficiencies by having it compared to other drivers' efficiencies.

The proposed framework deals with data produced from vehicles with a vehicle-to-cloud (V2C) connection and stored in data lakes. A data treatment was proposed before clustering models could be trained and tested. Also, a method for training and validating multiple clustering models was proposed as a tool to find the optimal models. Several clustering algorithms were considered and tested, but ultimately, the k-means with improved initiation was chosen. The coupling of the k-means++ with our model's optimization method created fast models that generated well-separated clusters.

Given the definition of context in this research, the type of road in which the vehicle was driven, and the features used in our clusterings, the chosen nomenclature for our clusters was from the roads' hierarchy. As such, this hierarchy is divided based on the speed limit, traffic flow, and accessibility to property. The application of our method to data from actual vehicles of different models and types from Europe showed how well trips can be separated into different contexts. Overall, all vehicles showed similar clustering of their trips, and vehicles of the same models showed almost equal clustering. The clusterings were well-defined and fitted well with their corresponding road hierarchy. Furthermore, with correlation analysis, we demonstrated how each individual feature affected the models' decisions.

Our proposed context-based clustering solves the unfairness of comparing drivers' efficiencies without caring about their situations. We demonstrated that, according to common knowledge and our initial hypothesis, context, mainly defined by the vehicle's speed, trip distance, and duration, has a real influence on the vehicle's fuel efficiency. Moreover, our empirical tests with real vehicles' data showed that the different clusters created by the constructed clustering models have different fuel efficiency distributions. This fact further clarifies the fundamental importance of context for fuel efficiency comparison, as the positions in a ranking comparing efficiencies from trips in different contexts would mainly be influenced by the context and not the drivers' driving styles.

We demonstrated the main application of our method by building two types of fuel efficiency rankings separated by context. Our rankings further demonstrated the disparity in efficiency between the clusters, with the best of some

clusters being worse than the worst of other clusters. Also, using the ranking results from individual drivers, we showed how their rankings information, positions, and efficiencies in each cluster throughout time could be displayed in a way that instigates them to analyze and improve their efficiencies in different environments. Thus, this indicates an approach that can possibly passively decrease CO_2 emissions.

Although our method's primary purpose was to create unbiased fuel efficiency rankings, its applications are not limited to it. We demonstrated that by exposing how functional the context-based clustering can be for analyzing fleet behavior, an essential element for many businesses that depend on fleet management. In our analysis, we detected many non-trivial global behaviors of the fleet, such as the fluctuations of the clusters over time and the relation between the clusters.

In this research, we aimed to demonstrate the importance and usefulness of a context-based clustering of drivers for the automotive industry. Although we presented some of its applications, much remains to be explored. An application that could open up to a whole new research is the integration of the context-based clustering with recommendation systems [40]. The information on what context the driver fits the best in a specific time interval could help in choosing the recommendations the system should give the driver at that time. The area of product development also has a lot to gain by using the context-based clusters information, which can provide new viewpoints in data analysis. One example of this is in electric vehicles battery size analysis. The battery pack is one of the most expensive and pollutant parts of electric vehicle production. Thus, analyzing how its size can be reduced is of great importance. The context-based clustering can provide a more focused view of different interest groups, possibly showing the main differences between urban and highway drivers. This also allows the discovery of vehicle features used by drivers from different clusters, allowing vehicles to be produced with fewer unnecessary features.

Acknowledgment

The authors thank Renault from Brazil and Araucária Foundation for the partnership and financial support in the development of this research.

References

[1] C. Llopis-Albert, F. Rubio, F. Valero, Impact of digital transformation on the automotive industry, Technological Forecasting and Social Change 162 (2021). doi:doi:https://doi.org/10.1016/j.techfore.2020.120343.

[2] S. L. Bangare, S. Prakash, K. Gulati, B. Veeru, G. Dhiman, S. Jaiswal, The architecture, classification, and unsolved research issues of big data extraction as well as decomposing the internet of vehicles (iov), in: 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 566–571. doi:doi:10.1109/ISPCC53510.2021.9609451.

[3] R. Bohnsack, H. Kurtz, A. Hanelt, Re-examining path dependence in the digital age: The evolution of connected car business models, Research Policy 50 (2021). doi:doi:https://doi.org/10.1016/j.respol.2021.104328.

[4] F. Kuhnert, C. Stürmer, A. Koster, Five trends transforming the automotive industry, 2018. URL: https://www.pwc.com/gx/en/industries/automotive/assets/pwc-five-trends-transforming-the-automotive-industry.pdf.

[5] K. L. Lim, J. Whitehead, D. Jia, Z. Zheng, State of data platforms for connected vehicles and infrastructures, Communications in Transportation Research 1 (2021). doi:doi:https://doi.org/10.1016/j.commtr.2021.100013.

[6] M. Swan, Connected car: Quantified self becomes quantified car, Journal of Sensor and Actuator Networks 4 (2015) 2–29. doi:doi:10.3390/jsan4010002.

[7] A. Karmańska, The benefits of connected vehicles within organizations, Procedia Computer Science 192 (2021) 4721–4731. doi:doi:https://doi.org/10.1016/j.procs.2021.09.250.

[8] G. Abdelkader, K. Elgazzar, A. Khamis, Connected vehicles: Technology review, state of the art, challenges and opportunities, Sensors 21 (2021). doi:doi:10.3390/s21227712.

[9] A. Athanasopoulou, M. de Reuver, S. Nikou, H. Bouwman, What technology enabled services impact business models in the automotive industry? an exploratory study, Futures 109 (2019) 73–83. doi:doi:https://doi.org/10.1016/j.futures.2019.04.001.

[10] Y. Wang, Q. Miao, The impact of the corporate average fuel economy standards on technological changes in automobile fuel efficiency, Resource and Energy Economics 63 (2021) 101211. doi:doi:https://doi.org/10.1016/j.reseneeco.2020.101211.

[11] S. M. Garcia, A. Tor, T. M. Schiff, The psychology of competition: A social comparison perspective, Perspectives on Psychological Science 8 (2013) 634–650. doi:doi:10.1177/1745691613504114.

[12] J. Duffy, T. Kornienko, Does competition affect giving?, Journal of Economic Behavior & Organization 74 (2010) 82–103. doi:doi:https://doi.org/10.1016/j.jebo.2010.02.001.

[13] J. Brankovic, L. Ringel, T. Werron, How rankings produce competition: The case of global university rankings, Zeitschrift für Soziologie 47 (2018) 270–288. doi:doi:doi:10.1515/zfsoz-2018-0118.

[14] C. Anderson-Hanley, A. L. Snyder, J. P. Nimon, P. J. Arciero, Social facilitation in virtual reality-enhanced exercise: competitiveness moderates exercise effort of older adults, Clinical Interventions in Aging 6 (2011) 275–280. doi:doi:doi:10.2147/CIA.S25337.

[15] H. Zhang, J. Sun, Y. Tian, The impact of socio-demographic characteristics and driving behaviors on fuel efficiency, Transportation Research Part D: Transport and Environment 88 (2020). doi:doi:https://doi.org/10.1016/j.trd.2020.102565.

[16] J. Hou, Z. Song, A hierarchical energy management strategy for hybrid energy storage via vehicle-to-cloud connectivity, Applied Energy 257 (2020). doi:doi:https://doi.org/10.1016/j.apenergy.2019.113900.

[17] Y. Ma, H. Ding, Y. Liu, J. Gao, Battery thermal management of intelligent-connected electric vehicles at low temperature based on nmpc, Energy 244 (2022). doi:doi:https://doi.org/10.1016/j.energy.2021.122571.

[18] J. Shi, B. Xu, Y. Shen, J. Wu, Energy management strategy for battery/supercapacitor hybrid electric city bus based on driving pattern recognition, Energy 243 (2022). doi:doi:https://doi.org/10.1016/j.energy.2021.122752.

[19] A. Dowthwaite, D. Cook, A. L. Cox, Privacy preferences in automotive data collection, Transportation Research Interdisciplinary Perspectives 24 (2024). doi:doi:https://doi.org/10.1016/j.trip.2024.101022.

[20] R. J. Lempert, B. Preston, S. M. Charan, L. Fraade-Blanar, M. S. Blumenthal, The societal benefits of vehicle connectivity, Transportation Research Part D: Transport and Environment 93 (2021). doi:doi:https://doi.org/10.1016/j.trd.2021.102750.

[21] C. Zhang, W. Huang, T. Niu, Z. Liu, G. Li, D. Cao, Review of clustering technology and its application in coordinating vehicle subsystems, Automotive Innovation 6 (2023) 89–115. doi:doi:https://doi.org/10.1007/s42154-022-00205-0.

[22] L. D. Seixas, Vehicle industry big data analysis using clustering approaches, Master's thesis, Federal University of Technology – Paraná, Ponta Grossa, Brazil, 2022. Available at https://repositorio.utfpr.edu.br/jspui/bitstream/1/31331/1/vehicleindustrybigdata.pdf.

[23] A. Kabra, Clustering of Driver Data based on Driving Patterns, Master's thesis, Blekinge Institute of Technology, Karlskrona, Sweden, 2019. Available at https://www.diva-portal.org/smash/get/diva2:1337166/FULLTEXT02.

[24] L. Chabariberi, F. Sobral, S. Peres, Driving behavior analysis: An approach using clustering algorithms, in: WORKSHOP ON UNDERGRADUATE RESEARCH ON INFORMATION SYSTEMS - BRAZILIAN SYMPOSIUM ON INFORMATION SYSTEMS (SBSI), volume 15, Sociedade Brasileira de Computação, 2019, pp. 25–28. doi:doi:10.5753/sbsi.2019.7433.

[25] D. I. Tselentis, E. Papadimitriou, Machine learning approaches exploring the optimal number of driver profiles based on naturalistic driving data, Transportation Research Interdisciplinary Perspectives 21 (2023). doi:doi:https://doi.org/10.1016/j.trip.2023.100900.

[26] A. Mohammadnazar, R. Arvin, A. J. Khattak, Classifying travelers' driving style using basic safety messages generated by connected vehicles: Application of unsupervised machine learning, Transportation Research Part C: Emerging Technologies 122 (2021). doi:doi:https://doi.org/10.1016/j.trc.2020.102917.

[27] P. Ping, W. Qin, Y. Xu, C. Miyajima, K. Takeda, Impact of driver behavior on fuel consumption: Classification, evaluation and prediction using machine learning, IEEE Access 7 (2019) 78515–78532. doi:doi:10.1109/ACCESS.2019.2920489.

[28] A. Mohammadnazar, Z. H. Khattak, A. J. Khattak, Assessing driving behavior influence on fuel efficiency using machine-learning and drive-cycle simulations, Transportation Research Part D: Transport and Environment 126 (2024). doi:doi:https://doi.org/10.1016/j.trd.2023.104025.

[29] E. Hajhashemi, P. Sauri Lavieri, N. Nassir, Identifying electric vehicle charging styles among consumers: a latent class cluster analysis, Transportation Research Interdisciplinary Perspectives 27 (2024). doi:doi:https://doi.org/10.1016/j.trip.2024.101198.

[30] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on

Mathematical Statistics and Probability, University of California Press, USA, 1967, p. 281–297. URL: https://api.semanticscholar.org/CorpusID:6278891.
[31] F. Nielsen, Hierarchical Clustering, Springer International Publishing, 2016, pp. 195–211. doi:doi:10.1007/978-3-319-21903-5_8.
[32] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996, p. 226–231.
[33] U. von Luxburg, A tutorial on spectral clustering, Stat Comput 17 (2007) 395–416. doi:doi:https://doi.org/10.1007/s11222-007-9033-z.
[34] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, USA, 2007, p. 1027–1035.
[35] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65. doi:doi:https://doi.org/10.1016/0377-0427(87)90125-7.
[36] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1 (1979) 224–227. doi:doi:10.1109/TPAMI.1979.4766909.
[37] V. Eppell, B. McClurg, J. M. Bunker, A four level road hierarchy for network planning and management, in: Proceedings of the 20th ARRB Conference, ARRB Transport Research Ltd., ARRB, Australia, 2001, pp. 1–7. URL: https://eprints.qut.edu.au/2349/1/2349.pdf.
[38] S. M. Stigler, Francis Galton's Account of the Invention of Correlation, Statistical Science 4 (1989) 73–79. doi:doi:10.1214/ss/1177012580.
[39] M. A. Hall, Correlation-based feature selection for machine learning, Department of Computer Science, University of Waikato (1999). URL: https://www.lri.fr/~pierres/donn%E9es/save/these/articles/lpr-queue/hall99correlationbased.pdf.
[40] M. M. Ibrahim, F. S. Mubarek, Survey on vehicles recommendation systems, AIP Conference Proceedings 3009 (2024). doi:doi:10.1063/5.0190397.

5 OPTIMIZING EV BATTERY SIZING WITH ICEV ENERGY CONSUMPTION AND CONTEXT-BASED CLUSTERING

In the same way as in the previous paper, in this one our objective is also to demonstrate how connected vehicles' data can generate value. However, in this one, the focus is on the size of batteries for electric vehicles (EVs).

In recent years, the optimization of the battery size for EVs has grown in importance as EVs are being produced and sold more than ever. A possible reduction of the battery size can translate directly to lower production costs and lower environmental impact. In this paper, our proposal to optimize battery size is also dependent on the need to sell EVs to drivers of internal combustion engine vehicles (ICEVs). For this to be possible, the battery size must not constrain the previously ICEVs drivers. Given this fact, data from trips of ICEVs are used to stipulate the optimal battery so that EVs can equal ICEVs, range-wise, with minimal changes in how they are driven.

As data from real ICEVs are used, there is no energy consumption data. To generate this value, a function that uses average speed and temperature as input and outputs energy efficiency values is used. With the energy consumption calculated, energy thresholds are set to simulate battery sizes. Another aspect considered is recharging, something that, ideally, should not take too much time given our proposal that the drivers are previous ICEVs drivers. For this, two situations are compared, one without recharging in the day and one with limited time in-between trips recharging.

In the previous paper, a context-based clustering was presented. Here, we use this method to analyze how the different contexts affect energy consumption and, consequently, how it affects the battery needed to complete the trips in different contexts.

This paper was submitted to the journal "Energy" and, until the writing of this, is under review.

Optimizing EV Battery Sizing with ICEV Energy Consumption and Context-Based Clustering

João Gabriel Santin Botelho[a,*], Eduardo Alves Portela Santos[a], José Eduardo Pécora Junior[a], Alexandre Magrini[b], Rodrigo Balani[c], Janaine Rodrigues[c]

[a]*Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE), Universidade Federal do Paraná, Evaristo F. Ferreira da Costa 408, Curitiba, Paraná, 81530-015, Brazil*
[b]*Renault, Av. du Golf 1, Guyancourt, Île-de-France, 78280, France*
[c]*Renault do Brasil, Avenida Renault 1300, São José dos Pinhais, Paraná, 83070-900, Brazil*

Abstract

One of the most important sections in an electric vehicle (EV) production process is the choice of its battery. In the current state of battery technology, batteries for EVs are expensive, heavy, and volumetric. Therefore, an extensive study of how it can be optimally sized for specific vehicles can be very impactful. In this paper, by using data from actual internal combustion engine vehicles (ICEVs) we analyse how different driving profiles impact battery energy consumption and recharging. The energy consumptions of the trips are estimated using a function dependent on the trips' average speed and external temperature. The driving profiles are identified by a context-based clustering method, which mainly identifies what type of road the driver was mostly driving on in each of its trips. The recharging type used is the fast charge, as it is ever so more common in modern charging stations nowadays. Our analysis hopes to answer the question of which battery size is ideal for each driving profile and how a good fast-charging infrastructure can further help in reducing EVs' battery costs.

Keywords: Connected vehicles, Driving context, Battery Size, Data Analysis, Energy

1. Introduction

More than ever, the industry is shifting towards data-driven solutions and adapting to new technologies [1]. Nowadays, collecting and using vast amounts of data is already a reality for most medium to big-size companies. This growing accumulation of vastly different types of data is driven by the ever-growing possibilities in the generation of value that the advancements in data processing and machine learning techniques are creating [2].

In the automotive industry, data collection is becoming a common trend [3] as more connected vehicles enter the market each year [4]. A connected vehicle, precisely a vehicle with V2C (vehicle to cloud) connection [5], is a cellular-enabled vehicle that connects to a central server, enabling the receiving and sending of data using the mobile network [6]. The increase in the development and production of connected vehicles is a direct consequence of the many benefits it can bring to the drivers [7] and the many ways it can generate value for the vehicle's manufacturer [8]. Also, the increase in the production of electric vehicles (EVs) in recent years has had a direct impact on vehicle connectivity, as EVs with their mostly digital systems are perfect for the implementation of connectivity features.

From a manufacturer's internal point of view, the greatest gain it can have with connected vehicles is the value the collected data can generate. Aiming to decrease production costs and increase sales, a big part of vehicle development is identifying features that can be excluded and should be included in new models. Gathering and analyzing connected vehicles' data can easily replace most surveys performed to identify those features [9]. Furthermore, this data is often more reliable than the answers from survey participants as they contain the ground truth of how the vehicles are being

*Corresponding author
Email address: joaobotelho@ufpr.br (João Gabriel Santin Botelho)

used. Moreover, with the increased amount and quality of data comes the question of what is important enough to be the focus of analyses.

The manufacturing of EVs has seen an explosion in production and sales since 2020 [10]. The huge growth in market and production can be related to many factors, mainly environmental and technological. The recent European Union (EU) regulations on CO2 emission standards forced automotive companies with a market in Europe to increase investments in the production of EVs, and these regulations favoring low-emission vehicles are not restricted to Europe, with similar bills passing all around the globe [11]. Consequently, this huge growth in EVs' investment also created a big opportunity for technological advancements in this area, not only related to the direct manufacturing process but also on indirect parts of the process as the batteries and the supply chains. This in turn made EVs' cost decrease and production increase even further, culminating in the market we have today.

With this growing market of EVs a question that still remains for all manufacturers is how big an EV model battery should be. This question is of utmost importance due to many factors, but it mainly comes to cost. Even with the ever-declining cost per kWh of batteries in the past years [12, 13] due to technological progress, the battery still accounts for around 30% of an EV cost [14]. The main contributor to this high cost is the cathode of the battery cell, which can account for 35% of the battery cell price and can be composed of many different materials. In the case of lithium, nickel, and cobalt oxides-based batteries, these metals' high costs are the main contributing factors to the battery's high costs. Even though batteries composed of nickel and cobalt oxides such as lithium nickel manganese cobalt oxide (NMC), lithium nickel cobalt aluminium oxide (NCA), lithium nickel cobalt manganese aluminium oxide (NCMA), and Lithium cobalt oxide (LCO) are more expensive than iron phosphate-based batteries such as the lithium iron phosphate (LFP), they are very desirable for EVs due to their much higher energy density.

Even though it mainly comes to a matter of costs, smaller batteries equal cheaper EVs, another area that manufacturers must pay attention to is the clients' desires. Which battery size would be ideal for each type of driver? Which would be ideal for each type of vehicle given the vehicle's target market? These are questions that can be answered by analysing the drivers' behaviour. In this paper, our objective is to find an answer to a similar question: Which battery size would be ideal for drivers who are migrating from internal combustion engine vehicles (ICEVs) to EVs?

The answer is simple: the ideal size is the one that does not make the driver change his driving habits. However, as stated before, cost plays a big role in battery size for both the manufacturers and the drivers, and different drivers behave differently making the ideal battery size different for each individual. Therefore, to find a realistic answer, our analysis will be based on real drivers' data and how their behaviour with ICEVs translates to EVs. For this analysis to be possible we used real ICEVs data collected from many different drivers from all around Europe. To convert their ICEVs data to simulated EVs data we used a function that, given an average speed and an average temperature, returns an energy efficiency. This function is specific for the ICEV that generated our data, as it considers its weight and drag coefficient, using the data of electric engines from this vehicle's manufacturer. This simulated energy consumption allows us to analyse how much energy is being consumed by a trip and, supposing a vehicle can be fully recharged every day, how much energy is consumed in a day. These energy values are our reference to which battery size would be ideal, analysing which energy threshold would allow which full-day trips to be completed.

Infrastructure is a key requirement for the sales sustainability of the ever-growing production of EVs. As gas stations are imperative for ICEVs to be functional means of transportation, charging stations are also indispensable for EVs. Given this fact, the increase in the charging station infrastructure is following the growth of EV numbers in most developed countries [10]. Therefore, the existence of means to easily recharge an EV outside the house and to do so in a fast time, something made possible by DC fast chargers, can change our initial idea of just fully recharging the vehicle from one day to the next. Moreover, the recharging possibility can also decrease the battery size needed to travel the same distance with the drawback of having to do recharging stops. To include this outside-home recharging possibility, we also simulated trips with recharging in-between trips. The recharging curves used to simulate how much charge can be gained given a certain amount of battery state of charge (SoC) and a time of recharging were from 150 kW DC fast chargers. These are types of chargers that are publicly available and are growing in quantity due to the increase in the demand of high-speed recharging. The curves used simulate the recharging process simulate the recharging of EVs models with NMC-type batteries from the same manufacturer as the analysed ICEVs.

As stated before, not every driver behaves equally. There are drivers that almost never go out of city centers and drivers that take the highway every day. These differences in behaviour undoubtedly affect the battery size these drivers would need. To consider this, we also applied a context-based clustering to the drivers to classify each different day and analyse how the different contexts affect battery size. This clustering is done by k-means models and uses

context-related values such as average speed, trip distance and time, and time and different speed intervals. The contexts are identified as the road types in which the vehicle stayed for most of the trip.

The remainder of this paper is organized as follows: Section 2 discusses previous studies related to the topics presented in this paper. Section 3 presents the data used in our analysis. Section 4 presents the methods employed to assess the energy consumption of the ICEVs, battery size calculation, the context-based classification, and the recharging of the batteries. Section 5 presents the results obtained and how they affect our battery size decision. Lastly, Section 6 presents the study's conclusions and possible follow-up research related to the studied topic and reached results.

2. Related Works

Before producing EVs, it is important to understand how the drivers see EVs and what are the main obstacles to greater EV adoption. In [17] a survey of potential EV consumers was made, with the objective of investigating the barriers to EV adoption. They found that the main barriers were financial, technological, and infrastructure. In [16] a similar conclusion is reached, with the interesting additional finding that reputation and status also influence EV purchase, with reputation-driven customers opting less for EVs in the situation that EVs cost the same as ICEVs. Another survey is made in [15] but with a focus on EV drivers' satisfaction. They found that the intention for cost-saving during operation is a key factor for EV users' satisfaction and that satisfied users have the intention to repurchase and recommend EVs to others. An exhaustive literature review is done on what are the main factors that affect the consumers' intention to adopt EVs in [18]. It was found that situational type factors were the most influential. They also found that the lack of charging station availability and limited driving range were the most cited barriers to the adoption of EVs, while the reduction in air pollution and the availability of policy incentives were the most cited motivators.

To obtain the energy consumption from ICEVs' trips, we used a function based on average speed and external temperature, as there is plenty of literature on this topic. In [19, 20, 21, 22, 23, 24, 25] the effects of external temperature in energy consumption are well explored with different approaches. However, even while analysing different vehicles, all these studies showed similar results. The function used in this paper also shows similar results to these studies, with low and high temperatures affecting negatively the energy efficiency and with the worst results occurring in really low temperature ranges. Another important aspect of energy consumption that is accounted for in the function used in our study is the vehicle's average speed, which is also well explored in [19, 24, 25]. All these papers demonstrate how those two factors together affect energy consumption, also with the addition of other factors. For example, in [19] analysis, a division of the trips into 3 types, urban, rural, and motorway, is done. And in [24, 25] the driving style factor, with different levels of aggressiveness, is analysed, and [25] infrastructure factors like curves and slopes are also accounted for. In [32] the authors analyse real EVs' data, investigating factors that influence consumption. However, in this paper, the authors go one step further by using this data to train machine learning models to predict consumption curves.

The topic of required ranges and battery size optimization for EVs does not lack in papers, which is expected given its importance in recent years when EV production is growing exponentially. Before the explosion in EV production, [27] is a study that investigated, based on ICEVs' daily trips, the ranges that EVs should be able to do to be viable substitutes to ICEVs. A more recent study that has a similar objective to ours is [26]. Although lacking in sample size, as just one vehicle was investigated, it successfully showed how battery size can be reduced considerably without affecting the driver's usage. In [29], a very complete TCO (total cost of ownership) analytical method is proposed, which encompasses many variables that affect the cost of EVs as the different geographical regions, ranges, and policies. In their analyses, it is concluded that the TCO of all EVs, independent from battery size, is significantly lower than ICEVs and plug-in hybrid electric vehicles (PHEVs) in almost all regions. [30] analyses consumption data from real-world driving and uses models to generate driving cycles to investigate the usage of different battery sizes. Their findings show that different drivers require different battery sizes, depending on climate and range, with the maximum size needed reaching just 70 kWh. In [28], the authors' focus is another side of batteries that can be as important as their size: when and by how much they should be recharged. They show that usage outside the 20%-80% of SoC range can lead to more energy consumption and faster degradation.

When investigating the possibilities of different sizes of batteries, an important factor to consider is how they are recharged and how the recharging can affect the grid and the battery size. [31] studies how EV recharging can impact

the grid, showing the peak recharging hours of the analysed vehicles, and also how energy is consumed. In [33] the two main range extenders for EVs are considered: bigger battery or better fast-charging infrastructure. The study concludes that fast charging can significantly increase range, and analyses many aspects of the current infrastructure in Germany and how it could be further improved. [35] proposes an optimal hybrid-type battery pack with a low range but extremely fast recharging. Such a battery would only be possible if fast chargers were readably available. In [34] different recharging strategies are considered to optimize charging stations usage. Two strategies are compared: communication and reservation. They showed that the communication strategy, which has real-time information sharing, can reduce trip time and optimize charging station usage in comparison with the reservation strategy, which can waste the potential usage of a charging spot.

The high charging rates of fast chargers can significantly impact the grid and vary the energy demand. With the need to increase the number of stations with fast chargers, solutions to this problem are being researched. The use of battery storage units as buffers between the grid and the stations is a viable strategy. In [36], the author optimizes the size of those battery storage units and also demonstrates the feasibility of it in cases with energy arbitrage. [37] also makes this optimization and shows that it can reduce operational costs due to energy arbitrage. In [38] the battery storage units are optimized also considering the forecasting of photovoltaic power production.

3. Data

The data used in this study is from connected vehicles of a French automotive company from all over Europe, where each observation is a trip, initiated when the vehicle is turned on and finalized when it is turned off. Table 1 is an example of the type of data we have, where each row is a different trip, and each column holds different information about the trips.

VIN hashed	Start Time	Finish Time	Distance (km)	Avg. Speed (km/h)	Avg. Temp. (°C)	···	Model	Power
76973	01/01/2023 12:00:00	01/01/2023 12:30:00	10	60	15	···	A	P1
92928	01/01/2023 12:20:00	01/01/2023 13:50:00	111	150	17	···	A	P1
36348	01/01/2023 12:35:00	01/01/2023 12:40:00	2	30	16	···	A	P1
76759	01/01/2023 12:40:00	01/01/2023 13:05:00	12	120	19	···	A	P1
23915	01/01/2023 12:42:00	01/01/2023 13:22:00	24	230	17	···	A	P1
46851	01/01/2023 12:50:00	01/01/2023 13:15:00	7	130	20	···	A	P1

Table 1: Example of the data structure.

Although data from many vehicles and different models are available, here we analyse a single hybrid model. From this single model, we had almost 10 million trips from more than 8 thousand different vehicles, spanning from the beginning of 2023 to the end of 2024. To increase the quality of our data, we made filters to discard data from drivers without periodic usage of their vehicles. Also, we focused our analysis on a specific one-year period, from July 2023 to July 2024. This was done so as to not unbalance our data by including periods with specific events like summer holidays more than one time while just including other specific events like winter holidays just one time, as in the case if the analysed period was from May 2023 to October 2024. With this data pruning, our final dataset had around 3.2 million trips from around 2.5 thousand different vehicles.

4. Methodology

4.1. Energy Consumption Calculation

The reason for our single model analysis comes from our method of stipulating the energy consumption of non-electric vehicles as if they were electric vehicles (EVs). The main causes of deviation in overall energy consumption of different models of EVs, disregarding deviations from individual usage, are the different power trains, weights, and drag coefficients. A simulated energy consumption table that assumes a specific electric power train in a hybrid model

body was provided to us. Therefore, we can only stipulate the energy consumption of this specific hybrid vehicle model, as the simulated data assumes its weight and drag coefficient.

The energy consumption table provides to us is a discrete function $D : \mathbb{Z}^2 \to \mathbb{Q}$, where the two integer variables are average speed (km/h) and average external temperature (°C) values, and the rational image is the energy efficiency in kWh/100km. Table 2 is a representation of this function.

		Average Speed (km/h)													
		10	20	30	40	50	60	70	80	90	100	110	120	130	140
Temperature (°C)	-5	34.36	27.34	23.40	20.94	19.66	19.25	19.43	20.02	20.84	22.05	23.55	25.57	27.75	30.12
	5	26.73	21.57	18.97	17.51	16.89	16.90	17.40	18.12	19.01	20.26	21.76	23.63	25.65	27.84
	14	19.86	16.38	14.98	14.41	14.39	14.79	15.56	16.41	17.36	18.65	20.16	21.88	23.76	25.78
	23	17.36	14.63	13.60	13.19	13.28	13.78	14.64	15.56	16.54	17.84	19.33	20.99	22.79	24.73
	30	20.32	16.43	14.77	13.87	13.60	13.96	14.76	15.64	16.53	17.58	18.78	20.38	22.13	24.01
	40	24.54	19.00	16.45	14.84	14.06	14.23	14.92	15.76	16.51	17.58	18.78	20.38	22.13	24.01

Table 2: Energy efficiency discrete function.

Our simulated table data exists only for discrete values. However, the trips' speed and temperature values from the data we need to stipulate the energy consumption are all rational. Therefore, a way to calculate energy efficiency given continuous speed and temperature values is needed. One alternative is function fitting, i.e., find a function $F : \mathbb{R}^2 \to \mathbb{R}$ that best fits our discrete set of points. This can be achieved with polynomial regression, and in our case, we used a cubic function. Figure 1 is a heat map with the results of our function fitting. In the fitting, we achieved a mean absolute error of 0.634 kWh/100km and a mean absolute percentage error of 3.423%.

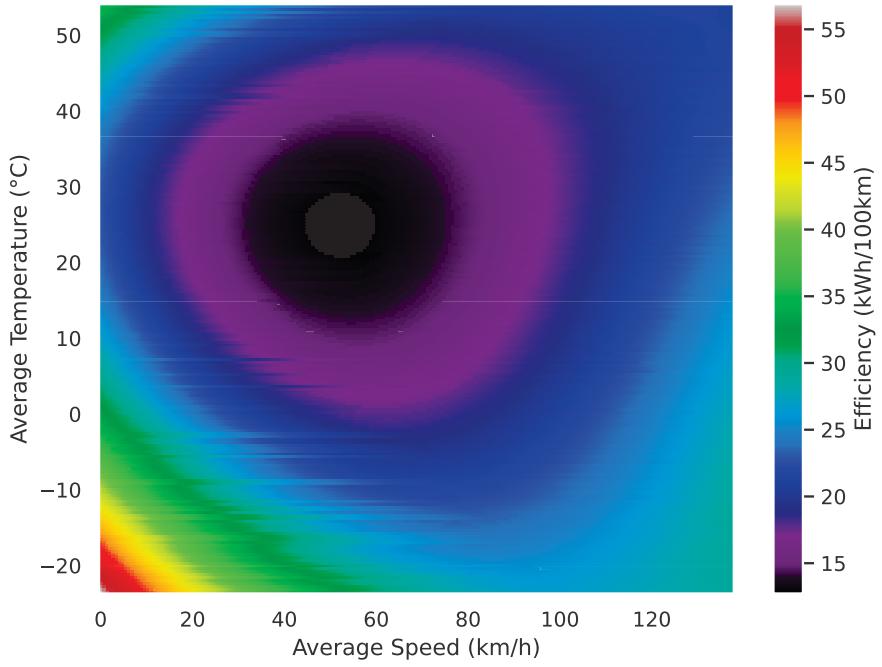

Figure 1: Continuous energy efficiency function heat map.

With the ability to calculate the energy efficiency of a trip, we also can calculate the energy consumption of a trip or a day. Table 3 is an example of these values calculated. Given a set of trips from a vehicle and their distances,

average speeds, and average temperatures, we can calculate their energy efficiency and energy consumption.

VIN	Date	Start time	End time	Distance (km)	Avg. Speed (km/h)	Avg. Temp. (°C)	Energy Efficiency (kWh/km)	Energy Consumed (kWh)	Day Consumption (kWh)
#1	25/03/2024	08:12	08:32	10	30	15	0.1558	1.5583	4.7658
#1	25/03/2024	12:32	13:02	25	50	22	0.1283	3.2075	4.7658
#1	26/03/2024	07:00	10:00	240	80	16	0.1536	36.8623	74.5125
#1	26/03/2024	10:15	12:15	180	90	18	0.1659	29.8548	74.5125
#1	26/03/2024	14:15	15:15	60	60	22	0.1299	7.7954	74.5125
#1	27/03/2024	08:15	08:35	10	30	12	0.1639	1.6392	1.6392

Table 3: Simulated energy consumption calculation.

4.2. Battery Size Calculation

Our battery size calculation is based on a daily energy usage threshold. Imposing many different daily energy thresholds, we can analyse the days and vehicles which did or did not respect each threshold, and if the threshold was not respected, it means that a battery of that size would not be sufficient. Table 4 has an example of these calculations.

VIN	Date	Energy Consumed (kWh)	Day Consumption (kWh)	Energy > 10kWh	Energy > 15kWh	Energy > 20kWh	⋯	Energy > 120kWh
#1	25/03/2024	1.5583	4.7658	No	No	No	…	No
#1	25/03/2024	3.2075	4.7658	No	No	No	…	No
#1	26/03/2024	36.8623	74.5125	Yes	Yes	Yes	…	No
#1	26/03/2024	29.8548	74.5125	Yes	Yes	Yes	…	No
#1	26/03/2024	7.7954	74.5125	Yes	Yes	Yes	…	No
#1	27/03/2024	1.6392	1.6392	No	No	No	…	No

Table 4: Battery size thresholds calculations.

Using the thresholds we can measure the number and percentage of days in which each battery size threshold was respected for each vehicle. Table 5 has an example of these calculations. With these values we can calculate the amount of coverage each battery size would have on a sample of vehicles, making it possible to choose the best size given a pursued objective such as a percentage of vehicles that need to be covered on a specific percentage of time.

VIN	N° of days	N° of days < 10kWh	N° of days < 15kWh	⋯	N° of days < 120kWh	% of days < 10kWh	% of days < 15kWh	⋯	% of days < 120kWh
VIN#2	240	156	179	…	238	65,00%	74,58%	…	97,08%
VIN#3	312	262	268	…	311	83,97%	85,90%	…	99,68%
VIN#4	230	138	145	…	220	60,00%	63,04%	…	95,65%
VIN#5	320	207	215	…	296	64,69%	67,19%	…	92,50%
VIN#6	290	213	218	…	281	73,45%	75,17%	…	96,90%

Table 5: Coverage of each battery size for each vehicle.

4.3. Battery Recharging Calculation

Besides the consumption of electric energy, another important aspect of EVs is the recharge of this energy. To simulate a more precise electric energy usage, we also included the recharging process in our analysis. In this study, we assume that at each interval between trips bigger than 2 minutes, the vehicle can be recharged. For example, if the interval between trips is 5 minutes, we assume that the vehicle was recharging for at most 3 minutes. Given this hypothesis, the recharging speeds are set by direct current (DC) fast charger curves, which in our case can reach an output of up to 150 kW. The decision to use this kind of recharging technology comes from the fact that their numbers are growing in public recharging stations due to the increase in the demand for high-speed recharging [10]. Figure 2 shows the state of charge (SoC) by time curves of different battery sizes on a DC fast charger for EVs models with NMC-type batteries from the same manufacturer from the analysed ICEVs.

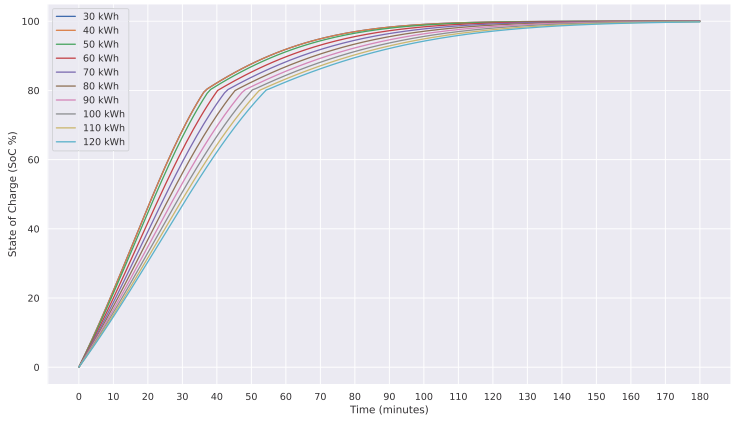

Figure 2: Fast charger simulated SoC by Time curves.

With the energy consumption of a trip and the time interval between trips, we can simulate SoC curves for each day, vehicle, battery size, and recharging time. Table 6 and Figure 3 represent how we did these calculations for an exemplifying case where the battery size was 60 kWh, and the maximum recharging time was 10 minutes. In this example, it is possible to see the cases when there is no time to recharge, less than 2-minute intervals, and the cases when there is more time, but it was limited to 10 minutes. Also, this example really shows the difference in how much can be recharged in 10 minutes when the SoC is as high as 90%, with only around 3% being recharged, and when it is as low as 10%, with more than 20% being recharged.

VIN	Date	Start Time	End Time	Distance (km)	Avg. Temp. (°C)	Avg. Speed (km/h)	Energy Effi. (kWh/km)	Energy Consu. (KWh)	Start SoC (%)	End SoC (%)	Recharged SoC (%)
#1	2023-12-24	06:59:54	07:33:39	23.021	-0.150	40.930	0.194	4.472	100.000	92.547	
#1	2023-12-24	13:04:57	13:31:59	22.580	6.455	50.127	0.159	3.589	95.168	89.186	2.620
#1	2023-12-24	14:26:20	16:37:36	196.639	8.440	89.885	0.180	35.379	92.704	33.740	3.518
#1	2023-12-24	16:39:17	17:43:39	87.201	5.590	81.297	0.174	15.201	33.740	8.405	0.000
#1	2023-12-24	19:15:01	19:29:08	8.320	2.050	35.360	0.193	1.606	29.462	26.786	21.057
#1	2023-12-24	23:38:47	23:53:17	8.080	1.168	33.417	0.201	1.624	48.723	46.017	21.938

Table 6: Simulated discharge and recharge calculations.

4.4. Context-Based Clustering

An aspect that cannot be ignored when dimensioning batteries for EVs is the type of driver for which those vehicles are being made. From a commercial viewpoint, it makes no sense to produce a small EV with low power and carry

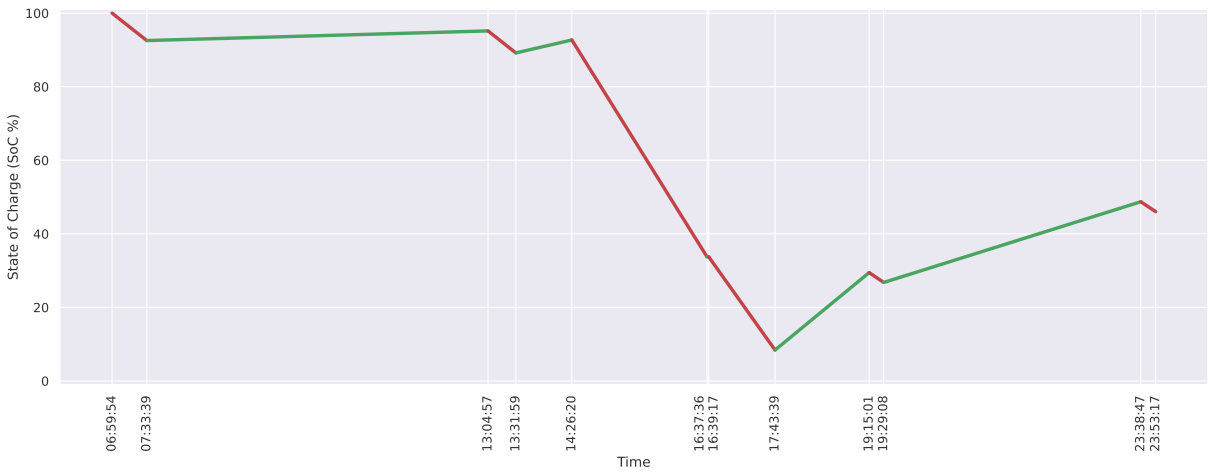

Figure 3: Representation of the SoC by time of Table 6.

capacity, but with a large battery, as it is known that most clients of this kind of small vehicles never go out of the city and only use it to commute from home to work. The same is true for the other side of the spectrum: producing a big SUV-type EV with medium to high power and carry capacity, but with a small battery. We will be using a context-based clustering to identify the different types of roads the vehicles are being driven, and how these different contexts affect electric energy usage both for consumption and recharging.

The context-based clustering method we are using is a k-means++ model based on features mostly related to speed, but also distance and duration. The contexts found and separated by this clustering model can be identified as road types due to their strong speed segregation. In Figure 4 we have the boxplot distributions of the three main features and in Figure 5 we have the correlations between the clustered groups, road types, and all the features. These figures in combination with Table 7 that has indicators of the clustered groups, demonstrate how and which of the trips are separated.

	Context				
	Arterial	**Collector**	**Highway**	**Local-center**	**Local-outskirts**
Avg. Speed (km/h)	46.86	27.45	78.61	14.97	15.50
Avg. Distance (km)	16.48	4.89	75.63	3.94	1.59
Avg. Duration (min)	20.59	10.20	54.05	13.59	5.53
Avg. Temp. (°C)	14.69	15.28	15.42	16.07	16.47
Freq. Count (%)	28.48	33.16	6.61	12.61	19.14
Freq. Distance (%)	57.90	22.90	14.05	2.67	2.48
Freq. Time (%)	48.46	32.55	6.85	6.27	5.87

Table 7: Indicators of the different road types.

However, given the way we are calculating the energy consumption, analysing a whole day, we also need to classify the driver's days, not only the trips. This driver's classification has one parameter, the viewpoint. Here, the viewpoint is related to the application of the classification, which is based on what the classification will be used for. For example, it could be based on the clustered trips' frequency, distance, or duration. Figure 6 exemplifies this driver's days classification method by using three different viewpoints: absolute frequency, weighted frequency based on distance, and weighted frequency based on time duration. In the frequency-based classification, just the number of occurrences of each cluster in a day is accounted for, with the classification of a day being the cluster with the most trips in a day. This classification method could be attractive for applications related to short trips. On

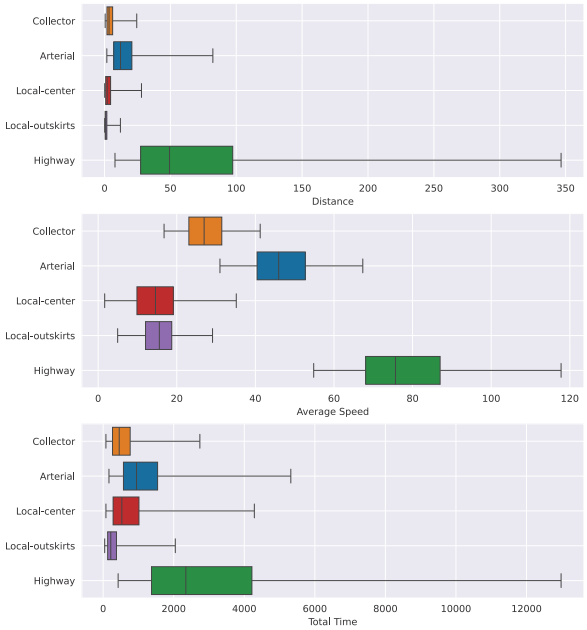

Figure 4: Boxplot distributions of the cluster features.

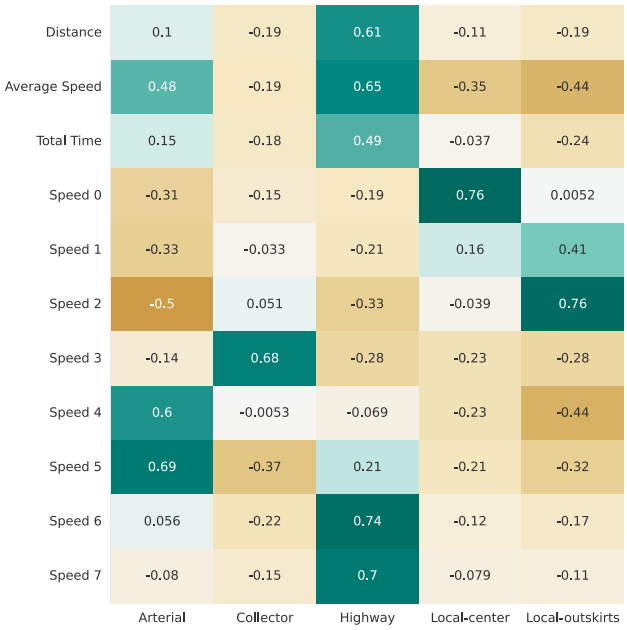

Figure 5: Correlation between the features and clusters.

the other hand, the distance and time-based ones could be better for longer and lengthier trips. In those cases, the weight of each trip is the distance or total time of that trip, making longer trips more important in the classification. This increased importance of longer trips makes sense when the analysis is focused on energy consumption. For our energy consumption analysis, we will be using the classification with the weighted frequency based on distance, as the energy consumption is totally related to the distance traveled.

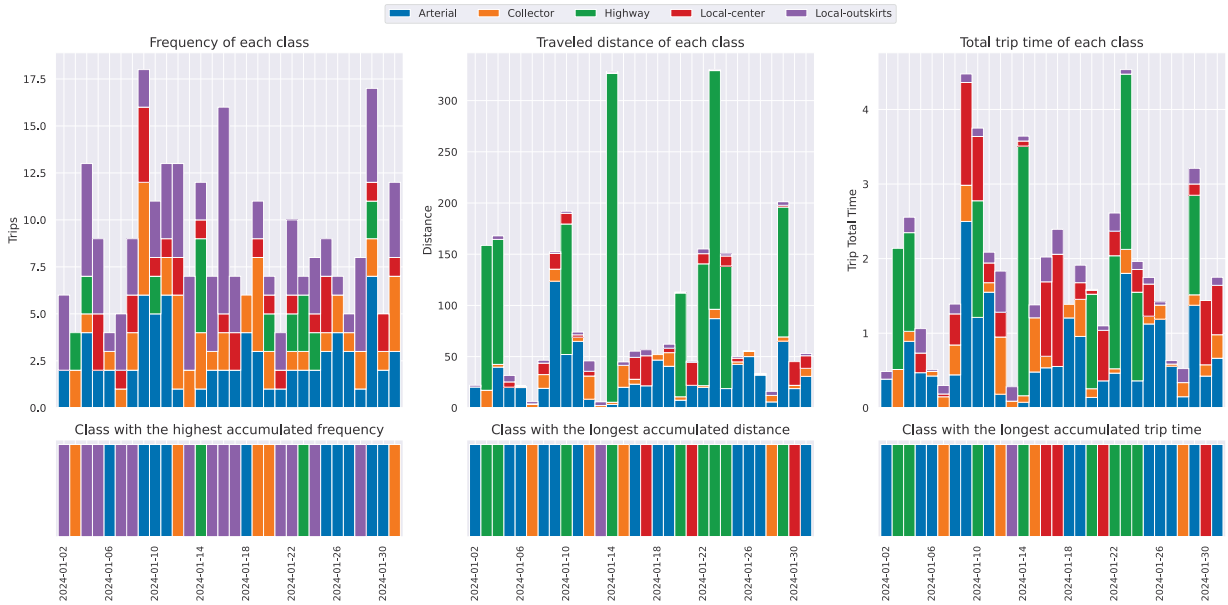

Figure 6: Example of the different ways a day can be classified.

5. Results and Discussions

In this analysis, we will be constantly comparing two possibilities of battery energy usage: regular usage without in-between trips recharge, and with in-between trips recharge. The energy data for both possibilities are obtained with the energy efficiency conversion function. For the data without recharge, we created datasets like the ones from Tables 3 and 5, and for the data with recharge, we created datasets like Table 6 for a 10kWh to 120kWh range of battery sizes. Also, a maximum time of recharging of 10 minutes per stop was imposed, with 2 minutes needed to start the recharging process. For both cases we apply the context-based clustering to the data, analysing how the context, road type, affects energy usage and efficiency.

5.1. Energy Consumption and Thresholds

Using the energy efficiency conversion function and setting energy thresholds ranging from 1 to 120 kWh, we can make graphs such as Figure 7. In this graph, using the calculated percentage of days covered, i.e., the percentage of days in which the energy used was less than the battery size, we calculated for each battery size the percentage of vehicles covered. For example, in Figure 8 we have a closer look at the graph from Figure 7 and it is possible to see that for a battery size of 60 kWh, in the slightly darker green range, 90% of the vehicles were covered 95% of the days, i.e., there was at least one vehicle on this 90% which used more than 60 kWh of energy in 5% of its days.

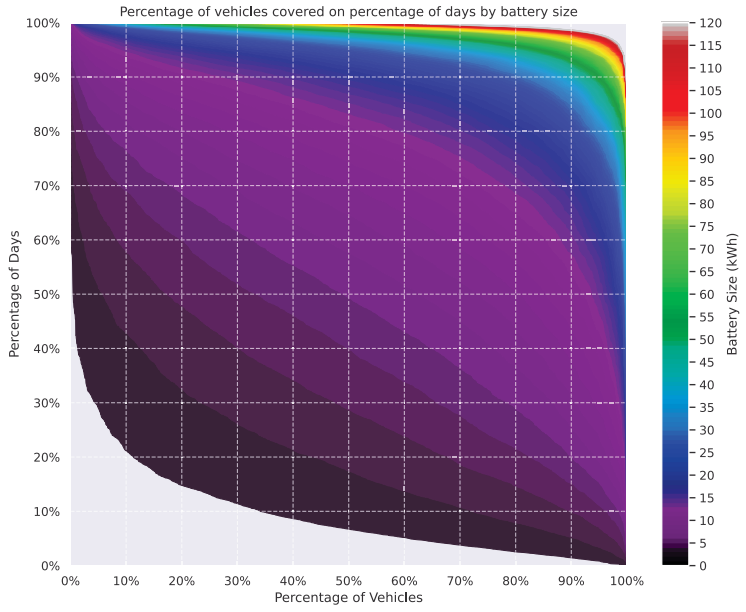

Figure 7: Vehicular and time coverage heat map.

For the case with recharging, we count a day as covered if the vehicle SoC did not get lower than 0% for that vehicle on that day. Figure 9 is the same type of graph as figures 7 and 8 but with the new SoC calculation. Here it is already noticeable how the introduction of recharging increased the vehicles by days coverage curves. Looking at the 40 kWh battery size, in the light blue range, before, around 70% of the vehicles were covered 95% of the days, with the recharging in the same percentage of days around 80% can be covered, an increase of 10% in vehicular coverage.

Another way we can analyse the daily energy consumption of the vehicles is with graphs like the one in Figure 10. In this graph, we have the energy consumption by the percentage of vehicles throughout many ranges of days. For example, it is possible to see that more than half of the vehicles didn't even hit the 100 kWh mark in a single day in a whole year. However, it also shows that around 5% of the vehicles used at least 100 kWh of energy in 10 days. This graph demonstrates that almost all vehicles consume a small amount of energy for most of the days and that the vast majority of drivers just consume a lot of energy in very few days in a year. Also, there is a small percentage of vehicles that consume a lot of energy in a considerable number of days.

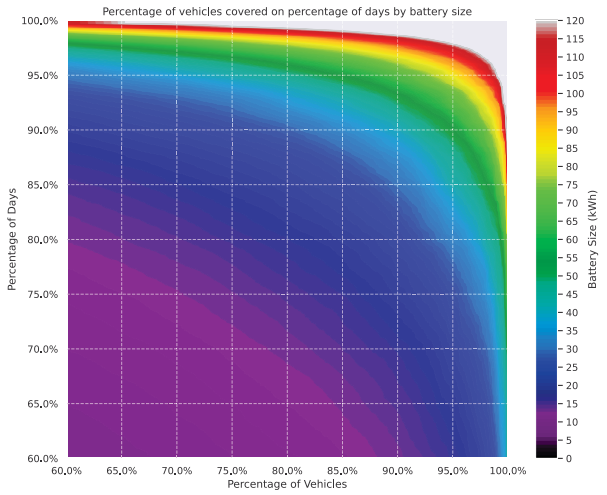

Figure 8: Zoom on a section of Figure 7.

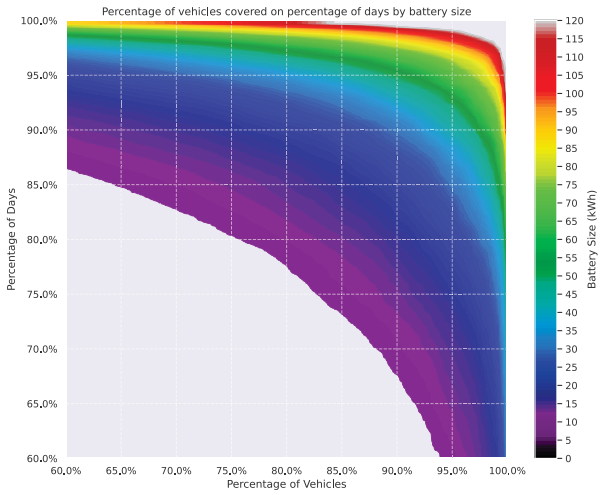

Figure 9: Vehicular and time coverage heat map applying the recharging.

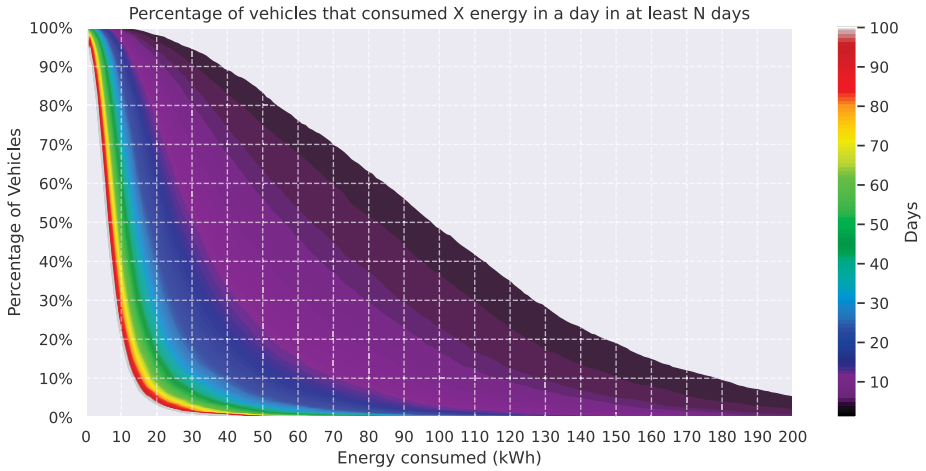

Figure 10: Vehicular energy consumption by days heat map.

From the energy consumption information in Figure 10 two points can already be raised: considering a worst-case scenario of no in-between trips recharging, a medium to big size battery, 60 to 80 kWh, would be more than enough for regular usage of the vehicles by the vast majority of the drivers; it does not make sense to increase battery size with the objective of covering all days for most vehicles, as energy usage values can get really high for low numbers of days.

Although very informative, the range of percentage of days covered in the graphs from figures 7, 8 and 9 are too big. By setting some threshold values for what percentage of days we need to cover, we can make graphs such as the one in Figure 11, relating vehicular coverage and battery size. With this graph, it is possible to see how difficult it gets to cover more vehicles for percentages of days covered closer to 100%, with really big batteries of 120 kWh covering just about 65% of the vehicles on all the days without recharging and 84% with recharging. This percentage without recharge corroborates with the daily consumptions from Figure 10, which shows that for 1 day and 100% days coverage, 35% of drivers consumed more than 120 kWh, which are the drivers excluded from the 65% from Figure 11. Figure 11 also demonstrates how much the addition of recharging increases vehicular coverage, with the most noticeable differences occurring on low percentages of days covered as 80% and 90%, and small batteries as 10 to 20 kWh, and on high percentages of days covered as 99% and 100%, and big batteries as 60 to 120 kWh.

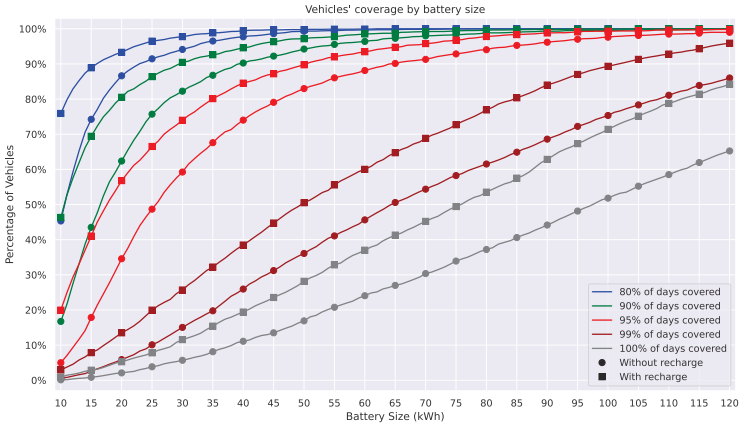

Figure 11: Vehicular coverage on specific percentages of days covered.

Given the difficulty of covering all days strictly, as in the 100% of days coverage a vehicle counts as not covered if just a single trip surpasses the battery size, we made a less strict coverage of days rule. This new rule ignores a single day that was not covered in a whole year. Figure 12 compares the days' coverage when we set it as not covered when just a single day was not covered and when we ignore one day. Figure 13 shows the difference for both cases, without and with recharging, of ignoring a single day. This comparison shows the percentage of vehicles that are not covered in just one day for each battery size. Both curves for the cases with and without recharging are very similar but the one with recharging is always higher until the 85 kWh battery size mark. After this mark, the percentage of vehicles in the case without recharging stays constant while it starts to drop in the case with recharging. This sudden drop can be explained by the types of days that need bigger battery sizes. As with all the data we have, those days were driven on ICEVs which do not need recharging stops. With this lack of stops, there is no opportunity for an in-between trip recharge to occur. Another important piece of information on the curves from Figure 13 is the huge number of vehicles that have just one big day in a year, further confirming that traveling long distances in a day is the exception for most drivers, not the rule. Given this increase in coverage by ignoring only one day in the year, from here on out we will be using this less strict rule.

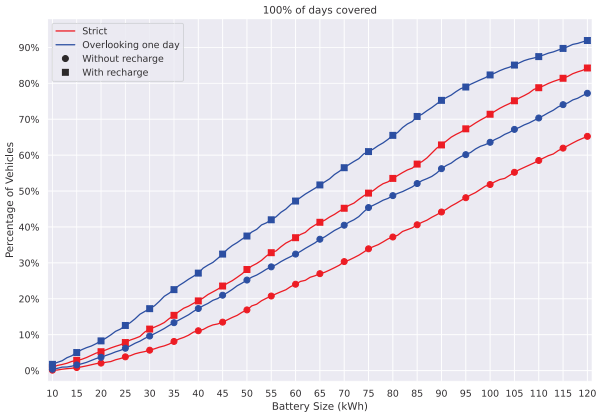

Figure 12: Vehicular coverage while covering all days.

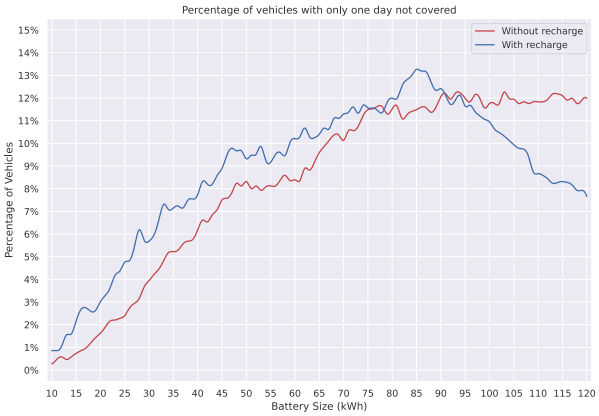

Figure 13: Difference between the strict and not strict rule.

Using the rule of overlooking one day for 100% of days covered, we can directly compare the cases without and with recharging. Figures 14 and 15 show this comparison from two different viewpoints: percentage of vehicular coverage and battery size. Figure 14 compares the percentage of vehicular coverage by calculating the difference in these values between the cases with and without recharging. The curve is always positive and peaks for a battery size

of 89 kWh at around a 19% difference, which means that the recharging case is always better and that the battery size where the recharging is the worthiest is 89 kWh. Also, for more standard battery sizes such as 40, 60, and 80 kWh the increase in the percentage of vehicular coverage is around 10, 15, and 17%.

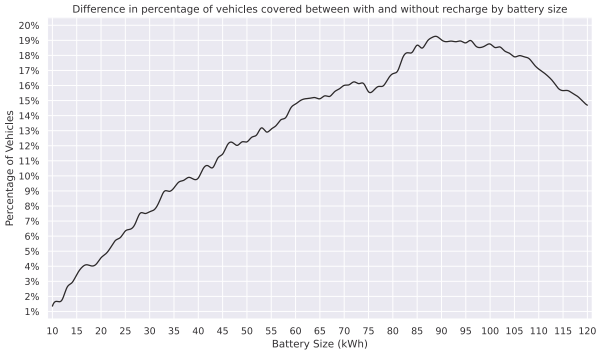

Figure 14: Vehicular coverage comparison between recharging and not recharging.

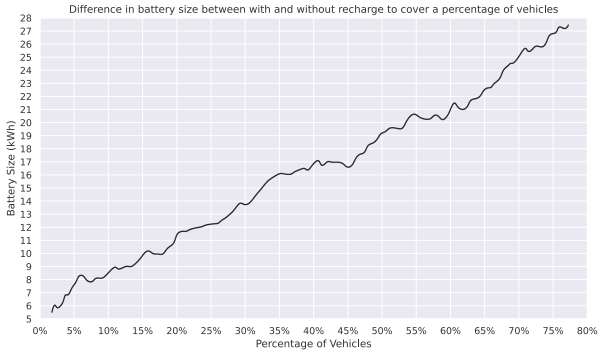

Figure 15: Battery size comparison between recharging and not recharging.

Figure 15 has a viewpoint focused on battery size, which can be more easily translated to cost. This graph shows the difference in battery size needed to cover different percentages of vehicles. For example, if it is decided that 70% of the vehicles need to be covered in all days, Figure 15 shows that there is a 25 kWh difference between recharging and not recharging the vehicles. This means that just by having some in-between trips recharging it is possible to make a vehicle with a battery 25 kWh smaller. Supposing a battery production cost of $115.00 per kWh [39], this would save $2,875.00 in production cost. Unfortunately, as the percentage of vehicles in this graph is limited to the lower bound in the comparison, the curve without recharging, we only have values between 0% and 80%. However, this interval already demonstrates the gains of an infrastructure that allows for easy recharging of EVs.

5.2. Context-Based Classification

All Those figures give us a general vision of the data. However, it is well known that the trips that are forcing us to have a big battery to increase vehicular coverage are the low-frequency long-distance trips. Those trips can be clustered from the others by applying our context-based clustering. In our clustering, they are recognized as "Highway" trips. We clustered our data and did the classification process from Figure 6 using the distance viewpoint. Table 8 shows how the days were separated, with 14% of the total number of days from all vehicles being classified as "Highway".

	Context				
	Arterial	**Collector**	**Highway**	**Local-center**	**Local-outskirts**
Classification Frequency (%)	42.97	28.69	14.16	7.45	6.73

Table 8: Frequency distribution of the classification of the days by context.

In Figure 16 we have graphs of energy consumption by percentage of vehicles over a range of days, like the one from Figure 10 but now with the data separated on the different contexts. These graphs in Figure 16 make clear when energy is most consumed, with what we will be calling the totally urban days, "Collector" and "Local" contexts, having less than 3% of the vehicles consuming more than 30 kWh in at least one day. On the other hand, for days classified as "Arterial" this percentage is up to 50%, and for days classified as "Highway" it goes to around 93% of the drivers. Considering the amount of energy of medium size batteries, 60 kWh, the totally urban days can be totally covered, while around 7% of the "Arterial" days would still be uncovered and almost 75% of the "Highway" days would not be able to be completed. This shows the gap in energy usage of urban contexts to the highway context, and how it is a reasonable choice to separate them when considering different battery sizes.

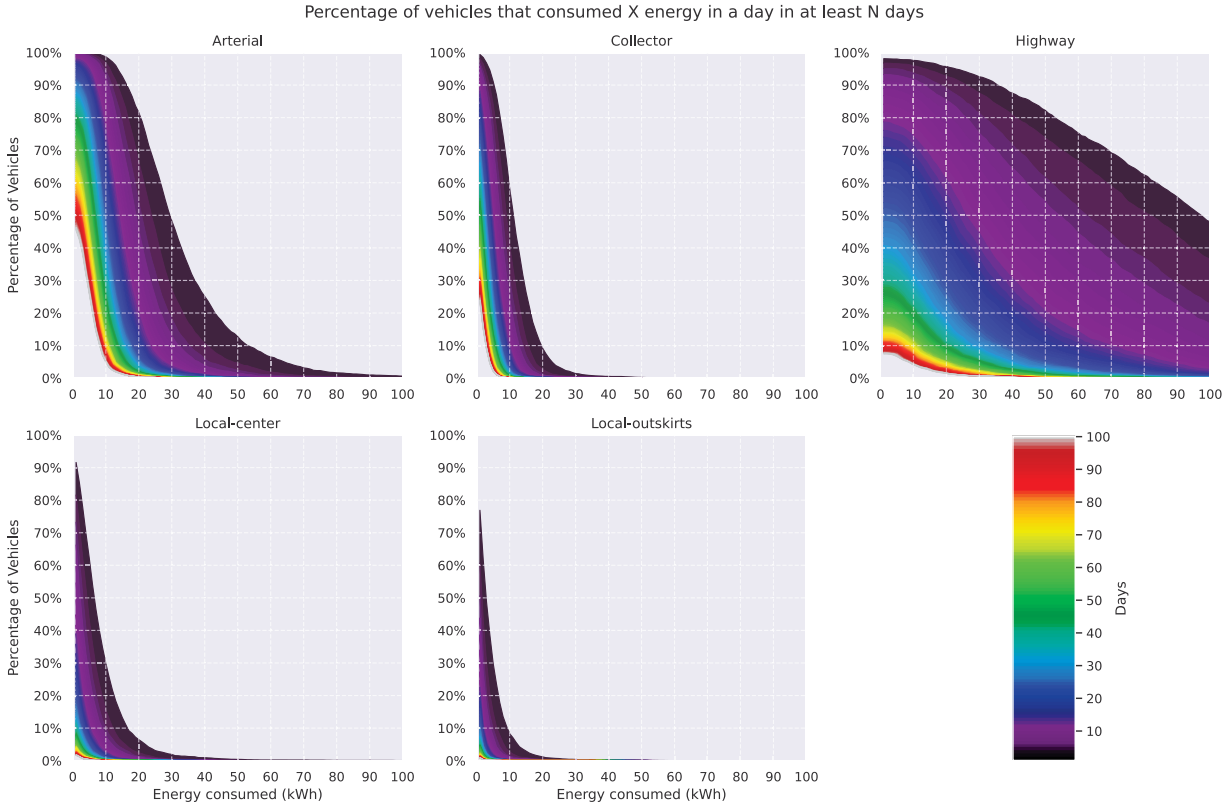

Figure 16: Vehicular energy consumption by days with context division heat map.

Setting the days coverage as 100% and using the less strict rule of ignoring one day not covered in a year, we built battery size by vehicular coverage graphs separated by the context-based classification. Figures 17 and 18 show these curves for each context and for all the data, and for both the with and without recharging cases. These graphs make clear how easy it is to cover the totally urban days, as 30 and 20 kWh batteries cover around 100% of the vehicles for the without and with recharging cases, respectively. Looking at the frequencies in Table 8, this means that at the very least 43% of all the days can be covered with 30 or 20 kWh batteries. For days that are between urban and highway, "Arterial" context, 30 kWh batteries can cover 73% or 93% of the vehicles, with a 60 kWh battery covering around 98% or 100% of the vehicles. However, although we can cover at least 86% of all the days with a 60 kWh battery, the "Highway" days are still unreachable, with this battery size just covering 32% or 46% of the vehicles.

An interesting behaviour of the "Highway" and "All data" lines can be observed in Figure 18. Around 90 kWh battery size there is a small change in the tangential angle of the curves. For battery sizes smaller than around 90 kWh, the curves can be approximated by a linear function with an angular coefficient of 1, e.g., for each 1 kWh increase in battery size there is a 1% increase in vehicular coverage. However, for battery sizes bigger than around 90 kWh, the angular coefficient of the approximated linear function changes to 0.5. This means that increasing the battery by the same amount after around 90 kWh only results in half of the gains in vehicular coverage.

Comparing the results from figures 17 and 18 really demonstrates how recharging can decrease battery size or increase coverage. Figures 19 and 20 make the differences of these cases clearer. Figure 19 compares the percentage of vehicular coverage of the graphs from figures 17 and 18. This graph demonstrates that recharging is always better, as its curves are always positive, but it also shows for which battery sizes the recharging is more worthy. For example, for the total urban contexts, the peak in vehicular coverage gain is at the smallest size, 10 kWh batteries, with "Local-center", "Local-outskirts", and "Collector" days gaining around 2%, 11%, and 35%, respectively. For the "Arterial" days the maximum gain is at 19 kWh batteries, with a gain of around 41% in vehicular coverage. And for the "Highway" days, its curve is similar to the one from Figure 14, which peaks at a battery size of 89 kWh, with

a gain of around 19% in vehicular coverage.

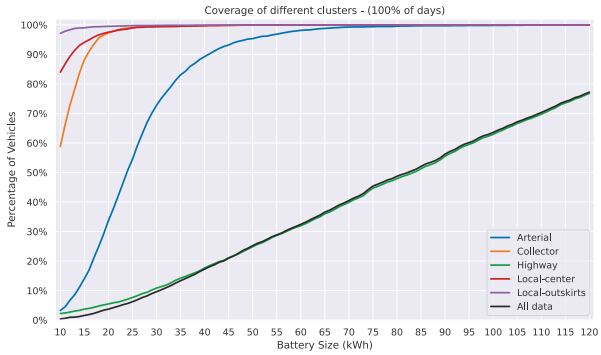

Figure 17: Vehicular coverage for all days with context segregation and without recharging.

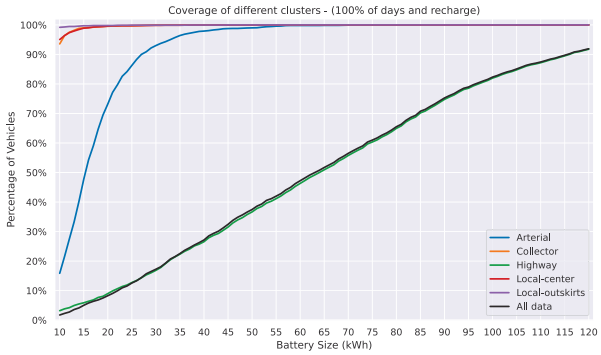

Figure 18: Vehicular coverage for all days with context segregation and with recharging.

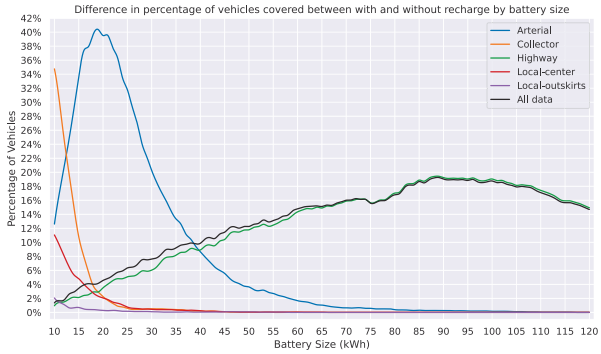

Figure 19: Vehicular coverage comparison between recharging and not recharging for different contexts.

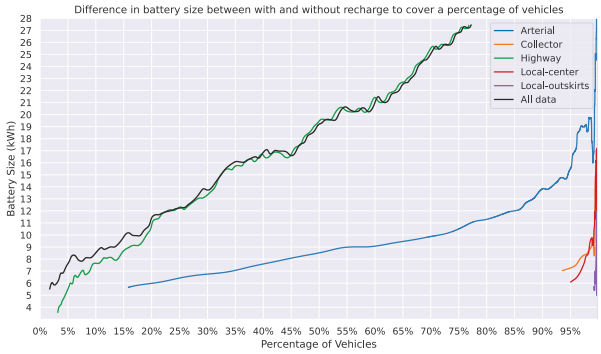

Figure 20: Battery size comparison between recharging and not recharging for different contexts.

Figure 20 also compares the cases without and with recharging but looks at the difference in battery sizes. Its graph gives the gain of recharging in battery size given a percentage of vehicles that must be covered. For example, it is noticeable how the curve for the “Arterial” days has the longest domain of percentage of vehicular coverage and how it is almost always increasing, with high values at high percentages. If it is decided that is necessary to build an EV with a battery that covers 95% of the vehicles for all “Arterial” days, Figure 20 shows that there is a 16 kWh difference in the battery size needed, or a decrease of $1.840,00 in production cost, if the vehicle will or will not be recharged between its trips.

Instead of looking at the vehicular coverage, i.e., the percentage of vehicles that had all their days covered by a specific battery size, we can look at the days’ coverage or the opposite of that. The first two graphs from Figures 21 show the percentage of days that could not be covered by a range of battery sizes, for all the different contexts and for all the data, for the cases with and without recharging. The coverage of days axis in these graphs is in the logarithmic scale, which allows us to see with more detail the minor gains, consequently allowing for better decisions. For example, in the “Arterial” curve of the graph with recharging, a 99.9% coverage of the days can be achieved with a 41 kWh battery, and a 99.99% coverage of the days can be achieved with a 65kWh. However, would an increase in days covered of 0.09% be worth the cost of a 24 kWh bigger battery? Most probably not. Even more when we analyse the differences between with and without recharging. The last graph of Figure 21 demonstrates this difference, showing the percentage of days covered gained from recharging compared to the case without recharging. This graph shows the impressive gains from recharging for small battery sizes, less than 30 kWh, for all contexts. Also, for the “Highway” days the days gained remain high even for big battery sizes, demonstrating that, even with just the

in-between trips stops that drivers already do with their ICEVs, the addition of in-between trips recharging increases a lot the coverage of high distance days.

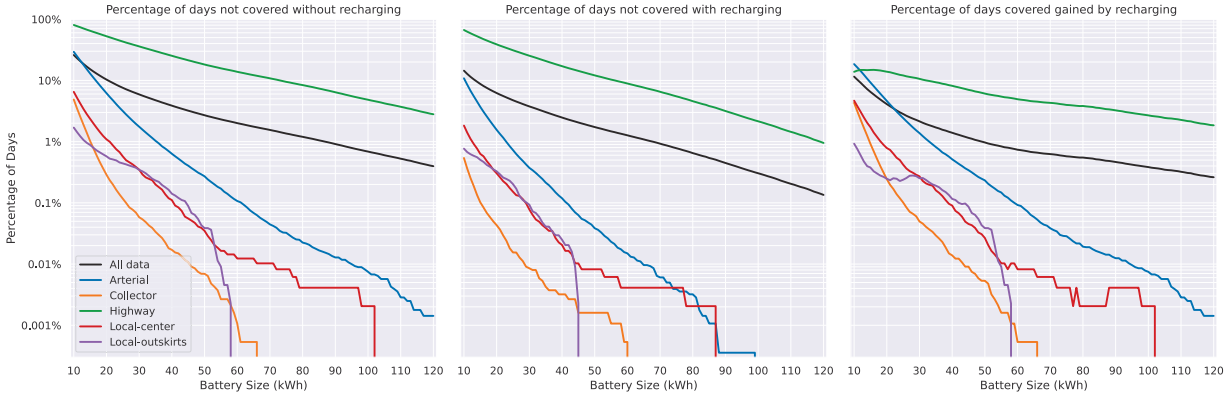

Figure 21: Total of days not covered by battery size and context, and comparison between the cases without and with recharging.

The graph of Figure 22 compares the results in the graphs of Figure 21 with a focus on the difference in battery sizes. For each context and for all data this figure shows how much bigger a battery would need to be to cover the same percentage of days but without the in-between trips recharging. This comparison makes much clearer the gains from the application of recharging. For the realistic percent off day range, where increase in coverage still has a significant difference, i.e., 0.01 to 100%, the days in the "Arterial" context show the biggest difference, with a peak of 30 kWh difference in battery size to cover the same 99.99% of the days. In this case, while a 65 kWh battery can cover 99.99% of the days with recharging, a huge 95 kWh battery would be needed if recharging is not applied.

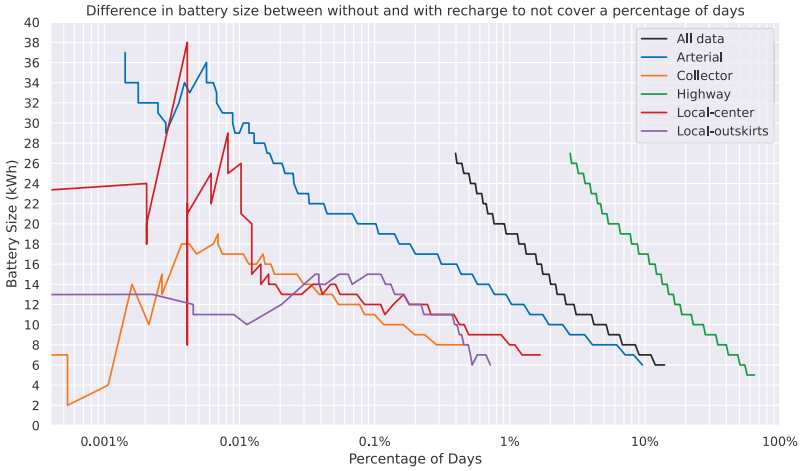

Figure 22: Battery size comparison between the cases without and with recharging for a same percentage of days covered.

6. Conclusion

In recent years the production of EVs has grown non-stop. This growth can be mainly associated with environmental and technological factors. Year after year the impacts of global warming are getting more tangible, forcing countries to attack what they conclude as the main cause: CO2 emission [11]. Regulations on combustion engines pressure the automotive industry into changing their products, increasing even further the rate of idealization and

production of EVs. With the necessity of increasing the production of EVs, advances in manufacturing technologies that allow for faster and cheaper production of this somewhat new technology are pursued.

This increase in EV production is happening alongside another transformation in the automotive industry: vehicle connectivity. The many advantages vehicular connectivity brings to both the drivers and manufacturers [7, 8] are the reasons why the industry is increasingly including connectivity on new vehicles. Moreover, vehicular connectivity is mainly related to the digitalization of vehicles, which is the default on EVs, making the implementation of vehicular connectivity on EVs even more natural.

It is from the data collected from connected vehicles that we answer one of the biggest questions in today's automotive industry: what is the ideal EV battery size? With our main focus being drivers who are and will need to eventually migrate from ICEVs to EVs. To find an answer, we analysed the trips from many different vehicles of an ICEV model, getting simulated energy consumption values by using a conversion function. As the increase in the circulation of EVs caused an increase in public recharging infrastructure, two scenarios were tested: without and with in-between trips recharging. Another important factor that was accounted for is the difference in energy usage when the vehicle is in different contexts, mainly between urban and highway trips. This was included by applying a context-based clustering method that can cluster the trips into different types of roads.

The various graphs showing energy usage, and coverage of vehicles and days for different battery sizes, scenarios, and contexts led to interesting conclusions. The first and perhaps the most important one is that just increasing battery size is not the right answer, yet. Even without accounting for the massive increase in weight and reduction in internal space that a big battery would cause, our analyses showed that even the biggest proposed size, 120 kWh, was not nearly enough to make the longest days possible, which were few but existed for a big share of the analysed vehicles. This conclusion is further confirmed by the days classified as "Highway" in our context-segregated analysis. These days were the only ones that could not be properly covered by reasonably sized batteries. This makes the hypothesis of making EVs that can compete with ICEVs in the range of trips without stops impossible for vehicles in the same category, as the current battery technology still cannot compete with the energy density of fossil fuels. This conclusion is for the current technology, as batteries' energy density continues to increase in recent years [40, 41].

The comparisons between the scenarios with and without in-between trips recharging is what gave us another, less hopeless, conclusion: there are significant gains from short recharging stops. Even if an unreasonably huge battery still cannot achieve a bigger range than a tank fuel of gas, if a smaller, more realistic, battery can be well recharged in a short amount of time, the same distance can be achieved in similar times. In our tests, even though a fairly short duration for the recharging stops was used, with the objective of not upsetting previously ICEVs drivers, the gains from it were still pretty clear in the graphs, independently from the context. This demonstrated that a smaller battery, cheaper to produce, lighter, and occupying less space, can achieve the same results as a bigger battery if occasionally recharged between trips. However, for this to be possible a good infrastructure of public fast recharges is needed. Currently, such ideal infrastructure only exists inside urban centers of well-developed countries. Nevertheless, their numbers are growing following the huge growth in EVs' production [10].

Following the conclusion that reasonable battery sizes with in-between trips recharging is the best option in opposition to a huge battery that wouldn't require stops, we can suggest some sizes given the study's results. The context segregation showed that covering urban days is fairly easy, thus, if a small vehicle for mostly the urban market is being made, a battery of around 40 kWh would be sufficient, as it can cover all urban days and almost all the "Arterial" days, which are between urban and highway. For the general market, which encompasses most drivers, an adequate size would be around 60 kWh, as it can easily cover all non-highway days allowing for comfortable urban usage, cover almost half of the vehicles in highway days, and cover around 99% of all days. Lastly, to maximize range without being unreasonable, a battery of around 90 kWh could be ideal, as it does everything that a 60 kWh one would do but increasing its coverage of vehicles on highway days to 75%. Furthermore, looking at the "Highway" curve from Figure 18, after this battery size the slope of the curve starts to drop, making further increases in vehicular coverage less cost-efficient. Moreover, in Figure 19 around 90 kWh is where the coverage of vehicles gained from doing the in-between trips recharging is maximized.

We compared in different contexts a hypothesis of no recharging and one with a specific recharging strategy. After our analysis, we confirmed that the best and most reasonable strategy for EV batteries is to apply a recharging strategy. However, only one specific recharging strategy was tested, where in every stop lengthier than 2 minutes a maximum of 10 minutes of fast recharging could be done. To further consolidate our analysis of EV batteries, a future study could be done comparing different recharging strategies that could be employed, which could include different stop

times and frequencies, charger speeds, and battery sizes.

Acknowledgment

The authors thank Renault from Brazil and Araucária Foundation for the partnership and financial support in the development of this research.

References

[1] C. Llopis-Albert, F. Rubio, F. Valero, Impact of digital transformation on the automotive industry, Technological Forecasting and Social Change 162 (2021). doi:doi:10.1016/j.techfore.2020.120343.

[2] S. L. Bangare, S. Prakash, K. Gulati, B. Veeru, G. Dhiman, S. Jaiswal, The architecture, classification, and unsolved research issues of big data extraction as well as decomposing the internet of vehicles (iov), in: 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 566–571. doi:doi:10.1109/ISPCC53510.2021.9609451.

[3] R. Bohnsack, H. Kurtz, A. Hanelt, Re-examining path dependence in the digital age: The evolution of connected car business models, Research Policy 50 (2021). doi:doi:10.1016/j.respol.2021.104328.

[4] F. Kuhnert, C. Stürmer, A. Koster, Five trends transforming the automotive industry, 2018. URL: https://www.pwc.com/gx/en/industries/automotive/assets/pwc-five-trends-transforming-the-automotive-industry.pdf.

[5] K. L. Lim, J. Whitehead, D. Jia, Z. Zheng, State of data platforms for connected vehicles and infrastructures, Communications in Transportation Research 1 (2021). doi:doi:10.1016/j.commtr.2021.100013.

[6] M. Swan, Connected car: Quantified self becomes quantified car, Journal of Sensor and Actuator Networks 4 (2015) 2–29. doi:doi:10.3390/jsan4010002.

[7] A. Karmańska, The benefits of connected vehicles within organizations, Procedia Computer Science 192 (2021) 4721–4731. doi:doi:10.1016/j.procs.2021.09.250.

[8] G. Abdelkader, K. Elgazzar, A. Khamis, Connected vehicles: Technology review, state of the art, challenges and opportunities, Sensors 21 (2021). doi:doi:10.3390/s21227712.

[9] A. Athanasopoulou, M. de Reuver, S. Nikou, H. Bouwman, What technology enabled services impact business models in the automotive industry? an exploratory study, Futures 109 (2019) 73–83. doi:doi:10.1016/j.futures.2019.04.001.

[10] IEA, Global ev data explorer, 2024. URL: https://www.iea.org/data-and-statistics/data-tools/global-ev-data-explorer.

[11] IEA, Global ev policy explorer, 2024. URL: https://www.iea.org/data-and-statistics/data-tools/global-ev-policy-explorer.

[12] DOE, Electric vehicle battery pack costs for a light-duty vehicle in 2023 are 90estimates, 2024. URL: https://www.energy.gov/eere/vehicles/articles/fotw-1354-august-5-2024-electric-vehicle-battery-pack-costs-light-duty.

[13] IRENA, Critical materials: Batteries for electric vehicles, IRENA - International Renewable Energy Agency (2024). URL: https://www.irena.org/Publications/2024/Sep/Critical-materials-Batteries-for-electric-vehicles.

[14] S&P, Global battery market: First movers will likely keep their leads, 2024. URL: https://www.spglobal.com/_assets/documents/ratings/research/101606071.pdf.

[15] Y. Kwon, S. Son, K. Jang, User satisfaction with battery electric vehicles in south korea, Transportation Research Part D: Transport and Environment 82 (2020). doi:doi:10.1016/j.trd.2020.102306.

[16] K. M. Buhmann, J. R. Criado, Consumers' preferences for electric vehicles: The role of status and reputation, Transportation Research Part D: Transport and Environment 114 (2023). doi:doi:10.1016/j.trd.2022.103530.

[17] A. Pamidimukkala, S. Kermanshachi, J. M. Rosenberger, G. Hladik, Evaluation of barriers to electric vehicle adoption: A study of technological, environmental, financial, and infrastructure factors, Transportation Research Interdisciplinary Perspectives 22 (2023). doi:doi:10.1016/j.trip.2023.100962.

[18] A. Pamidimukkala, S. Kermanshachi, J. M. Rosenberger, G. Hladik, Barriers and motivators to the adoption of electric vehicles: A global review, Green Energy and Intelligent Transportation 3 (2024). doi:doi:10.1016/j.geits.2024.100153.

[19] Y. Al-Wreikat, C. Serrano, J. R. Sodré, Effects of ambient temperature and trip characteristics on the energy consumption of an electric vehicle, Energy 238 (2022). doi:doi:10.1016/j.energy.2021.122028.

[20] J. Taggart, Ambient temperature impacts on real-world electric vehicle efficiency & range, in: 2017 IEEE Transportation Electrification Conference and Expo (ITEC), 2017, pp. 186–190. doi:doi:10.1109/ITEC.2017.7993269.

[21] P. Iora, L. Tribioli, Effect of ambient temperature on electric vehicles' energy consumption and range: Model definition and sensitivity analysis based on nissan leaf data, World Electric Vehicle Journal 10 (2019). doi:doi:10.3390/wevj10010002.

[22] H. Yu, Y. Liu, J. Li, T. Fu, Investigation of energy consumption characteristics of electric passenger car under high and low temperature conditions, in: 2020 5th Asia Conference on Power and Electrical Engineering (ACPEE), 2020, pp. 742–746. doi:doi:10.1109/ACPEE48638.2020.9136212.

[23] X. Hao, H. Wang, Z. Lin, M. Ouyang, Seasonal effects on electric vehicle energy consumption and driving range: A case study on personal, taxi, and ridesharing vehicles, Journal of Cleaner Production 249 (2020). doi:doi:10.1016/j.jclepro.2019.119403.

[24] S. Sagaria, R. C. Neto, P. Baptista, Modelling approach for assessing influential factors for ev energy performance, Sustainable Energy Technologies and Assessments 44 (2021). doi:doi:10.1016/j.seta.2020.100984.

[25] A. Donkers, D. Yang, M. Viktorović, Influence of driving style, infrastructure, weather and traffic on electric vehicle performance, Transportation Research Part D: Transport and Environment 88 (2020). doi:doi:10.1016/j.trd.2020.102569.

[26] N. Jones, S. Nazarenus, K. Stamatis, D. Potoglou, A. Zachariah, L. Cipcigan, Optimisation of electric vehicle battery size, Transportation Research Procedia 70 (2023) 338–346. doi:doi:10.1016/j.trpro.2023.11.038, 8th International Electric Vehicle Conference (EVC 2023).

[27] N. S. Pearre, W. Kempton, R. L. Guensler, V. V. Elango, Electric vehicles: How much range is required for a day's driving?, Transportation Research Part C: Emerging Technologies 19 (2011) 1171–1184. doi:doi:10.1016/j.trc.2010.12.010.

[28] E. D. Kostopoulos, G. C. Spyropoulos, J. K. Kaldellis, Real-world study for the optimal charging of electric vehicles, Energy Reports 6 (2020) 418–426. doi:doi:10.1016/j.egyr.2019.12.008.

[29] X. Hao, Z. Lin, H. Wang, S. Ou, M. Ouyang, Range cost-effectiveness of plug-in electric vehicle for heterogeneous consumers: An expanded total ownership cost approach, Applied Energy 275 (2020). doi:doi:10.1016/j.apenergy.2020.115394.

[30] M. Etxandi-Santolaya, L. Canals Casals, C. Corchero, Estimation of electric vehicle battery capacity requirements based on synthetic cycles, Transportation Research Part D: Transport and Environment 114 (2023). doi:doi:10.1016/j.trd.2022.103545.

[31] X. Zhang, Y. Zou, J. Fan, H. Guo, Usage pattern analysis of beijing private electric vehicles based on real-world data, Energy 167 (2019) 1074–1085. doi:doi:10.1016/j.energy.2018.11.005.
[32] J. Zhang, Z. Wang, P. Liu, Z. Zhang, Energy consumption analysis and prediction of electric vehicles based on real-world driving data, Applied Energy 275 (2020). doi:doi:10.1016/j.apenergy.2020.115408.
[33] S. Árpád Funke, P. Plötz, M. Wietschel, Invest in fast-charging infrastructure or in longer battery ranges? a cost-efficiency comparison for germany, Applied Energy 235 (2019) 888–899. doi:doi:10.1016/j.apenergy.2018.10.134.
[34] A. Popiolek, Z. Dimitrova, J. Hassler, M. Petit, P. Dessante, Comparison of decentralised fast-charging strategies for long-distance trips with electric vehicles, Transportation Research Part D: Transport and Environment 124 (2023). doi:doi:10.1016/j.trd.2023.103953.
[35] F. Naseri, C. Barbu, T. Sarikurt, Optimal sizing of hybrid high-energy/high-power battery energy storage systems to improve battery cycle life and charging power in electric vehicle applications, Journal of Energy Storage 55 (2022). doi:doi:10.1016/j.est.2022.105768.
[36] V. Salapić, M. Gržanić, T. Capuder, Optimal sizing of battery storage units integrated into fast charging ev stations, in: 2018 IEEE International Energy Conference (ENERGYCON), 2018, pp. 1–6. doi:doi:10.1109/ENERGYCON.2018.8398789.
[37] P. Félix, L. A. Roque, I. Miranda, A. Gomes, Battery energy storage system optimal sizing in a battery electric vehicle fast charging infrastructure, U.Porto journal of engineering. 9 (2023). doi:doi:10.24840/2183-6493_009-005_001937.
[38] F. Aksan, V. Suresh, P. Janik, Optimal capacity and charging scheduling of battery storage through forecasting of photovoltaic power production and electric vehicle charging demand with deep learning models, Energies 17 (2024). doi:doi:10.3390/en17112718.
[39] BloombergNEF, Lithium-ion battery pack prices see largest drop since 2017, falling to $115 per kilowatt-hour, 2024. URL: https://about.bnef.com/blog/lithium-ion-battery-pack-prices-see-largest-drop-since-2017-falling-to-115-per-kilowatt-hour-bloombergnef
[40] PhysicsWorld, Lithium-ion batteries break energy density record, 2023. URL: https://physicsworld.com/a/lithium-ion-batteries-break-energy-density-record/.
[41] Q. Li, Y. Yang, X. Yu, H. Li, A 700 wh/kg rechargeable pouch type lithium battery, Chinese Physics Letters 40 (2023). doi:doi:10.1088/0256-307X/40/4/048201.

6 CONCLUSION

This work presented a compilation of four papers that demonstrate the applications of Machine Learning (ML) (Sarker, 2021), Process Mining (PM) (Aalst, 2016), and Data Science (Kelleher; Tierney, 2018) and Analysis (Kudyba, 2014) in very distinct areas of engineering: production planning and control, and vehicular connectivity.

Our motivation for this work was to demonstrate how unrelated problems, from vastly different areas of engineering, can be solved with modern data-driven methods. This compilation of four different studies shows how the field of data-related methods has unlimited applicability. If there is data related to the problem, methods can be used to investigate, diagnose, and solve it.

In this work, the first problem explored was in the area of production planning and control. Motivated by the necessity of better control of the plant and planning of future production orders, the remaining time of the production orders is explored. Having access to accurate remaining time information of production orders can optimize production in manufacturing plants, as production managers are able to correctly allocate production orders and provide clients with accurate completion times. The proposed solution to get accurate remaining times was to use data-driven prediction methods, from ML and PM, and data from the manufacturing process to build prediction models. However, another problem that appears when trying to predict the remaining times is the variation caused by the production of different products. Even if two products pass by the same process, here defined by the machines used, their times won't necessarily be the same. For example, the milling process to make spur gears with different numbers of teeth will result in different processing times. To solve this, individual prediction models for each product are proposed, making the models product-oriented. In the two first papers, we study these problems and present the solutions.

To obtain the remaining time information from the data in the form of manufacturing logs, PM techniques were applied. These techniques use the logs of the production, called event logs (Aalst, 2016). To accurately predict the remaining time of the production orders both papers presented data-driven methods based

on ML and PM, and also a hybrid (Choueiri et al., 2020) and a baseline method. The first paper compared two ML, one PM, a hybrid, and a baseline method. In the second paper, this was further expanded and improved, with the addition of two ML methods and the improvement of the baseline method.

To compare the methods and demonstrate their usefulness in predicting the remaining times, various tests were performed comparing across different test data and validation metrics. In the first paper, artificially generated event logs were created. These logs simulated the manufacturing process of different products, passing through machines with different processing times. These artificial logs also simulated activity rework, which included further complexity to the logs. In the second paper, the artificial logs were further expanded with the addition of a product and machines, and an event log from a real manufacturer was analyzed and used.

The results in the first paper demonstrated the potential of the presented methods, with all models making accurate predictions much better than the baseline and demonstrating how the different probabilities of rework affected the models. The results in the second paper complement the ones from the first. Besides the two new ML methods and the improved baseline method, the tests on the event logs of the real manufacturer exposed how the models behave for data generated from different origins. While most of the models performed well in all tests with the artificial logs, most models underperformed in the real logs. The second paper demonstrated the importance of choosing the right model for the right data, with the best example of this being the baseline model. Even though the baseline method was simple, it performed on the same level as the more complex methods in the artificial logs. However, in the more complex real log, it underperformed while the more complex models performed much better.

The models' results in the two first papers showed the potential of data-driven prediction models in the manufacturing context. They demonstrated how the remaining time can be predicted with accuracy, depending on the quality of the process, by using events logs and separating the different products. Some propositions related to future studies from the first paper were explored in the second, such as the test and analysis of real-world data and further investigation of the effects of the rework on the models' performance. However, the propositions

of the second paper are still open to future studies. These include the generation of logs with more complex anomalous behavior instead of just the possibility of rework, and the integration of additional attributes of the event logs to the prediction models with the objective of decreasing their errors.

The second problem explored was in the area of connected vehicles' data, more specifically in the usage of this data to increase drivers' fuel efficiency. Although the topic of fuel efficiency is essentially economical, as the higher the efficiency the smaller the fuel costs, it also is highly ecological. One of the most discussed topics in recent years is global warming and how the burning of fossil fuels accentuates it by releasing greenhouse gasses such as CO2. As such, solutions that aim at decreasing fossil fuel burning hates are of extreme importance. In our case, to achieve an increase in fuel efficiency, the solution proposed was to create fuel efficiency rankings of the drivers, comparing their driving performance. Such rankings have the potential to passively increase overall fuel efficiency, as the generated competitiveness between drivers makes them try to be better than other drivers (Brankovic et al., 2018) and, in doing so, they increase their fuel efficiency.

Ideally, the drivers' driving styles should be what influence the rankings the most. However, given all the external factors that influence fuel efficiency and are out of the drivers' control, simple fuel efficiency rankings would not rank the drivers fairly. To minimize the impact of these extern factors and create fair rankings, the solution proposed by the third paper is to make a clustering of the trips based on context. This context-based clustering allows for the construction of rankings with a separation of the trips driven in different road types (Eppell et al., 2001), which present different patterns of speed, distance, and time of the trips.

The third paper successfully demonstrated how the trips can be separated based on their contexts and validated the effects of the contexts on fuel efficiency. With the clustered trips, it was shown that trips driven in different contexts present significantly different fuel efficiency distributions, proving the unfairness of inter-cluster fuel efficiency comparisons. The primary objective of the paper was achieved with the building of different examples of context-segregated fuel efficiency rankings. The importance of these rankings was demonstrated with examples of drivers and trips that would not be fairly ranked if segregation had not been done. Applications outside fuel efficiency were also shown, with concrete demonstrations of how

context-based clustering can be used to analyze the behavior of a fleet of vehicles. Propositions of possible applications in the area of recommendation systems and product development were made. In future studies, the applicability of integrating the clustering information to a recommendation system to provide more information to it could be investigated. In product development, the use of clustering information to find interest groups is another area to be investigated. One proposed future study in this area is to use the clustering information in the analysis of battery size for electric vehicles (EVs).

The third problem explored comes from the last proposed application of the previous paper. Therefore, it also is in the area of connected vehicles' data. This problem is the optimization of battery sizes for electric vehicles (EVs). The sudden growth in EVs' production and sales is the main motivation for this optimization. As the production of EVs scales, the importance of optimizing the production to decrease costs also scales. In an EV, the battery still is one of the most expensive components, so reducing its cost can significantly reduce overall vehicle costs. Another consequence of the growth of EVs' production is the replacement of a portion of internal combustion engine vehicles (ICEVs). The replacement means that some ICEVs drivers will eventually need to migrate to EVs. Therefore, our optimization problem has to consider the needs and habits of previously ICEVs drivers. The solution proposed by the fourth paper it to use the data from trips of real ICEV and analyze their energy usage to find an optimal battery size. Its relation with context-based clustering comes from the fact that it can be used to separate the trips and drivers into different contexts, which can affect energy usage.

The fourth paper uses data from trips of real ICEVs to analyze how those vehicles are normally used. However, ICEVs do not have raw energy consumption, as what they consume is fuel. Therefore, to obtain energy consumption values a function that converts average speed and temperature into energy efficiency was used. However, as this function was obtained by simulating the energy efficiency of a specific ICEV model, with a specific drag coefficient and weight, it can only be used for data from this vehicle model. After obtaining energy consumption values, to infer the battery sizes needed to complete different trips, energy consumption thresholds were set. Considering that as ICEVs can refuel, EVs can recharge, two different recharging strategies were compared: one with only overnight recharging,

and another with fastcharging in-between trips. Moreover, segregated analysis using context-based clustering was performed to understand how different contexts affect energy consumption.

The analysis done in the fourth paper was able to show the energy consumption profile of ICEVs and how it can change depending on context. The segregated analysis using the context-based clustering coupled with the comparison between two recharging strategies successfully demonstrated how recharging and context affect the battery size. Ultimately, an optimal recharge strategy was chosen and three optimal battery sizes were obtained. Even though the recharging strategy can annoy unaccustomed ICEVs drivers, its much better performed over the no recharging strategy overcome this issue. Using the values obtained from this strategy, considering different types of drivers, three battery sizes were proposed: 40 kWh for urban, 60 kWh for general, and 90 kWh for highway drivers. Although we were able to obtain optimal battery sizes, only two strategies were compared. Future works could further refine these results by comparing other recharging strategies.

This work demonstrated the potential of data-driven methods for engineering problems. The four presented papers explore vastly different areas and problems, having one subject in common: the use of data. Ultimately, what the compilation of these papers shows is how having good enough data so that different methods can be applied to analyze it can help solve difficult problems.

REFERENCES

AALST, W. van der. Data Science in Action. In: PROCESS Mining: Data Science in Action. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. P. 3–23. ISBN 978-3-662-49851-4. DOI: 10.1007/978-3-662-49851-4_1. Citado 4 vezes nas páginas 10, 11, 92.

BRANKOVIC, J.; RINGEL, L.; WERRON, T. How Rankings Produce Competition: The Case of Global University Rankings. **Zeitschrift für Soziologie**, v. 47, n. 4, p. 270–288, 2018. DOI: doi:10.1515/zfsoz-2018-0118. Citado 2 vezes nas páginas 13, 94.

CHOUEIRI, A. C.; SATO, D. M. V.; SCALABRIN, E. E.; SANTOS, E. A. P. An extended model for remaining time prediction in manufacturing systems using process mining. **Journal of Manufacturing Systems**, v. 56, p. 188–201, 2020. ISSN 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2020.06.003. Citado 2 vezes nas páginas 11, 93.

EPPELL, V.; MCCLURG, B.; BUNKER, J. M. A four level road hierarchy for network planning and management. In: ARRB TRANSPORT RESEARCH LTD. PROCEEDINGS of the 20th ARRB Conference. Melbourne, Victoria: ARRB, 2001. P. 1–7. ISBN 0-86910-799-2. Disponível em: https://eprints.qut.edu.au/2349/1/2349.pdf. Citado 2 vezes nas páginas 13, 94.

IEA. **Global EV Data Explorer**. [S.l.: s.n.], 2024. Disponível em: https://www.iea.org/data-and-statistics/data-tools/global-ev-data-explorer. Citado 1 vez na página 14.

KELLEHER, J. D.; TIERNEY, B. 1 WHAT IS DATA SCIENCE? In: DATA Science. [S.l.]: MIT Press, 2018. P. 1–38. Citado 2 vezes nas páginas 10, 92.

KUDYBA, S. chapter 10 - Transforming Unstructured Data into Useful Information. In: BIG Data, Mining, and Analytics. [S.l.]: Auerbach Publications,

2014. ISBN 9780429095290. DOI: https://doi.org/10.1201/b16666. Citado 2 vezes nas páginas 10, 92.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: PROCEEDINGS of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, California: University of California Press, 1967. P. 281–297. Disponível em: https://api.semanticscholar.org/CorpusID:6278891. Citado 1 vez na página 13.

S&P. **Global Battery Market: First movers will likely keep their leads**. [S.l.: s.n.], 2024. Disponível em: https://www.spglobal.com/_assets/documents/ratings/research/101606071.pdf. Citado 1 vez na página 14.

SARKER, I. H. **Machine Learning: Algorithms, Real-World Applications and Research Directions**. [S.l.]: SN Computer Science, 2021. DOI: 10.1007/s42979-021-00592-x. Citado 2 vezes nas páginas 10, 92.

ZECCA, A.; CHIARI, L. Fossil-fuel constraints on global warming. **Energy Policy**, v. 38, n. 1, p. 1–3, 2010. ISSN 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2009.06.068. Citado 1 vez na página 12.