POLITECNICO DI TORINO

Master's degree in
Mathematical Engineering

Master Thesis in collaboration with Tierra S.p.A.

Multivariate time-series clustering for vehicle usage states identification

Supervisor
Prof. Francesco Vaccarino
Co-Supervisors
Prof. Luca Cagliero
Prof. Roberto Notari
Company Advisors
Lucia Salvatori
Riccardo Loti

Candidate
Andrea Megaro
s258985

Academic year 2019-2020

To my beloved Family
and special Friends

Acknowledgements

In questa prima parte di ringraziamenti desidero ricordare i professori, i mentori e le figure professionali che hanno permesso la realizzazione di questo elaborato.
Il primo che intendo menzionare è il Professor Vaccarino, con il quale fin da subito ho collaborato in maniera costruttiva. Il rapporto che si è creato ha permesso di lavorare senza perdite di tempo in formalismi e pensando soprattutto allo scopo del lavoro che abbiamo portato a termine. Le nostre chiamate di allineamento sono state fondamentali dal punto di vista teorico ma mai noiose grazie alle digressioni su interessi comuni.
Ringrazio il Professor Cagliero per i suoi consigli tecnici, soprattutto nella prima fase, che hanno portato allo sblocco in alcune situazioni difficili.
Entrambi fanno parte di un ecosistema più ampio, il gruppo *Smart Data*. Un team di ricercatori e professori che collaborano per la ricerca in ambito Data Science e Big Data. Ringrazio tutti coloro che ne fanno parte e che sono stati presenti, soprattutto nelle ultime fasi di revisione. Un grazie anche al Professor Notari, che ha accettato l'impegno di co-relatore permettendo il completamento del percorso di Double Degree. Per questa opportunità ringrazio l'Alta Scuola Politecnica che mi ha permesso di scoprire nuove realtà attraverso le ASP weeks e altri progetti multidisciplinari di grande interesse.

Continuo i ringraziamenti, passando a tutti coloro che fanno parte della realtà aziendale Tierra. Un grazie speciale a Lucia, che ha avuto la pazienza di ascoltare i miei dubbi e di rispondere ad ognuno di questi. Lucia è stata la guida principale, nonchè tutor aziendale, che mi ha riportato sulla diritta via ogni volta che le mie ricerche iniziavano a prendere una piega troppo teorica. Questo mi ha permesso di crescere dal punto di vista della collaborazione aziendale. La tendenza a divagare è sempre stata in me, ma grazie hai suoi modi gentili, mi ha fatto capire l'importanza di rimanere su un obiettivo e perseverare su questo.
Lei non era sola nel team di ricerca e sviluppo di analisi dati. Infatti a fiancheggiarmi nel percorso di tesi ci sono stati anche Riccardo, secondo tutor aziendale, e Calogero. Ringrazio Riccardo per i prezioni consigli in fase di revisione dei risultati settimanali e per i suggerimenti dei nuovi obiettivi per gli incontri successivi.

Un grazie a Calogero, che mi ha aiutato con i programmi che periodicamente avevano dei problemi oppure ad inserire i file su GitHub anche quando era in ferie, tranquillo sul divano. Per questo, ma anche molto altro, lo considero un amico più che un collega. Infine ringrazio la mia collega di lavoro Silvia per i consigli su Python e lo scambio di opinioni, soprattutto nella fase iniziale di esplorazione dati.

In conclusione di questa prima parte di ringraziamenti, vorrei citare sicuramente tutti i tecnici dell'area IT e del lato gestionale dell'azienda Tierra. Una realtà che sicuramente mi lascia un bel ricordo.
Mi sarebbe piaciuto vivere le persone e la struttura molto di più, ma a causa delle chiusure persistenti che ci sono state durante tutto il 2020, purtroppo non è stato possibile. Non è detto che questo non possa mai più capitare in futuro. Questa tesi infatti rappresenta un grande "arrivederci", non un "addio".

In queste altre righe, cercherò di ricordare tutte le persone che mi sono state vicine durante il mio percorso di vita e accademico, partendo da quando ho memoria, fino ad arrivare agli amici incontrati durante il percorso di laurea magistrale. Questa parte sarà emotivamente coinvolgente, quasi sentimentale. Quindi voi, deboli di cuore, tenete i fazzoletti sotto mano, potrebbero servirvi.

Sono stati anni pieni di gioie, soddisfazioni e grandi gratificazioni, ma sarebbe da ipocriti dimenticare le incertezze i dubbi e le sensazioni di sconforto che sono state mitigate dalle persone a me più care. Quindi, in queste pagine, vorrei ricordare chi è riuscito a farmi ritrovare la calma o la motivazione dopo momenti difficili, oppure chi mi sostiene davanti alle nuove sfide quotidiane.
Soprattutto in quest'ultimo periodo legato alla pandemia globale COVID-19, e caratterizzato da momenti di solitudine prolungata in casa, queste persone si sono fatte avanti e hanno continuato a fare l'impossibile per rimanere in contatto con me, per ciò che sono e per l'affetto che ci lega. Specialmente durante la prima chiusura totale, ho compreso più a fondo il reale valore di coloro che tutti i giorni sono al mio fianco, e ho anche imparato ad apprezzare maggiormente gli attimi di vita quotidiani, che troppo spesso si danno per scontati.
E' proprio vero, ci si accorge dell'importanza di qualcosa solamente quando ne siamo privati ma facciamo di tutto per riaverla.

I primi ringraziamenti vanno ai miei genitori. Loro mi hanno accompagnato lungo tutto il percorso di crescita, sotto tutti i punti di vista. In particolare, durante le scelte accademiche sono sempre stati al mio fianco e se oggi sono arrivato a questo traguardo è sicuramente grazie a loro. Mi hanno sempre supportato, basandosi sui miei interessi e le mie attitudini, e hanno sempre fatto il tifo per me

senza mai saltare una gara di vita. Non avrei potuto desiderare genitori migliori.
Un ringraziamento di cuore va anche alla mia Federica, che dopo ben cinque anni riesce ancora a sopportarmi. Lei è stata un pilastro solido durante molte situazioni difficili, come ad esempio il periodo all'estero, partito come un'allegra vacanza in Svizzera ma subito trasformato in un incubo dove ero costretto a studiare notte e giorno. Grazie a questi momenti ho capito una cosa, ovunque sia o sarò, son sicuro che potrò fare affidamento su di lei.
Ma non da meno sono stati i miei parenti più stretti, la zia, lo zio (nonchè padrino), mia cugina e il cugino Giò (o così gli piace essere chiamato davanti agli altri), che con i gesti di tutti i giorni mi hanno dimostrato il loro affetto e lo dimostrano ogni volta che ci vediamo, ad esempio durante i frequenti pranzi domenicali. Qui, le grandi chef di famiglia, si divertono a rimpinzarmi sempre più, finchè un giorno non scoppierò... d'amore.
Un ringraziamento speciale penso sia doveroso anche a chi purtroppo non c'è più, i miei nonni. La loro importanza è indubbia, soprattutto per avermi cresciuto, almeno in parte, in alcune fasi iniziali della vita. I consigli e i momenti passati insieme hanno stimolato sempre la mia curiosità, che è il motivo principale per cui ogni volta accetto sfide nuove che mi portano a migliorare me stesso.

Gli amici storici di cui parlerò ora, sono gli "amici di sempre". Le persone che fin dal liceo, e qualcuno anche prima, hanno vissuto con me decisioni, gioie, ma anche paure, come quella della Carrà, professoressa di inglese che dal secondo al quarto anno si è divertita a terrorizzarmi con la rimandatura a settembre (sempre scampata all'ultimo). Ma questo è solo uno dei tanti aneddoti che potrei raccontare per molte pagine, come ad esempio le mitiche avvenure mappanesi con il fantastico compagno di avventure nonché incredibile amico, Manu. Dato che vorrei dedicare qualche pagina anche alla tesi vera e propria, passo direttamente ai ringraziamenti. Manu, Lerri, Norbi, Jack, Dido, Maffo e Jallu, sono contento di avervi avuto al mio fianco tutto questo tempo e grazie di cuore per avermi sempre stimolato a fare meglio grazie ad una sana competizione o ad un aiuto sempre presente. Sicuramente è anche merito vostro se sono arrivato a questo traguardo.
Ringrazio anche gli amici incontrati durante il percorso triennale, tra cui Stefano, alias Ste. Grazie per avermi passato i tuoi appunti di analisi due quella mattina, non tanto per il quaderno in sè, ma perchè senza quel primo scambio, forse, non sarebbe cominciato tutto e non avrei avuto l'amico prezioso che sei.
Un grazie va ora a tutto il gruppo di amici di laurea magistrale che ho incontrato al Politecnico. In particolare Silvia, Lucrezia e Julien con i quali ho collaborato in maniera più stetta ed è nato un bel rapporto di amicizia. E' stata molto dura lavorare in gruppo per un lupo solitario come me, ma con qualche sacrificio ce l'abbiamo fatta... e non è poi andata così male.

Grazie Lorenzo, compagno di corso ma anche di vita, per quell'anno passato insieme. E' partito tutto da un interrogatorio durante una lezione di probabilità, ma poi il rapporto si è subito trasformato in un'amicizia duratura.
Un grazie speciale va ora alle persone che ho conosciuto durante l'erasmus in Svizzera. Come anticipato, questo non è stato un periodo facile, soprattutto dal punto di vista psicologico a causa del forte stress. Tobi, Lore, Nicolas ma anche tutti gli altri ragazzi della casa dove sono rimasto durante i sei mesi di scambio sono riusciti ad alleggerire la forte pressione dell'EPFL, permettendomi di trovare qualche momento di tranquillità e svago. Inutile dire che siamo subito diventati grandi amici. La stessa cosa è successa con Claudio, Anita e Francesco, grazie ai quali l'esperienza di studio si è conclusa positivamente. Abbiamo studiato insieme, ci siamo aiutati durante gli homework che non finivano mai e soprattutto ci siamo rincuorati a vicenda quando le cose sembravano impossibili. Se ripenso a quel semestre... non so come avrei fatto senza di voi.

Per ultimi ma non meno importanti vorrei ringraziare i ragazzi di Gymnasio: Lapuz, Giuse, Andre e Dani. Un esperienza iniziata veramente nel peggiore dei modi, dato che ero proprio sicuro di non voler accollarmi il progetto SEI a cui ero stato assegnato. Da quel momento, ho imparato a conoscervi e siete entrati a far parte della mia vita, ogni giorno di più. Mi avete insegnato tanto, e insieme siamo riusciti a diventare un gruppo così unito che quasi stento a credere a quello che stiamo vivendo. Sono orgoglioso di tutti noi, e spero con tutto me stesso che la nostra idea (o idee chi lo sa) sia un giorno realtà. Per ora, quello che posso dire con certezza, è che siete già diventati capisaldi delle mie amicizie e, grazie al vostro aiuto, sono riuscito a migliorare in molti aspetti, personali e professionali.

A questo punto, spero di non aver dimenticato qualcuno, ma dato che la mia memoria non è proprio perfetta, con queste ultime righe ringrazio anche tutti coloro che hanno contribuito alla persona che sono e al percorso che ho intrapreso e con gioia concluso. Un grazie di cuore.

Contents

List of Tables

List of Figures

« In the era where artificial intelligence and algorithms make more decisions in our lives and in organizations, the time has come for people to tap into their intuition as an adjunct to today's technical capabilities. Our inner wisdom can embed empirical data. »

[Abhishek Ratna]

Chapter 1

Introduction and relative work

The following work is an analysis of unsupervised learning techniques concerning multivariate series. They have been applied to the context of the agricultural vehicle, and therefore time-series involved concerned the values of complex machines that need a lot of maintenance for correct functioning. This analysis is designed to find the right balance of "states of the vehicle" to identify the behavior of the latter in a balanced way. The "states of the vehicle" definition will be clarified in the next paragraph concerning the statement of the problem [Subsection 1.1.2]. The conclusion of this analysis, bring very important results that can be exploited in this industry in many ways. Here are two of them.

One way to benefit from a correct identification of the states of a vehicle is to assign a certain degree of usury to each state and derive an overall consumption score based on the time spent on each state. In fact, if a wear score is given to all states, it is possible to obtain an overall degree of wear and assessments regarding the condition of the vehicle. Subsequently, starting from this analysis, it is possible to proceed with preventive maintenance or to anticipate any permanent breakages. This would certainly allow much more precise cost estimates and more prudent business planning.

A second method of using this information is the preventive correction of users' habits.
The statistics on the number of hours spent in a certain state are certainly very useful to understand which situation occurs more frequently and which are the routines. Based on the wear scale associated with the states (mentioned above) it is possible to change the users' habits by suggesting the most appropriate behavior

according to the different situations. In this way, the lifetime of the machines can be extended.

1.1 An in-company thesis

Data are provided by Tierra S.p.A., a society involved in the IoT sector. The IoT takes its value from the internet connection that allows transforming all internet-connected devices into more adaptable and smart objects. The company has a wide range of products that provide various digital services to its customers through the creation of a network whose nodes are Tierra's devices, which are directly connected to customers' vehicles. This ensures a more dynamic and faster exchange of information, bringing added value to customers' machines.

This work is based on a dataset generated by a test vehicle working in a test field, and so I was not in contact with a third party involved. The vehicle is a Valtra-T182 and it is about twelve years old. During the data recording, the vehicle tested both the Tierra device that generated the dataset involved in the analysis and other devices. Therefore, it did not carry out any agricultural activities, the real purpose of the vehicle. This fact made the analysis not extremely varied, as it would have been possible to highlight other types of behavior under different stress conditions. However, the method used is flexible and can be proposed also with different datasets. Moreover, the low precision of the vehicle was one of the most evident problems. For this reason, some methods to attenuate excessive oscillations have been used.

1.1.1 Nature of the data: CAN system, PNG and SPN messages

All data provided by Tierra had an industrial format and in this section, a short description of the procedure used to translated data into an easy-to-use structure is provided. This part is only illustrative since it is not crucial for the thesis aim. So, it does not intend to go into the details of the protocols involved.
All off-road vehicles use the Controller Area Network (CAN), a standard BUS protocol that allows easy access to information from the control system of the machines. In this way, an efficient secure, and integrated network for data transmission is created. It is based on a serial communication protocol, namely the SAE J1939 protocol, but for further details on it, the reader can refer to the application report [HPL02]. To summarize its content, it is a standard for networking and communication between commercial vehicles without using a host computer.

Each CAN message is generated with a high frequency (up to 100Hz), and they are managed by a controller that pre-processes them. In this network, messages are consistent in each node of the vehicle. These signals concern different parts of the machine such as engine speed, coolant temperature, etc.
These signals are clustered in macro-categories: the PNGs. Within every PNG group or cluster, it is possible to associate all individual signals to a unique code: the SPN (Suspect Parameter Number). In conclusion, an SPN is an ID that uniquely identifies a specific signal of the vehicle.
After a bunch of data fills the memory of the Tierra hardware (about 10Mb), the latter sends this data package to the server via a mobile connection (a SIM card is involved). Finally from the raw data contained in the server, the analysis starts.

Figure 1.1. Data flow stylization: from the creation of the data in the vehicle to the servers for data analysis.

1.1.2 Statement of the problem

Agricultural vehicles are heavily exploited during their lifetime and very often it is necessary to check their condition according to the number of hours spent in a certain state. To clarify the meaning of state one can refer to those identified so far by Tierra. In [Table 1.1] three states are reported as an example.

Status	Speed	IO	Engine Speed
Idle	$= 0$	$= 1$	$= 0$
Work / moving	$> \alpha$	$= 1$	$> \beta$
High workload	$> \gamma$	$= 1$	$> \iota$

Table 1.1. Short description of three main status of vehicles. Threshold values are not specified and parameters considered are more than the ones showed.

Referring to the reported case, the "Idle" status identifies a vehicle that is on but not moving. Some threshold values are set to identify this behavior, considering a limited number of parameters, based on what common sense suggests.
To generalize the concept, one can think of the vehicle as a physical system that

passes in different states based on two factors: the manual inputs of the user and the external inputs of the surrounding environment. According to these conditions, the vehicle will be in a certain working mode which characterizes differently the working state.
To identify the influences that internal and external agents cause to the vehicle, the levels of some machine components are examined.

As shown before in [Table 1.1], until now the company has been identified the states of vehicles with a threshold system. One of the tasks of this thesis, on the other hand, is to find the optimal method of separating (clustering) vehicle behavior in a more precise and automatic way.
A crucial aspect that the company stressed is the granularity of the dataset and the pre-processing phase, the second challenge of this thesis. In fact, the data that have been provided have a very fine granularity (order of milliseconds) and therefore considerable problems have arisen compared to the previous situation. In the threshold system, the series considered had a granularity of a few minutes, as processes for aggregation through simple steps were automatically integrated. For example the aggregation by average.
The most evident problem in the finer data case is evidence of noise. Since the data are not filtered, they present many anomalous behaviors, sometimes caused by the vehicle itself. Secondly, the finer data to be transmitted from the vehicle to the server were in very large quantities and not so useful due to the raw nature of the vehicles involved. Therefore an optimal method of aggregation is certainly needed to avoid slowdowns in any real-time processes. For these and other reasons, special attention has been paid to the alignment and pre-processing of the dataset.

1.2 Summary and main references

The thesis is divided into six chapters which can be summarized as follows. The first is an introduction to the work. The second and third are related to the dataset preparation, and more specifically the second is the theoretical background on which the third (real case application) is based. I will refer to this part as "part one" of the thesis.
The fourth and the fifth chapter have been written with the same logic: the fourth contains the theoretical background on which the fifth chapter is based.
They are the core of the thesis, and they concern unsupervised methods for multivariate time series clustering. I will refer to this part as "part two" of the thesis.
The sixth chapter contains my conclusions, observations, and possible future developments.

The first part of the thesis, which concerns data preparation, includes operations to transform the rough data into the multivariate series needed for subsequent analysis. This was made possible by several steps. First, alignment and synchronization techniques were used to make the time series homogeneous with respect to their clock. In [Section 2.1], time-frequency theory involved is reported.
Subsequently, more datasets have been produced, each built according to a specific granularity and method of aggregation/downsampling.
Starting from the various datasets produced, the work cycles were highlighted using a technique based on the trend of particular series, since this information was not directly available.
Then, since a fixed window length method has been used, the optimal value of the window length was needed. This was made possible by a motif recognition based algorithm called VALMOD [LZPK18b]. After the window identification, data was ready for the second part of the analysis. This part of the thesis concerns the multivariate clustering techniques, the core of the research.
To assess the best cluster configuration, since all data are completely unlabeled, it was necessary to rely on the internal CVIs (Cluster Validity Indices), which consider the intra-cluster and inter-cluster distance as goodness indices. Most of the ones used in this part are taken from the article [AGM+13].
Since not all CVIs always agree on the same ranking of best clustering methods, a ranking algorithm was necessary. The *RankAggreg* [PDD09] has been exploited, which identifies an optimal ranking, based only on goodness indices, by using them as weights.
All clustering methods exploited in the further analysis have been divided into single-signal based or dataset oriented techniques.
The first takes as input the single signals and then generates different clusters according to the merging of the results of the single signals. In [Section 5.1] this approach is exploited. This part is based on the study of unsupervised univariate clustering techniques such as the k-shape algorithm [PG15].
Afterward, a dataset-based approach was explored. This time, multivariate series are exploited simultaneously. In this part, three approaches have been compared to achieve optimal clustering results. A novel multivariate time-series clustering method is proposed in [Subsection 5.2.2].
In the final chapter, conclusions are reported. An optimal model is proposed and future developments are suggested.

Chapter 2

Time-series preprocessing theory

This chapter is proposed as the theoretical background for the next chapter, where preliminary analyses have been conducted. It is divided into three parts. The first is an introduction to the time-frequency analysis, where series are considered as signals to be transformed and studied through the Fourier (or other) transform. Then, time-series are seen with a probabilistic lens by stressing the stochastic processes under them. In the last section, the VALMOD algorithm is explained to the reader. This matrix profile-based algorithm will be exploited to find an optimal length of the time window in the segmentation phase; crucial for the next part of the thesis.

2.1 Time-frequency analysis

The theory explained in this part has been applied to the dataset as the first step of transformation to align time-series at a unique clock. In fact, techniques exploited in the next sections need aligned time-series to be properly evaluated. To obtain a homogeneous series, it was necessary to use re-sampling techniques, and afterward, to obtain the desired granularity, the functions explained in the following part have been exploited. Most of the following contents are taken from the book [VKG14].

2.1.1 Introduction

In the beginning, the mathematical concepts that will be used for the filtering and transform section are mentioned. Not being the core of the thesis, this introduction will report only those concepts strictly essential to understand what follows.

Let's begin this part by defining the main normed and inner product spaces involved in this chapter.

- $\mathbb{C}^N$ **spaces.** The normed vector space of complex-valued finite-dimansional vectors is generally provided with the $p-$norm defined as

$$||x|| = \left(\sum_{n=0}^{N-1} |x_n|^p \right)^{1/p}.$$

If $p = 1$, it is defined *Manhattan norm.* In case $p = 2$, one get the usual Euclidean square norm, which is induced by an inner product. In this case the inner product and the norm are defined respectively,

$$\langle x, y \rangle = \sum_{n=0}^{N-1} x_n \overline{y_n}, \qquad ||x|| = \left(\sum_{n=0}^{N-1} |x_n|^2 \right)^{1/2}.$$

If $p = \infty$ the norm is defined as

$$||x||_\infty = \max\left(|x_0|, \ldots, |x_{N-1}|\right).$$

For $p \in (0,1)$ it is not a norm, but it is not important for the purpose of this chapter to deep on this particular case.

- $\ell^p(\mathbb{Z})$ **spaces.** One can define the norm on $\mathbb{C}^\mathbb{Z}$ as done in the previous case,

$$||x||_p = \left(\sum_{n \in \mathbb{Z}} |x_n|^p \right)^{1/p}.$$

To satisfy the norm properties, also in this case values of $p \in [1, \infty)$ are considered, while for $p = \infty$ one get the extension,

$$||x||_\infty = \sup_{n \in \mathbb{Z}} |x_n|.$$

Definition 2.1 *For any $p \in [1, \infty]$, the normed vector space $\ell^p(\mathbb{Z})$ is the subspace of $\mathbb{C}^\mathbb{Z}$ consisting of vectors with finite ℓ^p norm.*

In the specific case of $p = 2$, the ℓ^p norm is induced by an inner product, defined as follows,

$$\langle x, y \rangle = \sum_{n \in \mathbb{Z}} x_n \overline{y_n}, \qquad ||x|| = \left(\sum_{n \in \mathbb{Z}} |x_n|^2 \right)^{1/2}.$$

- $\mathcal{L}^p(\mathbb{R})$ **spaces.** In this space, the norm is defined on $\mathbb{C}^{\mathbb{R}}$,

$$||x||_p = \left(\int_{-\infty}^{+\infty} |x(t)|^p \, dt\right)^{1/p}.$$

Also in this case, the above definition is valid for $p \in [1, \infty)$ while for $p = \infty$ the norm is extended as follows,

$$||x||_\infty = \operatorname{ess\,sup}_{t\in\mathbb{R}} |x(t)|.$$

Definition 2.2 *For any $p \in [1, \infty]$, the normed vector space $\mathcal{L}^p(\mathbb{R})$ is the subspace of $\mathbb{C}^{\mathbb{R}}$ consisting of vectors with finite $\mathcal{L}^p$ norm.*

In the specific case of $p = 2$, the $\mathcal{L}^p$ norm is induced by an inner product, defined as follows,

$$\langle x, y\rangle = \int_{-\infty}^{+\infty} x(t)\overline{y(t)}\, dt, \qquad ||x|| = \left(\int_{-\infty}^{+\infty} |x(t)|^2 \, dt\right)^{1/2}.$$

The above mentioned spaces are all *complete*, and so they met the following definition,

Definition 2.3 A normed vector space V is *complete* when every Cauchy sequence in V converges to a vector in V.

This definition is recalled in the following one, which defines the most important spaces for this section.

Definition 2.4 A complete normed vector space is called a *Banach space*. A complete inner product space is called a *Hilbert space*.

So, in the definition of a *Hilbert space*, a scalar product is set, and the norm of this space derives from this. Besides, the triangular inequality is met, which says that the module of the scalar product of two vectors is less than or equal to the product of their norms. Equality occurs when two vectors are linearly dependent (or at least one of the two is null).
It is crucial to underline that thanks to this, the scalar product defines the similarity between vectors.
Now, let's see the definition of an orthonormal basis in this context.

Definition 2.5 A set of vectors $\{\varphi_k\}_{k\in J} \subset V$ (where J is finite or countably infinite), is called *basis* for a normed vector space V when

- it is complete in V, meaning that, for any $x \in V$, there is a sequence $\alpha \in \mathbb{C}^J$ such that
$$x = \sum_{j \in J} \alpha_j \varphi_j.$$
- For any $x \in V$, the sequence above mentioned α, is unique

Definition 2.6 If H is an Hilbert space, a set of vectors $\Phi = \{\varphi_k\}_{k \in J} \subset H$ (where J is finite or countably infinite), is an *orthonormal basis* if

- Φ is a basis for H, and
- Φ is orthonormal, that is
$$\langle \varphi_k, \varphi_j \rangle = \begin{cases} 1, & \text{if } k = j, \\ 0, & \text{if } k \neq j. \end{cases}$$

Now that the main basic concepts have been shown, the following crucial theorem, formalize the digitization concept.

Theorem 2.1 Riesz theorem. If $\{\varphi_k\}_{k \in J}$ (where J is finite or countably infinite) is an orthonormal base of H, the application
$$\begin{aligned} \Phi^* : H &\to \ell^2(J) \\ x &\mapsto \{\langle x, \varphi_k \rangle\}_{k \in J} \end{aligned}$$
is an isometric isomorphism. The inverse is
$$\begin{aligned} \Phi : \ell^2(J) &\to H \\ \alpha = \{\alpha_k\}_{k \in J} &\mapsto \sum_{k \in J} \alpha_k \varphi_k. \end{aligned}$$

So, $x = \sum_{k \in J} \langle x, \varphi_k \rangle \varphi_k$ converges unconditionally on H (no matter the order of the indices).

The fact that the series converge unconditionally is implicitly written in the summation, since it is not specified the order of the terms.
Another important clarification concerns the therms *isometric isomorphism*.
Isometric means that the norm is preserved by applying the map:
$$||x||_H^2 = \sum_{k \in J} |\langle x, \varphi_k \rangle|^2.$$

Thanks to the term *isomorphism*, it is possible to say that the application is bijective.

More in general, if the $\{\varphi_k\}$ is an orthonormal system of H, one can define Φ as the synthesis operator and Φ^* as the analysis operator.
In the theorem, the adjoint map coincides with the inverse and so, one can state that Φ is a unitary operator.

In the last part of this subsection, the most important case for further developments is proposed. In finite dimension, if $H = \mathbb{C}^N$ and $\Phi : \mathbb{C}^N \to \mathbb{C}^N$, one can define the Fourier basis $(f^{(0)}, \dots, f^{(N-1)})$ as

$$f_n^{(k)} = \frac{1}{\sqrt{N}} e^{\frac{2\pi}{N} jkn} \qquad \text{where } k, n = \{0, \dots, N-1\}.$$

This basis will be used to perform the Fourier transform in the case of finite series (but it can also be extended with a similar form into the countably infinite case). In this way, signals can be decomposed efficiently.

2.1.2 Filtering

The starting setting for this subsection is a discrete signal of infinite length $\{x_k\}_{k\in\mathbb{Z}}$. One can also consider a signal of finite lengths, in fact the concept is easily extensible in infinite dimension in the following two ways:

- extension with zeros

$$(x_{-1}, x_0, x_1) \mapsto (\dots, x_{-1}, x_0, x_1, \dots);$$

- periodic extension

$$(x_{-1}, x_0, x_1) \mapsto (\dots, x_{-1}, x_0, x_1, x_{-1}, x_0, x_1, x_{-1}, x_0, x_1, \dots).$$

A discrete-time system is an operator A, that maps an input sequence $x \in V$ into an output sequence $y \in V$,

$$y = A(x).$$

In the following section, space V considered is always $\ell^2(\mathbb{Z})$, so if it is written only ℓ^2, it refers to the aforementioned space.

Definition 2.7 A discrete-time system A is called linear when for any given input x and y, and any α and $\beta \in \mathbb{C}$,

$$A(\alpha x + \beta y) = \alpha A(x) + \beta A(y).$$

Once established the bases of the domain and the codomain (in this chapter always the standard one), the linear operators can be represented in a unique way by a matrix.
By defining the Kronecker delta sequence $\delta \in \ell^2(\mathbb{Z})$ as follows,

$$\delta_k = \begin{cases} 1, & \text{if } k = 0, \\ 0, & \text{if } k \neq 0, \end{cases} \tag{2.1}$$

each column k of the unique matrix is the resulting output from taking the shifted Kronecker delta sequence as the input of the system, δ_{n-k}. In fact, the Kronecker delta sequence and its shifts represent the standard basis of $\ell^2(\mathbb{Z})$.

Definition 2.8 A discrete-time system A is called memoryless when for any given integer k and inputs x and x',

$$\mathbb{1}_{\{k\}}x = \mathbb{1}_{\{k\}}x' \qquad \Longrightarrow \qquad \mathbb{1}_{\{k\}}A(x) = \mathbb{1}_{\{k\}}A(x').$$

Definition 2.9 A discrete-time system A is called shift-invariant when for any given integer k and inputs x,

$$y = A(x) \qquad y' = A(x'), \qquad \text{where } x'_n = x_{n-k} \text{ and } y'_n = y_{n-k}.$$

With reference to the previous definition of the unique matrix defined for a linear system, a clarification in the case of LSI (Linear Shift Invariant) is necessary. In fact, by definition, it is easy to understand why its columns are identical but shifted.

Now, a formal definition of the impulse response of an LSI is proposed and thanks to that, the convolution concept will be presented.

Definition 2.10 A sequence h is called the impulse response of an LSI discrete-time system H when the Kronecker delta input produces output h.

As said before, in the case of the LSI systems, the columns of the unique matrix are always the same but shifted, and so, the sequence resulting from the Kronecker delta sequence as input completely specifies the system.
Now let's introduce the key concept of convolution.
To obtain the matrix representation of an LSI system, impulse response and its shift are considered to form its columns. If one considers this operation as a sum, it is easier to understand the convolution concept. In fact, given an arbitrary input x, it can be expressed as

$$x = \{x_k\}_{k\in\mathbb{Z}} = \sum_{k\in\mathbb{Z}} x_k T_k \delta, \tag{2.2}$$

where T_k is the translation operator, which means that if it is applied to a generic $x \in \ell^2(\mathbb{Z})$ it produce a translation of k positions,

$$(T_k x)_n = x_{n-k}.$$

In this case T_k is applied to the Kronecker delta sequence, so that it produces the standard basis of $\ell^2(\mathbb{Z})$.
Moreover, T_k commute with the matrix H corresponding to a LSI system.

Let's introduce the context by defining $H : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ as a limited LSI operator, x as a generic input signal, and y as its output response.

$$\begin{aligned} y &= Hx \\ &= H \sum_{k\in\mathbb{Z}} x_k T_k \delta \qquad \text{(by the expression 2.2)} \\ &= \sum_{k\in\mathbb{Z}} x_k H T_k \delta \\ &= \sum_{k\in\mathbb{Z}} x_k T_k H \delta \\ &= \sum_{k\in\mathbb{Z}} x_k T_k h \qquad \text{(where } T_k h \in \ell^2 \text{ and can be read as synthesis operator)} \end{aligned}$$

At this point one can notice that $(T_k h)_n = h_{n-k}$ and so

$$(x * h)_n = \sum_{k\in\mathbb{Z}} x_k h_{n-k},$$

and this can be formalized in the following definition.

Definition 2.11 The convolution between sequences h and x is defined as

$$(Hx)_n = (x * h)_n = \sum_{k\in\mathbb{Z}} x_k h_{n-k} = \sum_{k\in\mathbb{Z}} h_k x_{n-k},$$

where H is called the convolution operator associated with h.

Now it is possible to link the previous definition with the *filtering* one.

Definition 2.12 A *filter* is the impulse response of a system while the convolution with the impulse response is called *filtering*.

There are different classes of filters, the most famous are listed.

- Causal filters: $h_n = 0 \,\forall\, n < 0$.
- Anticausal filters: $h_n = 0 \,\forall\, n > 0$.
- Finite impulse response (FIR) filters have only a finite number of coefficients $h_n \neq 0$.
- Infinite impulse response (IIR) filters have infinitely many nonzero terms.

2.1.3 Discrete-time Fourier transform

In this subsection, different ways to analyze sequences and discrete-time systems are described. First, the discrete-time Fourier transform (DTFT) definition is presented. It represent the Fourier transform for infinite-length discrete-time signals, and it is a 2π -periodic function of frequency $\omega \in \mathbb{R}$ that is written as $X(e^{j\omega})$. Its variation to the case of finite length sequences is explained below.
This transforms aim to bring to light hidden aspects of the signal or to reduce the computational cost of certain operations. Another purpose is to study the effects of a certain transformation to avoid problems of transmission of the information. This last one will be subsequently deepened.
Let's consider a complex exponential sequence

$$v_n = e^{j\omega n}, \quad n \in \mathbb{Z},$$

where ω is any real number. The quantity ω is called the angular frequency. Let's consider a convolution operator H and assume that its impulse response is in $\ell^1(\mathbb{Z})$. Under these hypothesis the convolution $h * v$ assumes a particular form.

$$\begin{aligned}
(Hv)_n &= (h * v)_n \\
&= \sum_{k\in\mathbb{Z}} v_{n-k} h_k \\
&= \sum_{k\in\mathbb{Z}} e^{j\omega(n-k)} h_k \\
&= \sum_{k\in\mathbb{Z}} h_k e^{-j\omega k} e^{j\omega n} \\
&= \lambda_\omega v_n \qquad \text{where } \lambda_\omega \coloneqq \sum_{k\in\mathbb{Z}} h_k e^{-j\omega k}
\end{aligned}$$

This proves that applying H to the complex exponential, a scalar is returned. It seems that v is an eigensequence of H with the corresponding eigenvalue λ_ω.

Definition 2.13 DTFT. The discrete-time Fourier transform of a sequence x is

$$X(e^{j\omega}) = \sum_{n\in\mathbb{Z}} x_n e^{-j\omega n}, \qquad \omega \in \mathbb{R}.$$

It exists if the above summation converges for all $\omega \in \mathbb{R}$ and it is called spectrum of x. The inverse DTFT of a 2π-periodic function $X(e^{j\omega})$ is

$$x_n = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X(e^{j\omega}) e^{j\omega n} d\omega, \qquad n \in \mathbb{Z}.$$

When the DTFT exists, the DTFT pair can be represented as

$$x_n \overset{DTFT}{\longleftrightarrow} X(e^{-j\omega}).$$

These concepts are then expandable to the case of finished or periodic signals in the following way. Let's consider a finite length signal $(x_0, \dots, x_{N-1})$ and extend to zero by obtaining the following $(\dots, 0, x_0, \dots, x_{N-1}, 0, \dots)$. In this way, it is possible to apply the DTFT by obtaining

$$\sum_{n=0}^{N-1} x_n e^{-j\omega n} = X(e^{j\omega}).$$

However, in this case, there are N degrees of freedom and so, it can be considered a function w.r.t. ω. In general, if one sets the degree N, the dimension of the space is also set.
Under this logic, if N equally spaced point in the interval $(0,2\pi)$ are set, the DTFT is evaluated only in for those points as

$$X_k \coloneqq X(e^{j\omega_k}) = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn}, \qquad \text{where } k = 0, \dots, N-1. \tag{2.3}$$

It also possible to indicate the complex number $e^{-j\frac{2\pi}{N}}$ as W_N. It will be used in the following definition.

Definition 2.14 DFT. The discrete Fourier transform of a length-N sequence x is

$$X_k = (Fx)_k = \sum_{n=0}^{N-1} x_n W_N^{kn}, \qquad k \in \{0,1, \dots, N-1\}.$$

It is called spectrum of x. The inverse DFT of a length-N sequence x is

$$x_n = \frac{1}{N}(F^* X)_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k W_N^{-kn}, \qquad n \in \{0,1, \dots, N-1\}.$$

The DFT pair can be represented as

$$x_n \overset{DFT}{\longleftrightarrow} X_k.$$

Within the definition, we have introduced $F : \mathbb{C}^N \to \mathbb{C}^N$ to represent the linear DFT operator. In the same way, one can define the discrete cosine transform (DCT), which changes form depending on the scale factor and real basis considered. The type II DCT, involved in the following chapter, is defined as follows

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi k}{N}\left(n + \frac{1}{2}\right)\right], \qquad k \in \{0,1, \dots, N-1\}.$$

The inverse transform in this case is

$$x_k = \frac{2}{N} \sum_{n=0}^{N-1} X_n \cos\left[\frac{\pi k}{N}\left(n + \frac{1}{2}\right)\right], \qquad k \in \{0,1,\dots,N-1\}.$$

In this section, the computational aspect of the transformation methods are not analyzed, however, all the implemented algorithms are based on the articles [Mak80, CT65].

2.1.4 Multirate systems

Multirate systems are combinations of filters and another category of operators in time-frequency theory that will be defined ahead: downsampling or upsampling. Let's start with the definition of the downsampling function of degree k in the case of a signal of infinite length, which will be indicated with the D letter.

$$\begin{aligned} D_n \colon l^2(\mathbb{Z}) &\to l^2(\mathbb{Z}) \\ x_n &\mapsto x_{kn} \end{aligned}$$

The matrix that represents this transformation is an identity matrix whose lines are translated by k units and seen from another point of view the operator maps

$$(\dots, x_{-1}, x_0, x_1, \dots) \xmapsto{D_k} (\dots, x_{-k}, x_0, x_k, \dots).$$

The upsampling function can be defined with the same logic, but since it has not been applied in the following chapter, it will not be further investigated. The operator described is not a filter because it is not shift-invariant and it may present some critical issues. The most important is the elimination of frequencies due to the sampling activity.
An important mathematical result in this context is the Nyquist-Shannon sampling theorem. In the case of finite band signals, this states that in order to sample a signal without loss of information, it must be sampled at a frequency at least twice the highest frequency among all the informative spectral components (also called Nyquist frequency).
If this theorem is not respected, one can incur the aliasing issue, that is the cancellation or the distortion of some frequencies important for the signal reconstruction. More formally the theorem for digital signals can be expressed as follows. Let $BL(-\frac{\pi}{N}, \frac{\pi}{N})$ a limited band sequence defined as follows.

$$BL\left(-\frac{\pi}{N}, \frac{\pi}{N}\right) \coloneqq \left\{x \in l^2(\mathbb{Z}) : X(e^{jw}) = 0,\ \frac{\pi}{N} \leq |w| \leq \pi\right\}$$

Let $x \in l^2(\mathbb{Z})$, its orthogonal projection in $BL(-\frac{\pi}{N}, \frac{\pi}{N})$ is

$$P_x = \sum_{k \in \mathbb{Z}} y_k T_{KN} g, \text{ where } y_k = < x, T_{KN} g >_{l^2} .$$

If $x \in BL(-\frac{\pi}{N}, \frac{\pi}{N})$, then

$$x_n = \sum_{k \in \mathbb{Z}} x_{KN} sinc\left(\frac{\pi}{N}(n - KN)\right).$$

In most real cases, this theorem is not respected. To overcome the problem of aliasing, it is necessary to define a multi-rate system.
In the downsampling case, the optimal system is a composition of an anti-aliasing filter before the downsampling operation.
Let x be the incoming signal, then the response signal y can be obtained in the following way to avoid/reduce the aliasing phenomenon.

$$u = G \cdot x = x * g$$
$$y = D_k \cdot u = D_k \cdot G \cdot x$$

In the system, the G represents a low-pass filter, and then, the k-order downsampling operator D_k is applied. A low-pass filter allows keeping only the frequencies that will be retained after downsampling and so, in this way, aliasing is avoided.

In the applications proposed in the next chapter, two aforementioned systems have been applied. The first is a method that is based on the Fourier transform and uses a pre-filtering with an order 8 Chebyshev type I filter. The second one is based on the cosine transform and exploit an ideal low pass filter. The ideal low-pass filter truncates the frequencies that cause the aliasing and the second filter is one of the most used to avoid the aliasing phenomenon. The construction of the two filters is not reported as it is not in the objectives of the thesis but in [VKG14], theoretical clarifications are reported.

2.2 Basic elements for time series study

Many of the theoretical contributions in this section have been taken from the following book and articles on time series [MC09, CC08].
To fully understand the concept of time series it is necessary to start from the definitions of stochastic processes in the theory of probability.

Definition 2.15 A stochastic process is a collection of random variables

$$\{X_t\}_{t\in T} \tag{2.4}$$

defined on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$.

The t index is usually called time. Each random variable X_t assumes values, called states, in a set E called space of the states of the process. In particular, since each random variable is defined as a function by a sample space S to the real set, a stochastic process is a set of functions

$$\{X_{t,s}\}_{t\in T, s\in S}.$$

If one fixes $t = t_0$ the function $X_{t_0,s}$, is a random variable on the sample space S. Likewise, if one fixes a particular point $s_0 \in S$, the function X_{t,s_0} is a realization of the stochastic process.
According to the type of domain of s and t different families of random variables are defined.

- The space of the states E can be discrete or continuous. In the first case (discrete space) the stochastic process is also called *chain* and space $E \in \{0,1,2,\dots\}$. In the second case, the set of values assumed by the random variable is uncountable.

- The time index can be discrete or continuous. A stochastic process at a discrete-time is also called stochastic sequence and it is denoted as $\{X_n\}_{t\in T}$, where the T set is countable. In this case, the state changes only at certain times. On the contrary, if the state changes occur at any time (in a finite or infinite set of real intervals), then you have a continuous-time process, denoted as $\{X_t\}_{t\in T}$.

For the rest of the thesis, all processes occur at discrete moments but the range of states has great variability. A realization of the stochastic process X_t will be called x_t but we will refer with the same name "time-series" both to the stochastic variable and to the realization.

2.2.1 Autocorrelation and partial autocorrelation

Key concepts for the following discussion are the mean, variance, and autocovariance functions of the stochastic process X_t. These are defined in the following way.

Definition 2.16 Let $\{X_t\}$ be a time series with $\mathbb{E}(X_t^2) < \infty$. The mean function of $\{X_t\}$ is

$$\mu_X(t) \coloneqq \mathbb{E}(X_t) \quad \forall\, t \in T. \tag{2.5}$$

The covariance function of $\{X_t\}$ is

$$\gamma_X(r,s) \coloneqq \mathrm{Cov}(X_r, X_s) = \mathbb{E}\left[(X_r - \mu_X(r))\,(X_s - \mu_X(s))\right] \quad \forall\, r,s \in T. \tag{2.6}$$

Definition 2.17 Let $\{X_t\}$ be a time series and let $F_X(x_{t_1+h}, \ldots, x_{t_n+h})$ represent the cumulative distribution function of the unconditional [1] joint distribution of $\{X_t\}$ at times $t_1 + h, \ldots, t_n + h$. Then, $\{X_t\}$ is said to be strictly stationary if

$$F_X(x_{t_1+h}, \ldots, x_{t_n+h}) = F_X(x_{t_1}, \ldots, x_{t_n}) \quad \forall\, h, t_1, \ldots, t_n \in T, \forall\, n \in \mathbb{N} \tag{2.7}$$

A weak concept of stationarity is now introduced.

Definition 2.18 A time series $\{X_t\}$ is (weakly) stationary if

- $\mu_X(t)$ is independent of t;
- $\gamma_X(t+h, t)$ is independent of t for each h.

Whenever the term stationarity is used, it shall mean weakly stationary as in [Definition 2.18], unless otherwise specified. To understand how much information occurs cyclically over time the following function is considered.

Definition 2.19 Let $\{X_t\}$ be a stationary time series. The *autocovariance function* (ACVF) of $\{X_t\}$ at lag h is

$$\gamma_X(h) \coloneqq \gamma_X(h,0) = \gamma_X(t+h, t) = \mathrm{Cov}(X_{t+h}, X_t) \quad \forall\, t \in T. \tag{2.8}$$

The autocorrelation function (ACF) of $\{X_t\}$ at lag h is

$$\rho_X(h) \coloneqq \frac{\gamma_X(h)}{\gamma_X(0)} = \mathrm{Cor}(X_{t+h}, X_t) \quad \forall\, t \in T. \tag{2.9}$$

[1] This means that the distribution has no reference to any particular starting value

These definitions, in particular the last one, allows obtaining the recurrent patterns within a given time series. However in practical cases, one will always have an observed historical series of finite lengths. To adapt the previous function to these cases, it is necessary to define the sample autocorrelation function. This is a good approximation of the autocorrelation function, in case the series is stationary. A comment is required at this point. The autocorrelation function (sample or not) is also applicable to non-stationary series. For series that are not stationary, the shape of the autocorrelation function can be analyzed and it can be used as a discrimination factor. For example in the case of the series with a trend, the downward movement of the autocorrelation function will be slower than other cases.
Now let's define the sampled version of the functions previously seen.

Definition 2.20 Let $x_1, \ldots, x_n$ be observations of a time series. The sample mean of $x_1, \ldots, x_n$ is

$$\bar{x} \coloneqq \frac{1}{n} \sum_{t=1}^{n} x_t. \tag{2.10}$$

The sample autocovariance function is

$$\hat{\gamma}(h) \coloneqq \frac{1}{n} \sum_{t=1}^{n-|h|} (x_{t+|h|} - \bar{x})(x_t - \bar{x}), \quad -n < h < n. \tag{2.11}$$

The sample autocorrelation function is

$$\hat{\rho}(h) \coloneqq \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}, \quad -n < h < n. \tag{2.12}$$

Finally the definition of partial autocorrelation is mentioned. The partial autocorrelation function measures the correlation between the series (x_t) and its lagged version (x_{t-h}) without taking into account the correlation with eh middle values $(x_{t+1}, \ldots, x_{t-(h-1)})$.

2.2.2 Correlation

Another fundamental concept is introduced. The correlation is a dimensionless measurement, which expresses how linearly two variables co-variate. In fact, it is a coefficient that belongs to the interval $[-1, 1]$ and it gives the following information.

- a correlation of 1 indicates exact positive linear association;
- a correlation of 0 indicates no linear association;
- a correlation of -1 indicates exact negative linear association.

To define this function, the previous process is not repeated but the formulas are directly reported as an extension of the autocorrelation case. In fact, if before the comparison was made on the same variable at different lags, now the comparison is made between two different variables.
Let's start with the definition of covariance between two random variables X_t and Y_t . It will be indicated with δ only to be not confused with the autocorrelation function.

$$\delta(X_t, Y_t) \coloneqq \frac{\mathbb{E}[(X_t - \mu_X(t))(Y_t - \mu_Y(t))]}{\sigma(X_t)\sigma(Y_t)}. \tag{2.13}$$

Starting from this definition and extending the concept to the case where X_t and Y_t are two random variables referring to two time-series, the formula of the sample correlation becomes the following.

Definition 2.21 Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ be observations of two time series. The sample correlation is

$$\hat{\delta}(x, y) \coloneqq \frac{\sum_t (x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_t (x_t - \bar{x})^2 \sum_t (y_t - \bar{y})^2}}. \tag{2.14}$$

2.2.3 The classical decomposition model

This last definition will be used in the next chapter to understand what signal is really indispensable. In this way, it will be possible to have a dataset without redundancies.

The autocorrelation, instead, will be exploited to understand which signals are characterized by seasonality. This term refers to those systematic components in the time inside a time series. Formally, in fact, the seasonality of order k is the correlation between the i-th and the $(i - k)$-th element. In fact, looking at the correlogram, that is the graph which on the x-axis has the lags and represents the autocorrelations for each of them, it is possible to identify recurrent patterns for signals marked by seasonal patterns.

Another element that can characterize a time series is the trend, which is long-term behavior that assumes. In general, the trend is the component of increasing or decreasing of the time series. A very effective method to detect long-term behavior is the "moving average" or the "moving median". The methods are simple but effective; in a nutshell, they evaluate a mean/median value within a scrolling window that goes along the signal.

Now, let's see how to decompose a time-series i.e. with respect to their trend

and seasonality components. This model is also known as "the classical decomposition model".

$$X_t = T_t + S_t + I_t. \tag{2.15}$$

This model decomposes the process through three variables. It reports the trend component T_t that is a very smooth function that represents the long-term progression, a function with a known period referred to as the seasonal component (S_t), while I_t is the variable that represents the noise, the irregular component.
The second model that is reported is the multiplicative model, where the components are the same but the relationship changes.

$$X_t = T_t \cdot S_t \cdot I_t. \tag{2.16}$$

2.3 Motif identification

In this section, an algorithm to find recurrent time-series patterns based on the matrix profile will be described; its name is VALMOD. As mentioned in the introductory chapter, a not-overlapped fixed window method will be proposed in the following chapter, and therefore it is necessary to understand what is the optimal window length by studying the probability of finding all different window lengths. In fact, the dataset is completely unlabeled and domain experts can only specify a range of lengths that could capture the variations and behaviors in time series windows.
Originally, the algorithm is used for different purposes, in fact, VALMOD aims to find all motifs in a given range of lengths within a data series very effectively. It is important to underline the last adjective because the datasets that will be analyzed during the applications of this chapter are very large, so, a scalable method was necessary. The chapter is based on [LZPK18a, LZPK18b]. However, in the section are not reported the details of the above-mentioned articles but an understanding and the basic ideas are reported.
From now on, the notation will change compared to before but the following definitions will clarify the context.

Definition 2.22 Data series. A data series $T \in \mathbb{R}$ is a sequence of n real-valued numbers $t_i \in \mathbb{R}$.

But in general, the section will investigate finding series subsequences.

Definition 2.23 Subsequence. A subsequence $T_{i,l} \in \mathbb{R}^l$ of a data series T, is a continuous subset of the values from T of length l starting from position i.

Given a particular subsequence, the goal will be to find his pair with minimum distance.

Definition 2.24 Data series motif pair. $T_{a,l}$ and $T_{b,l}$ is a motif pair iff $dist(T_{a,l}, T_{b,l}) \leq dist(T_{i,l}, T_{j,l}) \forall i, j \in [1, 2, ..., n - l + 1]$ where $a \neq b$ and $i \neq j$; $dist$ evaluate the z-normalized euclidean distance between the two input subsequences.

It is important to note that if one deletes the motif pair, then the second one will have the best result and thus become the new motif pair. This way, as result, one obtains an ordered list of pairs of length l. Let's always assume that n is the length of the starting series. One method to optimize the detection of motif pairs is to organize the distances of each subsequence with all the others in a matrix called matrix profile. From this derives the following two definitions.

Definition 2.25 Distance profile. Given a data series T, the distance profile $D \in \mathbb{R}^{(n-l+1)}$ with respect to a subsequence $T_{i,l}$ is a vector that stores $dist(T_{i,l}, T_{j,l}) \ \forall \ j \in [1, 2, ..., n - l + 1]$ where $i \neq j$.

Definition 2.26 Matrix profile. A matrix profile (MP) is a series of z-normalized Euclidean distance between each subsequence and its nearest neighbor. $MP \in \mathbb{R}^{(n-l+1)}$ and the data series motif is identified through the two lowest values is it.

To avoid trivial matches, the exclusion zone concept is used, according to which a region before and after a certain pattern is ignored. In this way, a pattern can't match itself or an almost identical series.
Once these definitions are given, it is useful to formalize the first problem that the algorithm solves. In fact, it aims to find all the motif pairs of all the lengths in a certain range that can be found in T. So, for all $l \in [l_{min}, \ldots, l_{max}]$.
The second problem that the algorithm solves is based on the concept of motif set. It is a definition of radius with respect to a motif pair $\{T_{a,l}, T_{b,l}\}$. In fact, a motif set S_r^l is defined as

$$S_r^l = \{T_{i,l} | dist(T_{i,l}, T_{b,l}) < r \vee dist(T_{i,l}, T_{a,l}) < r\}.$$

The cardinality of S_r^l is called the frequency of the motif set.
In this way it is possible to define another problem that Valmod solves, that is the possibility to find Variable-Length Motif Sets. This is defined as

$$S^* = \{S_r^l | S_r^l \text{is a motif set, } l_{min} \leq l \leq l_{max}\}.$$

The only restriction is that each subsequence can be included into a single motif set. More formally,

$$S_r^l, \bar{S}_{\bar{r}}^{\bar{l}} \in S^* \implies S_r^l \cap \bar{S}_{\bar{r}}^{\bar{l}} = \emptyset.$$

In the next chapter, we will use this algorithm for the detection of the motif sets. Then, the result that will be considered is the optimal length of l^* which identifies the Motif Sets with the higher cardinality.

To achieve the two objectives called up during the introduction, VALMOD starts by calculating the matrix profile for the minimum length in the range considered $[l_{min}, l_{max}]$. The intuition that leads to a significant reduction in the calculation of subsequent measurements is as follows. Note that if the l_{min}-length motif pair is $T_{i,l_{min}}, T_{j,l_{min}}$, then the motif $T_{i,l_{min}+1}, T_{j,l_{min}+1}$ will most likely be a motif pair for length $l_{min} + 1$. However, this is not always true, especially when l grows. However, it can be used as an initial intuition.

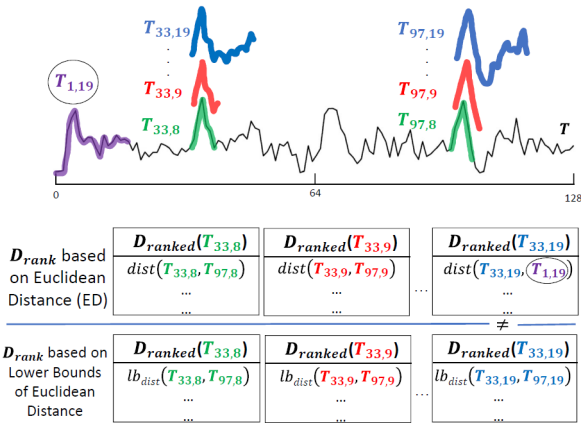

Figure 2.1. *(Top distance profile)* Ranking by true distance, based on Euclidean Distance, bring to a different ranking when l grows. *(Bottom distance profile)* Ranking order is preserved thanks to the Lower Bounds of Euclidean Distance.

In fact, although the distance profile values may change, a derived distance profile preserves the rank property. This is the lower bound distance profile.
To get a rough idea, this collects the lower bound distance between $T_{i,l+k}$ and $T_{j,l+k}$, $\forall k \in [1,2,3,\dots]$ by knowing the one between $T_{i,l}$ and $T_{j,l}$. This instrument will avoid many calculations and thus achieve a computationally efficient method to meet the two objectives mentioned above.
A more rigorous description of the creation of the lower bound distance profile is provided while the six algorithms on which VALMOD are fully described in the paper [LZPK18b].
From now on, the mean and variance of the subsequence $T_{x,y}$ will be indicated as $\mu_{x,y}$ and $\sigma_{x,y}$.
The problem can therefore be reformulated as follows. Knowing the distance $d^l_{i,j}$

between $T_{i,l}$ and $T_{j,l}$, the objective is to estimate a distance between $T_{i,l+k}$ and $T_{j,l+k}$. Last k values of $T_{i,l+k}$, both $\mu_{i,l+k}$ and $\sigma_{i,l+k}$, are unknown and can thus be considered as variables.

$$d_{i,j}^{l+k} \leq \min_{\mu_{i,l+k},\sigma_{i,l+k}} \sqrt{\sum_{p=1}^{l} \left(\frac{t_{i+p-1} - \mu_{i,l+k}}{\sigma_{i,l+k}} - \frac{t_{j+p-1} - \mu_{j,l+k}}{\sigma_{j,l+k}} \right)^2} \tag{2.17}$$

$$\min_{\mu',\sigma'} \frac{\sigma_{j,l}}{\sigma_{j,l+k}} \sqrt{\sum_{p=1}^{l} \left(\frac{t_{i+p-1} - \mu'}{\sigma'} - \frac{t_{j+p-1} - \mu_{j,l}}{\sigma_{j,l}} \right)^2} \tag{2.18}$$

The minimum value which is obtained from [Equation 2.17] can be set as the minimum possible $LB(d_{i,j}^{l+k})$. It can be solved by differentiating and imposing equal to zero.

$$LB(d_{i,j}^{l+k}) \begin{cases} \sqrt{l}\frac{\sigma_{j,l}}{\sigma_{j,l+k}}, & \text{if } q_{i,j} \leq 0; \\ \frac{\sigma_{j,l}}{\sigma_{j,l+k}}\sqrt{l(i - q_{i,j}^2)} & \text{otherwise.} \end{cases} \tag{2.19}$$

where $q_{i,j} = \dfrac{\sum_{p=1}^{l} \frac{(t_{j+p-1}t_{i+p-1})}{l} - \mu_{i,l}\mu_{j,l}}{\sigma_{i,l}\sigma_{j,l}}$.

Once the LBs have been ranked in an ascending order, one get the ranked lower bound distance profile

$$LB_{ranked}(D_l^{l+k}) = LB(d_{r_1,j}^{l+k}), LB(d_{r_1,j}^{l+k}), \ldots, LB(d_{r_{n-l-k+1},j}^{l+k}),$$

where $LB(d_{r_1,j}^{l+k}) \leq LB(d_{r_1,j}^{l+k}) \leq \cdots \leq LB(d_{r_{n-l-k+1},j}^{l+k})$.
The ranked lower bound distance profile is needed to speed the calculations. From now on, intuition will be provided.

A small number p is set and for each j, $LB(d_{r_p,j}^{l+k}) > dist_{BSF}$ inequality is verified. Where $dist_{BSF}$ is the distance of the best-so-far (BSF) pair of motifs. If the inequality hold true, it is necessary to evaluate only $d_{r_1,j}^{l+k}, d_{r_2,j}^{l+k}, \ldots, d_{r_{p-1},j}^{l+k}$. Otherwise it is necessary to compute all elements of D_j^{l+k}. Also $dist_{BSF}$ is updated when a smaller distance is found.
Thanks to this approach, to find the motif of length $l+k$, at least $O(np)$ operations are needed.

Chapter 3

Multivariate time-series preparation

Data preparation is crucial to understand which part of the dataset is most useful for the analysis. In this section, however, some transformations of the dataset will be discussed without drawing a clear and definitive conclusion on the most suitable pre-processing method for the analysis. In fact, as said in the introduction, the different pre-processed datasets will be compared by analyzing the clustering results, explained in the next part of the thesis.

So, the result obtained at the end of the chapter will be a set of datasets elaborated with different aggregations/granularities, which will be processed by the algorithms in the next part of the work.

3.1 Row data

All the data sent by the machine to the Tierra servers are text files weighing about 10Mb. For each line, there are a series of hexadecimal codes indicating the timestamp, the SPN (i.e. the signal identification code as explained in the introduction), and the measurement collected. From this file, it was then necessary to translate the signals into their alphanumeric understandable version.

All SPN codes refer to standard CAN signal messages only up to the number 10000. It means that any vehicle using the protocol mentioned has a unique association between an SPN code and its meaning in terms of the signal it refers to, only up to SPN 10000. All other SPNs are custom. Therefore they depend on customer specifications and they are often subject to corporate restrictions.

In this case, the company vehicle had a summary file of all the SPN codes which was used for the translation.
The interpreted data were stored in a single compact file and a preliminary analysis was carried out.

3.1.1 Preliminary analysis

The dataset is composed of 20 correctly interpreted SPNs and the number of signals for each one varies a lot as shown in [Figure 3.1].

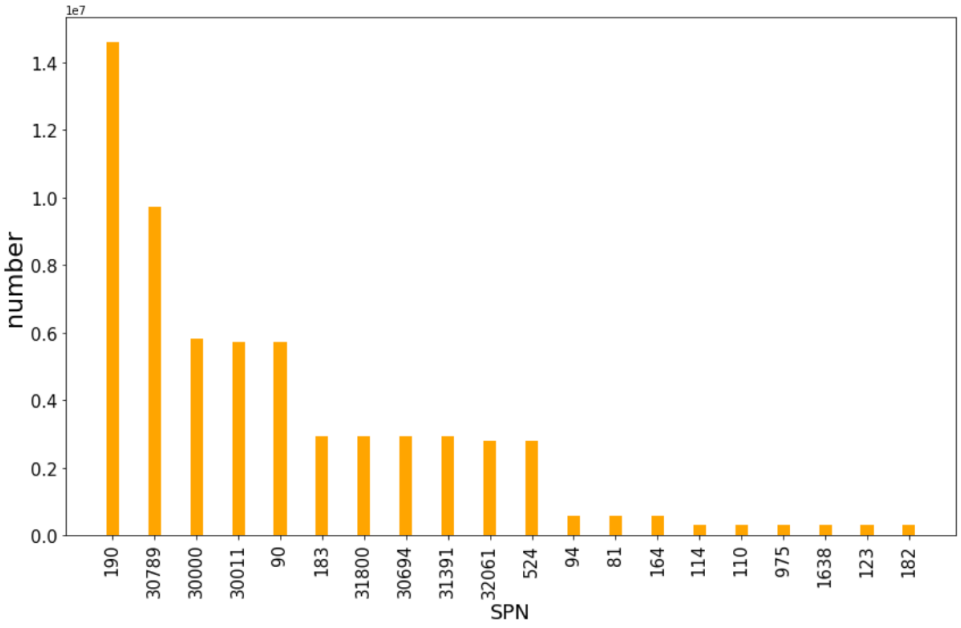

Figure 3.1. Number of messages for each SPN in the entire dataset.

The data has been collected between 7 November 2019 and 15 April 2020, but only 40 days within this period have been used to actively record them. A schematic picture of the amount of data collected for each day is given in [Figure 3.2].

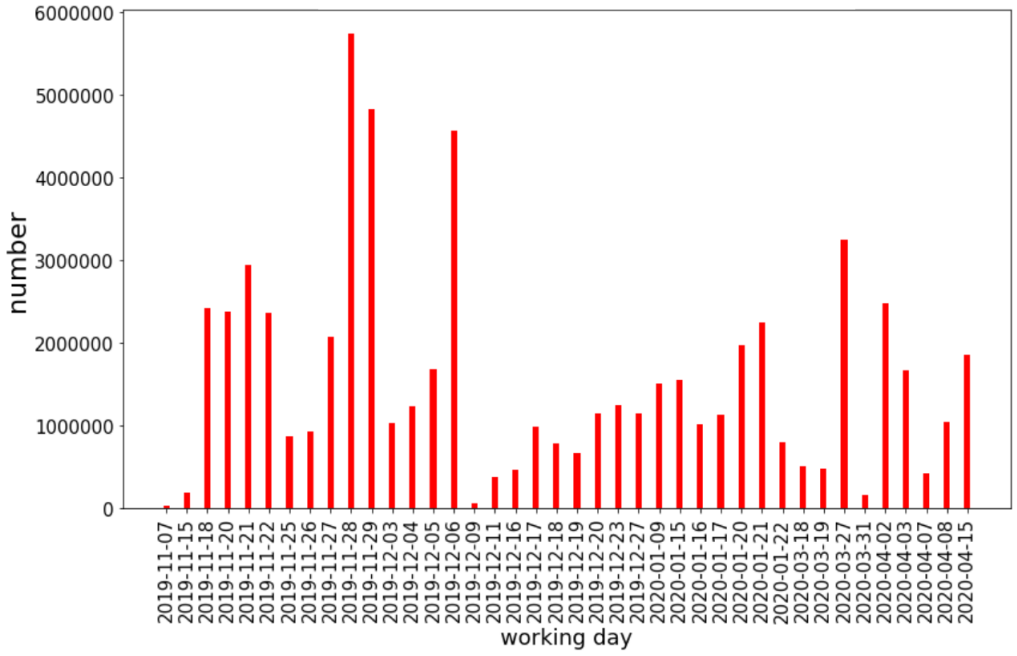

Figure 3.2. Number of signals per working day.

The dataset does not contain NaN values or missing values, however, it contains duplicate lines that have been removed. Moreover, the dataset had some inconsistencies: because of the high frequency of the signals, in some cases, two different values for the same combination (SPN, timestamp) were recorded. In these cases, because there were no significant discrepancies between the two values, the mean value was considered. Afterward, the signals that had a constant value, and those that were completely out of the feasible range have been removed. In conclusion 13 SPN on which to begin the preliminary analysis have been obtained.

3.1.2 Rate detection and work cycles

Since the data was raw and there was no additional information on the machine's operation, one of the objectives of the pre-processing analysis was to identify the sampling rate of each SPN and the work cycles. The sampling rate is the quite constant rate at which the individual components of the vehicle send signals to the control unit. To calculate it, a new feature was calculated: the time difference between one observation and the following for each SPN. The average value and the median of the frequency of these features were found (one for each SPN). Besides, by qualitatively examining the time difference distributions, it was found that the majority of the time differences coincided with the average and the median of the

distribution. The main reasons why some time differences did not coincide with these values have been identified.

- A signal delay in the control unit of some milliseconds.
- An anomalous interruption of the signal.
- The vehicle turned off.

However, since the distribution of these features was very concentrated in the median (which coincides with the average value in all SPNs), this value has been considered as the signal acquisition rate for each SPN (see [Figure 3.3]).

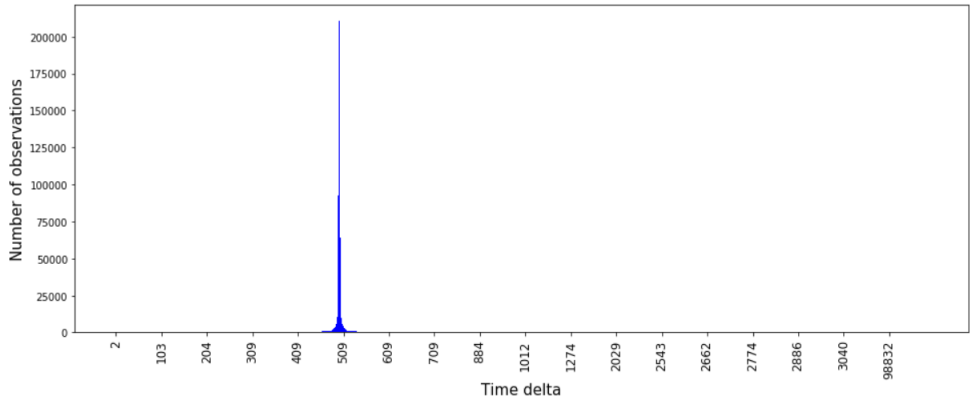

Figure 3.3. In blue, the bars representing the number of occurrences of a certain time difference between two consecutive observations of a specific signal: the number 94. This is just an example since all SPNs report the same graph.

The purpose of finding the original sampling rates is the following. The average operator and the downsampling operators, as intended in the theory chapter, work very well when the signal is sampled at constant rates. Less intuitively, one could treat the signal with other techniques to preserve the sampling irregularities, but these alternative methods have been avoided for two reasons. From the preliminary analysis and thanks to the comments of domain experts, preserving all the information would have been useless for the final goal. Besides, the vehicle studied in this work is not so precise to require such peculiar preservation of information. To obtain a regularized signal, therefore, it was necessary to center the observations and align them in a homogeneous time axis. This process has been carried out, without complications, for every SPN and if one observation overlapped with the next because it was incorrectly sampled, then the average with the following observation was taken. This last case, however, has happened a few times and does not deserve to be deepened. In the conclusion of this part, all signals have

been aligned at a constant clock.

As announced at the beginning, the second goal of this part is to find the "work cycles" of the vehicle. It means a period of time in which the machine is working continuously. It was found by considering a single significative SPN: the engine coolant temperature [Figure 3.4].

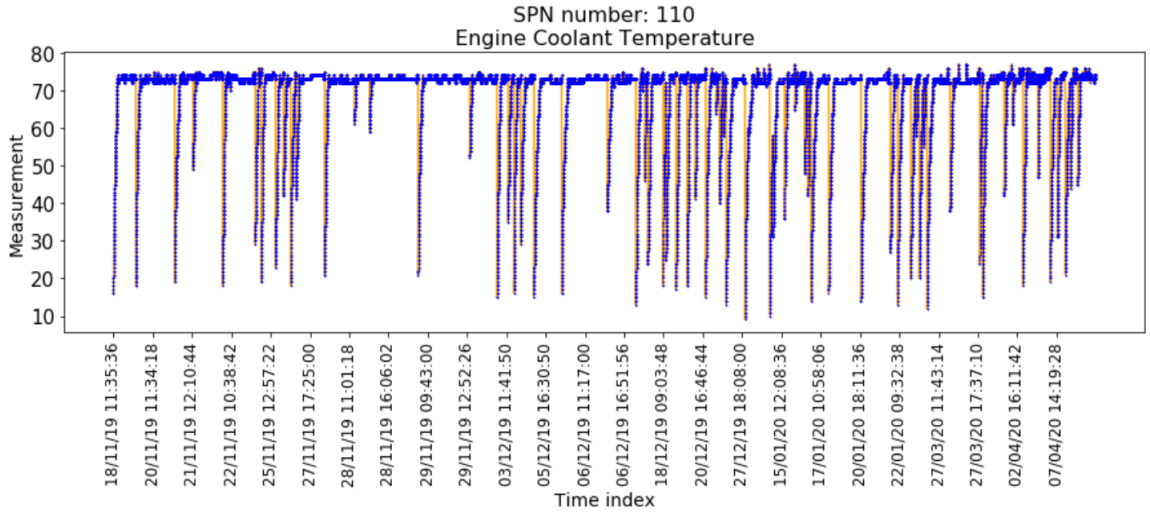

Figure 3.4. Plot of SPN 110 (coolant temperature) over time.

This signal represents the cooling liquid temperature trend and it helped to detect the moment of inactivity from work periods. This signal has a shape that strongly depends on the working cycles, as the machine heats up and cools down according to the use of the vehicle. Therefore a temperature threshold (below which the vehicle is considered to be in a heating phase) has been set by domain experts and it has been exploited to delimit the work cycles [Figure 3.5] by highlighting the moment of stability of this signal.

3.2 Multivariate time series alignment and aggregation

This section is dedicated to the procedure used for the creation of the final datasets which will be examined in the next part.

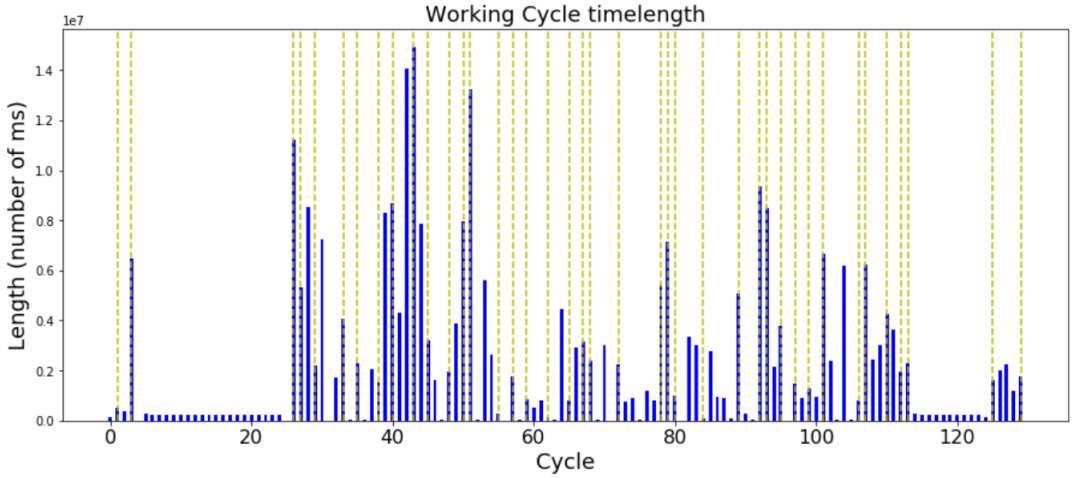

Figure 3.5. Blue bars represent the length of the working cycles identified. The yellow horizontal dotted lines represent the separation between days. This is to emphasize that, within one working day, several work cycles have been identified.

It collects the experimental results obtained from the tests carried out on relevant signals to the thesis purpose. In fact, the aggregation methods and the granularity has been chosen through the analysis of the results of the following experiments. Some other consideration has been suggested by continuous comparisons with domain experts.

3.2.1 Downsampling and aggregation

This part is dedicated to the choice of the best aggregation and downsampling methods that have been used to preprocess the dataset. The theoretical details on which this part is based are all reported in [Section 2.1].

Three methods have been compared: aggregation by averaging and downsampling by two transforms, one based on the Fourier transform and the other on the Cosine transform. The three methods obtain the same purpose in terms of point aggregation and signal synthesis, however, they achieve the purpose in a completely different way. By using an average operator, the signal is synthesized in such a way that the information is lost for sure. In the case of downsampling, on the other hand, it is known which frequency of the starting signal will be lost and which one will be preserved. In extremely regular cases, all the information could be retained (see [Subsection 2.1.4]). The synthesis of information through the sampling of the moving average has been used in the following analyses because it is a type of

aggregation used to pre-process signals by the company Tierra. So, to compare this method with one based on transforms, it was necessary to understand which of the two selected transforms was more suitable for the signals studied.
The first one uses a grade 8 Chebichec polynomial anti-aliasing filter. The second is based on the cosine transform. In the second method, the simple truncation of the expanded series at the appropriate level allows avoiding the aliasing phenomenon.

Regarding the level of aggregation, the following considerations have been made. The signal sampling rates range from 50Hz up to 1Hz. As mentioned in the previous section, the data obtained from the vehicle are noisy, and sometimes this high frequency is not necessary for an agricultural vehicle, therefore an aggregation or downsampling leads to an improvement since the signal is seen at a lower level of detail and these operators can smooth possible errors. However, to find a minimum level of detail, optimal to improve the signal without capturing noises, it was needed a comparison with the domain expert, who suggested to start from a granularity of 0.5Hz. The purpose to fix the optimal maximum level of aggregation was not to flatten the signal and to allow correct identification of the shape that characterizes the signal also after the smoothing operation. Experimentally, it was obtained that the optimal upper level for this purpose is 0.25Hz. In conclusion, the following levels of aggregation have been considered for the rest of the analysis: 2Hz, 1Hz, 0.5Hz e 0.25Hz.

The first comparison result between downsampling methods is quantitative. In fact, the comparison was made using the DTW measurement between downsampled series with the two transforms and the original series (see [Section 4.1.1]). The distance between the original series and the respective downsampled signals has been evaluated for all levels of aggregation and both methods. In this way, an objective measure of goodness that could be compared was evaluated.

The result obtained was ambiguous. There was no better outstanding method. In fact, even if the method that uses the Fourier transform seemed to perform better, it is not much better than the other one. An example summary table about the signal SPN 524 is reported. Two reference work cycles have been taken as an example, the longest (in terms of the number of milliseconds that ranges) and one of the shortest.

dimension work cycle	frequency (HZ)	distance Fourier	distance Cosine
large	2	32.16	65.59
large	1	402.04	422.81
large	0.5	441.70	457.38
large	0.25	509.55	538.59
small	2	230.36	589.65
small	1	8935.91	6468.68
small	0.5	9076.24	9333.66
small	0.25	7323.38	8601.01

Table 3.1. Table summarizing the results of distance between the two work cycles (large and small) and downsampled signals. The second column shows the frequency of the result signal after downsampling procedure. The last two columns are the similarity measurements.

Excluding one case, the comparison of the signal processed with the Fourier method gives better results in this situation, but also other signals give quite similar outputs.
The first consideration is that the results of the Fourier method are better but not significantly, the second consideration is that it is not always verified.

Since these results did not give a clear and well-defined choice, it was necessary to switch to a graphical and subjective comparison of the results obtained. The comparison that was made concerned the graphs of the reconstructed series at the same levels of granularity but with the two aggregation methods. In this way, it was possible to discover some added criticality for the analysis. It was noticed that the downsampled signal through the Cosine transform had the addition of oscillations near to the sudden changes of the signal, which can be easily explained from the theoretical point of view, as no filter was applied in this procedure. This was critical for subsequent analysis in which the similarities between time windows are identified by the occurrence of repeated patterns and motifs. For this reason, the Fourier-based method was preferred, where this criticality is not evident.

3.2.2 Time series optimal window

So far, the most suitable granularity and aggregation/downsampling methods have been identified, but to conduct the next analysis it is necessary to set two important parameters: the window length and window shift (see [Section 2.2]). The window shift determines the time-series clustering method. In this case, since the data are used for mining purposes a not-overlapped (jumping) method seemed more appropriate for the dataset division in windows.

To choose the optimal length of the window, two scenarios were possible: a Domain-driven or Data-driven approach. For the final goal of the analysis, however, it seemed more appropriate to use a fixed window length. In fact, given future applications in the real-time recognition of states, using a domain-driven approach considerable complications could occur. With a fixed-length, instead, data would be automatically packaged in windows without the need for previous calculations. To obtain the right length of the time window, a method based on the Matrix profile and Motif Discovery concept has been used: VALMOD.
As explained in the appropriate section (see [Section 2.3]) this method is used for a different purpose. It identifies all motif in a given time window length interval. In this case, it is used to compare the number of motif occurrences for each time length in a given range to select the most appropriate window length to be used in the time series segmentation. To understand which is the right interval of lengths to consider, a domain expert was consulted. Based on the characteristics of the machines involved and based on the purpose of the analysis, an interval of possible lengths was identified: from 2 to 10 minutes. Before proceeding with the analysis, then, the cycles with a duration of less than 10 minutes were discarded from the analysis, as this was the minimum length of a window in the upper extreme case. The optimal length found through the analysis of the VALMOD algorithm results was 2 minutes, considering all the datasets.
At the end of this process, the data was transformed in the following way. The eight datasets (four granularity times two aggregations/downsampling methods) were built. Each column of the dataset represents an SPN signal and the timestamps identify the lines, aligned according to the unique clock. A total of 2332 windows was obtained.

	94	110	182	183	190	524	975	30000	30694	30789	31391	31800	32061
1574073336000	144.0	16.0	11321.5	7.900000	1012.125000	1	100.0	55.000000	88.400000	215.000000	0.0	16449.0	0.800000
1574073337000	144.0	16.0	11321.5	7.900000	1013.035714	1	100.0	54.600000	88.400000	217.000000	0.0	16449.0	1.066667
1574073338000	144.0	16.0	11321.5	7.900000	1008.381579	1	100.0	53.500000	88.400000	216.071429	0.0	16449.0	1.200000
1574073339000	144.0	16.0	11321.5	7.931250	1006.808594	1	100.0	54.100000	88.400000	216.521739	0.0	16449.0	1.200000
1574073340000	144.0	16.0	11321.5	7.800000	1039.761111	1	100.0	56.846154	88.400000	216.551724	0.0	16449.0	1.200000
1574073341000	140.0	16.0	11321.5	8.475000	1095.649194	1	100.0	55.076923	88.400000	217.200000	0.0	16449.0	1.200000
1574073342000	146.0	16.0	11321.5	8.700000	1122.871951	1	100.0	51.250000	88.400000	217.321429	0.0	16449.0	1.200000
1574073343000	144.0	17.0	11321.5	8.700000	1147.981707	1	100.0	51.555556	88.400000	217.500000	0.0	16449.0	1.200000
1574073344000	144.0	17.0	11321.5	8.654545	1136.752500	1	100.0	49.714286	88.400000	217.428571	0.0	16449.0	1.200000

Figure 3.6. Head of the dataset sampled with a granularity of 1Hz.

3.3 Exploratory data analysis

In this section, some preliminary analyses are carried out to identify the most useful series for the thesis. First, the autocorrelation was calculated to understand the patterns inside the single series. The values that can assume this function are between -1 and 1. In these extreme cases, high autocorrelation is present, while a value towards the zero indicates that there is no autocorrelation. Among the 13 SPN, three categories have been found according to the autocorrelation function. A graph of these categories is reported in [Figure 3.7]. The analysis that follows is based on the aggregated dataset with the average method and with a granularity of 1Hz. In fact, comparing the results obtained with the other datasets, significant differences did not emerge. Therefore the analysis is to be considered valid without loss of generality for all datasets.

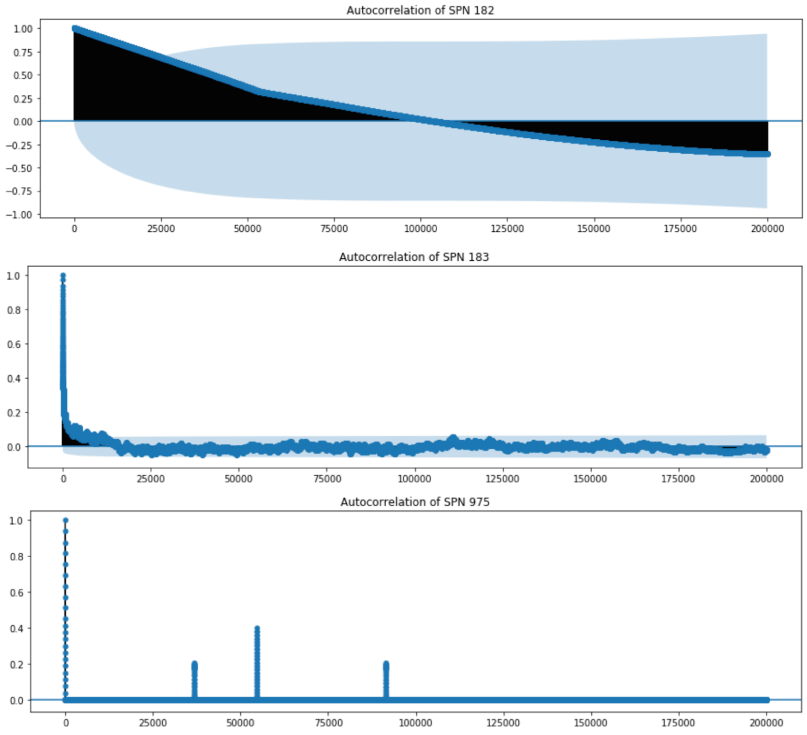

Figure 3.7. The three "types" of SPN signal autocorrelation represented by SPN 182, 183, 975. The first shape represents only SPN 182 and 30694 and it indicates a constantly increasing trend. The second graph reflects a trend without autocorrelation, being almost constantly at zero levels. The third is the typical trend of almost constant signals with peaks in certain areas. These lead to very high autocorrelation at those points.

The autocorrelation curve of the SPN 182, which represents the first category, is smooth and decreases steadily. If one could represent the autocorrelation for lags over 200000, it will finally rise to zero. This autocorrelation trend is linked to an increasing trend in the time-domain representation [Figure 3.8]. In this case, SPN 182 represents the total fuel consumed and therefore the level rises constantly.
The two SPNs that present this particular form (SPN 182 and 30694) will be analyzed later to understand if they could be useful for further analysis.

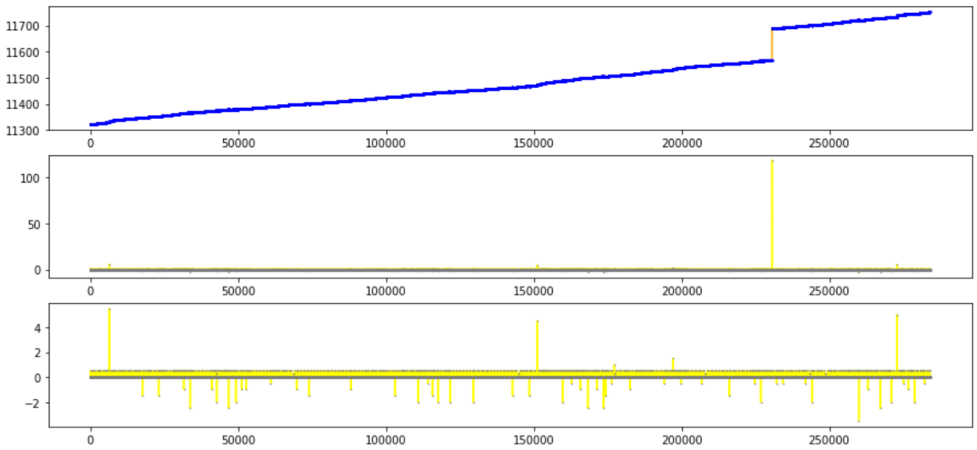

Figure 3.8. On top, the time-domain representation of SPN 182. In the center, the same signal after the "Trend Elimination by Differencing" technique. At the bottom, the same graph proposed in central position, plotted without the extremely large values.

The second group is represented by SPN 183 and presents an autocorrelation graph that starts from value 1 and then, descends very quickly to zero, remaining almost constant for the rest of the lags. The majority of SPNs belong to this category and in the time domain, they appear as random series. These signals present neither trend nor seasonality and therefore are considered in the following analysis without other pre-processing steps. For this category also the partial autocorrelation has been observed and a result that agrees with the previous was obtained. In fact, the curve goes down to zero even faster and then remains stable for the rest of the lags [Figure 3.9].

The third category represented by SPN 975 presents sudden and sporadic variations in the time domain. However, most of the time, this category of signals remains constant on a certain value.
The vehicle considered, as mentioned above, is about 12 years old, and some components could not work well anymore. Probably, these series are the result of

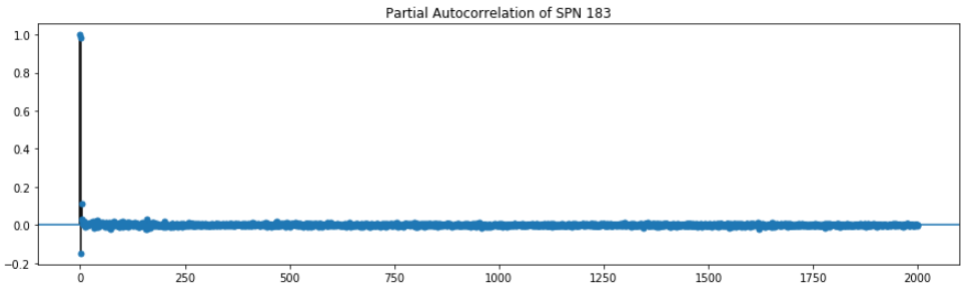

Figure 3.9. Partial correlation graph of the SPN 183.

malfunctions. The autocorrelation graph in these cases detects a very high correlation in the points where peaks appear. These SPNs were removed from the dataset for further analysis to avoid errors and for the impossibility to identify significant behaviors.

At this point, only the most useful portion of the dataset has been selected.
In fact, all SPNs that involve external components of the vehicle and that do not compromise the use of the engine, such as the position of the rear trailer hitch (SPN 30694), have been removed from the dataset.
Moreover, the SPN 182, representative of the first category of autocorrelation analysis, has been removed due to its low significance. Let's look at its graph [Figure 3.8] after the trend has been removed through the "Trend Elimination by Differencing" method. Many negative values indicate a bad functioning of the float that keeps track of the fuel level. Moreover, even after removing the negative values, the transformed signal does not appear significant for the analysis, not presenting relevant variations.
The last signal to delete before continuing is the number 110, the one representing the engine coolant temperature. In fact, although fundamental for the analysis of the identification of work cycles, it has a trivial behavior throughout its duration. At the beginning of the work cycle, the temperature rises, and then during the work cycle, it stabilizes at a fixed value, without generating any kind of useful information (see [Figure 3.10]).

The next step was the analysis of the correlation between the remaining SPNs. Before doing this, however, it was appropriate to conduct a statistical test to verify the stationarity of the SPNs; the Augmented Dickey-Fuller test. The test confirmed that it was not necessary to conduct further processes to the SPNs considered.
The correlation between the remaining SPNs has been considered and a strong correlation between 183 and 190 has been observed. The 183 represents the instantaneous fuel consumption while the 190 represents the engine speed. Therefore, the correlation is easily explained by the fact that an increase in engine speed causes an increase in instantaneous fuel consumption. The variable 183 has been removed from the dataset.
The variables left at the end of this chapter are the following.

- 94. The fuel delivery pressure in the vehicle engine. This parameter is influenced by engine speed but more generally considers the fuel requirements to carry out all the activities that the vehicle is conducting.
- 190. The signal represents the engine speed, which is essential to understand

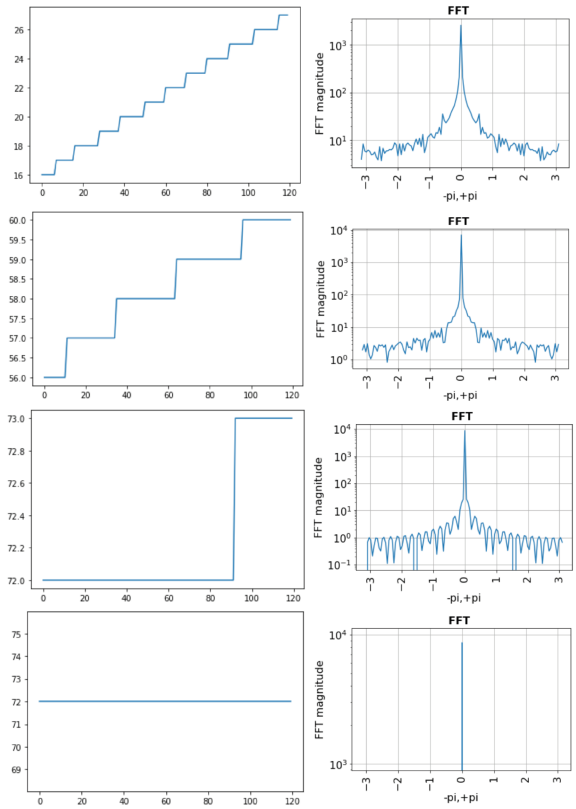

Figure 3.10. Four windows of the SPN 110, from when the temperature of the liquid starts to rise faster until it becomes constant at the maximum level. The 0, 1, 6, 11 windows are represented in sequence. This example reinforces the hypothesis of signal exclusion. In fact, it has a trivial behavior for the whole duration; every time a work cycle starts, it goes up for about 15 minutes with a staircase trend, and then, it stabilizes at a constant value.

the speed of the vehicle and the type of activity it has to perform.

- 524. This variable represents the vehicle's gears. It will have values between 1 and 5 as this vehicle has 5 gears.

Finally, it is reported a graph with all the SPNs signals. The shown dataset is always the one aggregated by average with a granularity of 1 Hz [Figure 3.11]. This is the representation for the rest of the thesis; blue points represent the single SPN messages, while the orange line connects the points.

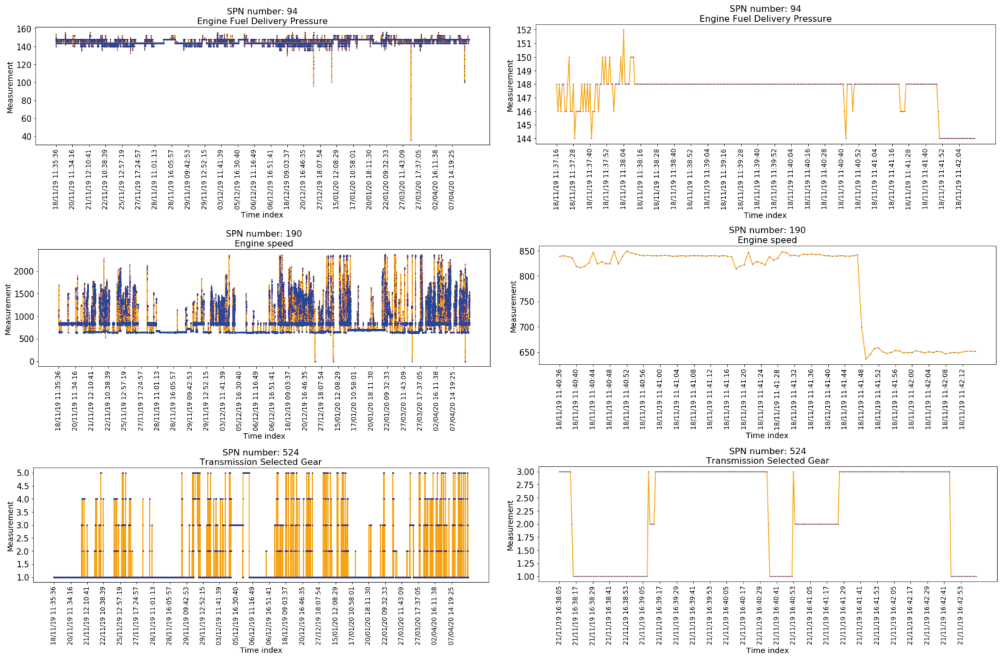

Figure 3.11. The three SPN 94, 190, 524 represented in the time-domain. On the left, it is represented the entire time domain, while on the right, some zooms are shown.

Chapter 4

Multivariate time series clustering techniques

This chapter collects all the mathematical tools that will be used in the next chapter. In particular, the first part deals with the most common methods to establish how similar two time-series are, based on their shapes: Shape-Based Distance and the Dynamic Time Warping. These allow understanding the similarity of two series considering all the shifts. Then, clustering algorithm techniques are reported. These will be analyzed both from an algorithmic and theoretical point of view.

4.1 Distance measure for Time-series clustering

As said above, this section is dedicated to the two main similarity measures exploited to perform the clustering algorithms. In fact, the classic Euclidean or Manhattan distances are not sufficient in the time-series context; very often the shapes that characterize a certain time-series window do not overlap perfectly in time with the ones in similar windows; the reason is the time shift. With the methods described below this limit is overcome.

4.1.1 Dynamic Time Warping

This part is dedicated to an in-depth study of Dynamic Time Warping (DTW), a dissimilarity measure that helps to find disparities between time series more effectively than the simple Euclidean point distance. Part of the concepts is based on the [Mül07].
The invention and use of DTW stem from the need to measure distances between time series that are not aligned or even of different lengths. Originally it was used to compare sequences in speech recognition and subsequently DTW was applied in

all contexts with time-dependent data. But now, let's deep into the technicalities. As said, the considered time series can also be of different length, so let's consider two series $X \coloneqq (x_1, \ldots, x_N)$ and $Y \coloneqq (y_1, \ldots, y_M)$ where $N, M \in \mathbb{N}$. These can be time-series or more generally sequences sampled at regular clock. $\mathscr{F}$ indicates the feature space and so $x_n, y_m \in \mathscr{F} \ \forall n \in [1:N]$ and $m \in [1:M]$. It is necessary to define a local cost measure which is a function c,

$$c : \mathscr{F} \times \mathscr{F} \to \mathbb{R}^+.$$

This function define a cost matrix where each element is represented by $c(x_n, y_m)$,

$$C(n, m) \coloneqq c(x_n, y_m).$$

The optimal path within this matrix produces the distance between the two sequences. The following definition will formalize the alignment concept.

Definition 4.1 An $(N, M)-$warping path is a sequence $p = (p_1, \ldots, p_L)$ with $p_l = (n_l, m_l) \in [1:N] \times [1:M]$ for $l \in [1:L]$ which satisfy the conditions:

- Boundary condition: $p_1 = (1,1)$ and $p_L = (N, M)$.
- Monotonicity condition: $n_1 \leq \cdots \leq n_L$ and $m_1 \leq \cdots \leq m_L$.
- Step size condition: $p_{l+1} - p_l \in \{(1,0), (0,1), (1,1)\}$ for $l \in [1 : L-1]$.

The third condition implies the second but it has been quoted for sake of completeness.
The three conditions implicitly explain the process of path creation. In fact, the optimal path must start and reach the extreme points of the two series according to the first condition. This forces the algorithm to consider a warped alignment of the series. The monotonicity condition, instead, imposes to preserve the temporal order of the series. This suggests a recursive approach. Step size condition can be seen as a sort of continuity condition, which implies that all the elements of the two series must be considered.

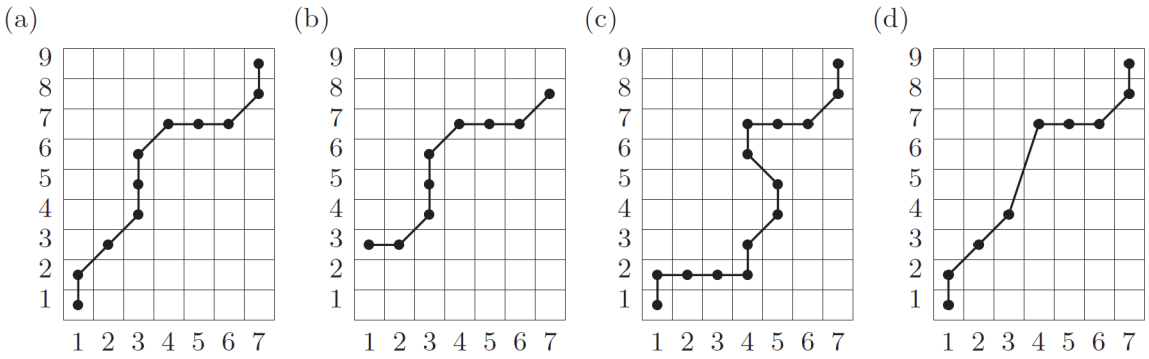

Figure 4.1. The four illustrations represent paths of index pairs of two sequences, one 9 in length and the other 7. (a) A path that satisfies the three properties. (b) Boundary condition is violated. (c) Monotonicity condition is violated. (d) Step size condition is violated. ([Mül07])

From the previous definitions the total cost of the warping path p between X, Y with respect to the local cost measure c is defined as following,

$$c_p(X, Y) := \sum_{l=1}^{L} c(x_{n_l}, y_{m_l}).$$

But now it is crucial to define the optimal warping path between X and Y. It is simply the one that reaches the minimum cost among all possible warping paths: p^*. The DTW dissimilarity measure is defined as the total cost of the optimal warping path:

$$DTW(X, Y) := c_{p^*}(X, Y).$$

It is important to underline that the DTW dissimilarity is well defined also if there are more optimal paths. Moreover, DTW is symmetric only in case c is symmetric. Last, the DTW cannot be considered a metric since it does not satisfy the triangular equality. This last fact is easily demonstrated with counter-examples.

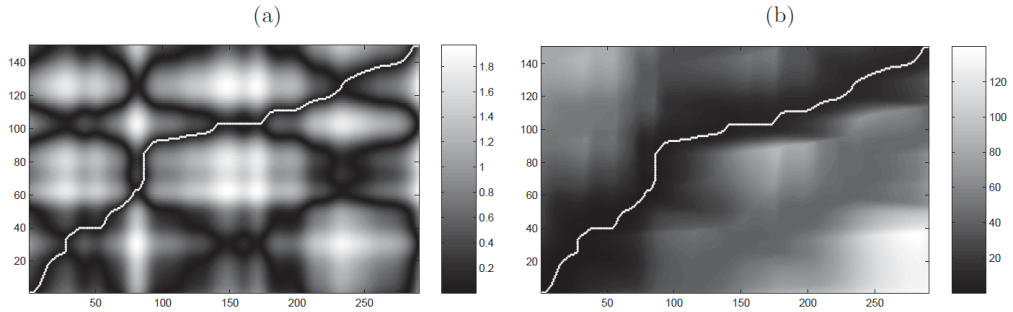

Figure 4.2. On the left, a cost matrix is represented in a black and white heat map. Low costs are indicated by dark colors and vice versa. On the right a graph that represents accumulated cost matrix D. In white, an optimal warping path is drawn. ([Mül07])

Now that the DTW has been defined, the methods by which this measurement is iteratively evaluated will be presented. In fact, the three properties can be transformed into constraints and the result can be evaluated through dynamic programming. In the case of this thesis, an algorithm contained in the package [SESL17] has been exploited. It is a symmetric algorithm so that two out of three metric properties are met (positivity and symmetry).

The process for DTW calculations is carried out in steps. For sake of completeness the whole process is described for multivariate time series, so that can be easily understood also in the univariate case.
The first step aims to find a local cost matrix (lcm) that has $n \times m$ entries. This is built for each pair of distances. Let's consider X and Y as input series and let's indicate with X_i^v the i-th element of the $v-th$ variable of the multivariate series. From now on, all multivariate series have the same length as in the cases considered in the following chapter. However, it is easy to generalize even in cases with different length series. Let the cost matrix be evaluated as in [Equation 4.1] for each element (i, j).

$$lcm(i,j) = \left(\sum_v |X_i^v - Y_j^v|^p \right)^{1/p} \tag{4.1}$$

Secondly, the DTW algorithm uses the lcm to evaluate the optimal path, starting from the point (1,1) and arriving at the point (n, n), in case the two series are of the same n-length. In the end, an ideal path is defined as $\phi = \{(1,1), \ldots, (n, n)\}$ which contains the nodes that must be joined for each of the two series. The final result of the DTW is a sum of the distances obtained as in the equation [Equation 4.2].

$$DTW_p(X,Y) = \left(\sum \frac{m_\phi lcm(k)^p}{M_\phi} \right)^{1/p} \ , \ \forall k \in \phi \tag{4.2}$$

m_ϕ is a per-step weighting coecient and M_ϕ is the corresponding normalization constant ([Gio09]).
In this definition of DTW, the p-norm is used twice, both in the first and second equation. However, the [Equation 4.1] is affected by the p-norm only if the series is multivariate.

4.1.2 Extension of the DTW

As previously discussed DTW dissimilarity cannot be considered a distance because of its inconsistency with the definition itself. A measure D, that induces a metric in the space of the time series, satisfies the following properties for each triplet of time series x, y, z:

1. Positivity: $D(x,y) \geq 0\,;\, D(x,y) = 0 \iff x = y$;
2. Symmetry: $D(x,y) = D(y,x)$;
3. Triangle inequality: $D(x,y) + D(y,z) \geq D(x,z)$.

However in the more classical algorithms for the DTW calculation, properties 2) and 3) are not verified. To satisfy property 2), it is sufficient to use an algorithm that symmetrically calculates the DTW and in the literature, there are routines for this task (also implemented in the most common libraries). The algorithm used in this thesis is symmetric and so this property is met (see [SESL17]).
The triangle inequality, instead, is very problematic to enforce. However, some studies show that this property, in practical cases, is respected very often. See for example [RNS85] or [Jai18] which state that the property is met in 99% of the cases explored.
So, from now on, let's consider the DTW dissimilarity measure, a distance measure (improperly), by considering the third property always met.

Now that this clarification has ended, the purpose of the following part is to extend the concept of DTW by proving the following statement. *By applying DTW to individual one-dimensional time-series and then aggregating them via C and M function, the result is a distance.*
But now, let's properly formalized this sentence.
Let V_n^p be a set of all the possible multivariate time-series with n dimension of length p. Let's consider the following two functions.

$$M(x,y) = \sum_i DTW(x_i, y_i); \tag{4.3}$$

$$C(x,y) = \max\{DTW(x_i, y_i) : i = 1, .., n\}, \tag{4.4}$$

where x and y are any two time-series and $DTW(x_i, y_i)$ refers to the DTW distance between the same features.
The aim is to prove that these two are (pseudo) distance too.
Let x,y, and z be three generic multivariate time-series that belong to V_n^p. First, let's prove that summing the DTW distances [Equation 4.3] leads to a new metric.

1. The first aspect required is the positivity. Using the hypothesis

   $$DTW(x_i, y_i) \geq 0 \text{ and } DTW(x_i, y_i) = 0 \iff x_i = y_i \qquad \forall i \in \{0, \ldots, n\},$$

   it follows that

   $$\sum_i DTW(x_i, y_i) \geq 0 \text{ and } \sum_i DTW(x_i, y_i) = 0 \iff x = y.$$

2. Since $DTW(x_i, y_i)$ is a distance for all $i \in 0, \ldots, n$, the equality

   $$DTW(x_i, y_i) = DTW(y_i, x_i)$$

   holds. Then, summing on both sides,

   $$\sum_i DTW(x_i, y_i) = \sum_i DTW(y_i, x_i) \quad \forall i \in \{0, ..., n\}. \tag{4.5}$$

3. As we assume that DTW is a metric, the triangle inequality is satisfied by each single feature. Thus,

   $$DTW(x_i, y_i) + DTW(y_i, z_i) \geq DTW(x_i, z_i) \qquad \forall i \in \{0, ..., n\} \tag{4.6}$$

   Again, summing on both sides we conclude that

   $$\sum_i DTW(x_i, y_i) + \sum_i DTW(y_i, z_i) \geq \sum_i DTW(x_i, z_i). \tag{4.7}$$

Let's now pass to the second function, which consists in taking the max distance between any component of two time series [Equation 4.4].

1. As before, the first step is the positivity. Using again the hypothesis

   $$DTW(x_i, y_i) \geq 0 \text{ and } DTW(x_i, y_i) = 0 \iff x_i = y_i \qquad \forall i \in \{0, ..., n\},$$

   it follows that $\max\{DTW(x_i, y_i) : i = 1, .., n\} \geq 0$ and

   $$\max\{DTW(x_i, y_i) : i = 1, .., n\} = 0 \iff x = y.$$

2. Since $DTW(x_i, y_i)$ is a distance for all $i \in \{0, \dots, n\}$, the equality

$$DTW(x_i, y_i) = DTW(y_i, x_i)$$

holds. Then, by taking the max on both sides,

$$\max\{DTW(x_i, y_i) : i = 1, .., n\} = \max\{DTW(y_i, x_i) : i = 1, .., n\}. \quad (4.8)$$

3. Let's consider the index j such that $j = \max_i DTW(x_i, z_i)$. Since we assume that DTW is a metric, the triangle equality is satisfied for each feature, also the j-th. Thus,

$$DTW(x_j, z_j) \leq DTW(x_j, y_j) + DTW(y_j, z_j) \quad (4.9)$$

By definition of max, both the followings statements hold true

$$DTW(x_j, y_j) \leq \max\{DTW(x_i, y_i) : i = 1, .., n\}$$

and

$$DTW(y_j, z_j) \leq \max\{DTW(y_i, z_i) : i = 1, .., n\}.$$

In conclusion

$$\begin{aligned} \max\{DTW(x_i, z_i) : i = 1, .., n\} \ \leq \ & \max\{DTW(x_i, y_i) : i = 1, .., n\} \\ & + \max\{DTW(y_i, z_i) : i = 1, .., n\} \end{aligned} \quad (4.10)$$

4.1.3 Shape-Based Distance (SBD)

The SBD was initially proposed in the article on the k-shape algorithm [PG15], which will be discussed in [Subsection 4.4.3]. In the article, the algorithm is presented as a faster version of the DTW. This is based on the cross-correlation with the coefficient normalization (NCCc) sequence between two series. This is therefore very sensitive to scale factors and for this reason, it is always advisable to use a z-normalization before applying it. The NCCc is obtained by convolving the two series, so, similar shapes are identified even if the characteristic features are shifted. However, point-wise warpings are never performed. Convolution is performed by Fast Fourier Transform (FFT) and this speeds up the calculations. In conclusion, the distance formula can be expressed as follows,

$$SBD(x, y) = 1 - \max\left(\frac{NCCc(x, y)}{||x||_2 ||y||_2}\right),$$

where the $||\cdot||_2$ is the l_2 norm. The distance ranges from 0 to 2 where 0 indicates perfect similarity. The article does not extend the distance in the multivariate case since k-shape is proposed only in the univariate context.

4.2 Time-series prototypes

This section aims to define the various methods that will be used to calculate summary information about a set of time-series that share the same label (are the same cluster). These methods are important in the partitional clustering methods, explained later. In fact, they rely on a reference central point for each cluster, and to identify it, a prototype time-series function is necessary to summarize all series in a given cluster. The three methods of prototyping used are Partition Around Medoids (PAM), DTW barycenter averaging (DBA), and Shape extraction one.

4.2.1 Partition Around Medoids (PAM)

This first method is also very common in classical clustering methods. This one evaluates the central element of a dataset by considering the intra-cluster distance. In fact, it defines the cluster representative as the element that has a minimum average distance from the other elements. In this way, an element already existing in the dataset is chosen. This method is convenient in case the distance matrix is already fully calculated, as it is possible to use it at each iteration of the prototyping evaluation.

4.2.2 DTW barycenter averaging (DBA)

In the previous section, the DTW has been explained. Now, a prototyping method based on this measurement is presented. The article on which this part is based is [PKG11]. This is an iterative and global method that means not influenced by the order of the series. It assumes that the series are grouped in clusters and a representative is found as a reference. Then, the time-series of the cluster are aligned with the chosen centroid through the warping algorithm. Subsequently, all points of all the series (general elements contained in the cluster) that correspond to the first point in the reference series (centroid) are grouped according to the DTW alignments, and the mean is computed. The algorithm proceeds in this way for each point of the centroid series. The result is the DBA.
It is important to note that there can be also many points of the cluster series that correspond to only one point of the centroid series because of the warping alignment.

4.2.3 Shape extraction

This prototyping method was firstly explained in the article [PG15] and it is directly linked to the clustering algorithm that it deals with. As for the previous prototyping method, starting from a set of series, a reference series μ^* is obtained.

An NCCc-based alignment is performed and this process can be seen as a shifting of the various series to overlap the same shapes of the series in the same points 4.3. The method to obtain the reference series can be written as an optimization problem where one looks for the series that meets the following condition.

$$\mu^* = \text{argmax}_\mu \sum_{x \in X} NCCc(x, \mu)^2, \tag{4.11}$$

where X is the set of time series.

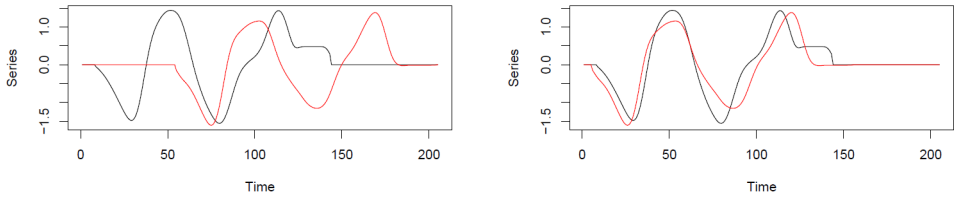

Figure 4.3. This is an example of the NCCc-based alignment performed on two sample series. On the left the series before the alignment. On the right the series after the alignment ([SE17]).

4.3 Unsupervised learning techniques

Unsupervised techniques are all those methods that receive as input a dataset whose samples are not identified by a label. The aim is to divide data points into a number k of clusters, to better satisfy an optimization condition/problem. The literature contains various examples of unsupervised clustering algorithms, but these can be grouped into three categories ([SE17]): hierarchical, partitional, and fuzzy (that technically are part of the partitionals).

4.3.1 Hierarchical clustering

Hierarchical clustering methods take as input data and a measure of similarity that is used to evaluate the distance between series points [HTF09]. It produces a hierarchical representation of the data where, at the lowest level, there are the single data points, while, going up in the hierarchy, the clusters join the closest groups recursively. There are two approaches, agglomerative (bottom-up) and divisive (top-down). The first merges groups starting from the single points using the minor inter-group dissimilarity as a criterion for the union. The second starts from a single cluster and then splits it into all its components.
The most common graphical representation of this clustering is the dendrogram.

This is a tree that shows each division, generating from each mother node. The height of each node depends on the inter-group dissimilarity between its two daughter nodes. As mentioned earlier, a dendrogram is created regardless of a fixed number of k clusters, and therefore, to understand what is the optimal number of clusters, one needs to study the largest change in dissimilarity between nodes. See example in [Figure 4.4]. Alternatively, one can fix an optimal number and stop the algorithm of division or union as soon as this threshold is reached.
The most evident disadvantages of this method are

- Time and memory complexity of $O(N^2)$ (the complete dissimilarity matrix is needed).
- The algorithm imposes a hierarchical structure even if data are not coherent with it.

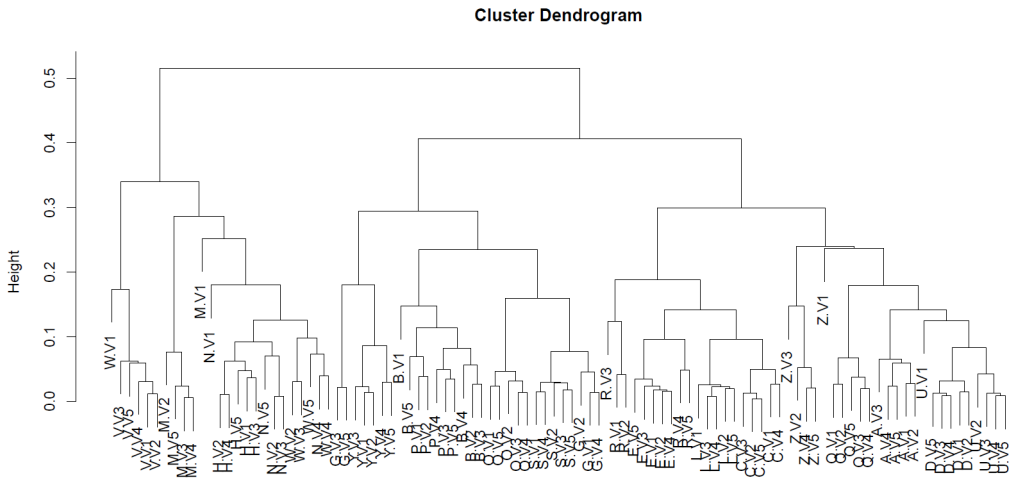

Figure 4.4. Dendrogram created with a random sample contained in the package [SESL17]

4.3.2 Partitional clustering

Partitional clustering uses a different strategy to create partitions. Initially, a number k is fixed, indicating the number of desirable clusters. Then, entries are assigned to only one cluster out of k created. Then, a combinatorial optimization problem dynamically changes the centroids to get the final result. The latter minimizes the intra-cluster distance and at the same time maximizes the inter-cluster dissimilarity. In a few words, the whole process is an optimization problem that takes advantage of iterative greedy descent strategies. However, it is likely to arrive at a local rather than a global minimum.

The steps of the algorithm are as follows. It randomly defines the starting k centroids. The distance between the data and the centroids is calculated and each element is assigned on this basis to the cluster with the nearest centroid. At this point, the centroids are changed to minimize the overall cost function. The procedure is then repeated until a function $\mathscr{F}$ is optimized. The latter is the mathematical translation of a given splitting condition that must be met to obtain a proper division. In other words, given k clusters $\{C_1, \ldots, C_k\}$ and N elements $X = \{x_1, \ldots, x_N\}$, the function

$$\mathscr{F} : \mathbb{P}_k(\Omega) \to \mathbb{R},$$

where $\mathbb{P}_k(X)$ are all the possible partitions of the dataset, must be optimized by obtaining k non empty clusters.

4.3.3 Fuzzy clustering

The two previous clustering methods are methods that produce a hard partition. This means that at the end of the division a point belongs either to a cluster or to another. The clusters are therefore mutually exclusive. The fuzzy clustering methods, on the contrary, associate each point to a cluster with a certain degree. The degree of belongingness is assigned to each point for each cluster, resulting in a matrix of belongingness $N \times k$, where N are the points and k are the considered clusters. All the rows must sum to 1. The most used version of this algorithm is the one proposed by [Pei83], where a fuzzy c-means version is described. The algorithm solves the optimization problem described in the [Equation 4.12].

$$\min \sum_{p=1}^{N} \sum_{c=1}^{k} u_{p,c}^{m} d_{p,c}^{2} \tag{4.12}$$

$$\sum_{c=1}^{k} u_{p,c} = 1, \qquad u_{p,c} \geq 0 \tag{4.13}$$

The u represents the membership matrix and is randomly initialized to meet the constraints. m is the fuzziness exponent and it commonly has a value of 2. The most common distance ($d_{p,c}$) used is the Euclidean one between the p-th object and the c-th fuzzy centroid.

4.4 State-of-the-art clustering algorithms

In this section, some well-known clustering methods are presented. Both time-series oriented and generic methods are explained. The first two, Clara and K-means, were used in the manual feature extraction section while the k-shape was used in the single SPN oriented technique.

4.4.1 K-means

The first method that is described is the most well-known clustering partitional algorithm, the k-means.
Let $X = \{x_1, \dots, x_N\}$ be the data points, $k \in [2, \dots N]$ the number of clusters one intends cluster data and $V = \{v_1, \dots, v_k\}$ the centers of the clusters $\{C_1, \dots, C_k\}$. The goal of the algorithm is to minimized the l_2 distance between each element and its centroid. In fact, the clustering criterion can be expressed as

$$\mathscr{F}(\{C_1, \dots, C_k\}) = \sum_{i=1}^{k} \sum_{j=1}^{k_i} \|x_{ij} - \bar{x}_i\|_2 .$$

In the previous definition, k_i is the number of data in the cluster i, x_{ij} is the j-th observation of the i-th cluster. The $\bar{x}_i$ is the barycenter of the i-th cluster and it represents the centroid. It is evaluated as

$$\bar{x}_i = \frac{1}{k_i} \sum_{j=1}^{k_i} x_{ij} \quad \forall\, i = 1, \dots k.$$

The algorithmic steps follow (see graphic example [Figure4.5]).
In the first step, the centers are established randomly. Then the distance between each data point and cluster centers is calculated. According to the previous distance evaluations, each data point is assigned to the cluster whose distance from the cluster center is minimum compared to the other centers. The new centroids are evaluated $\bar{x}_i$. Distance between each data point and new clusters is evaluated. If there are no changes with respect to the previous step the algorithm stops. Otherwise, another cycle of the algorithm is executed.

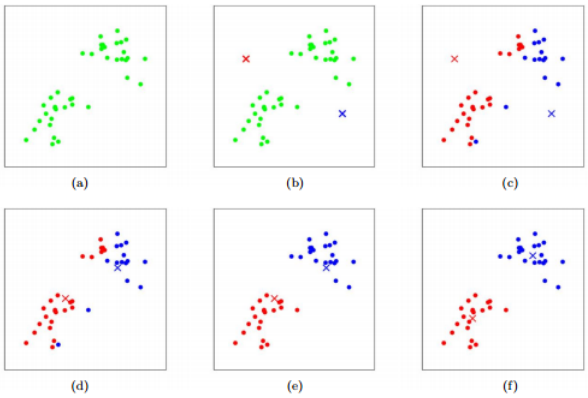

Figure 4.5. This is an example of the k-means algorithm. Points represent data, crosses represent centroids and colors represent clusters. Each image represent a step in the k-means execution.
Site: `https://stanford.edu/~cpiech/cs221/handouts/kmeans.html`

4.4.2 Clara

This method is described in the article [KR90] and is an extension of the well-known partitional algorithm k-medoid. It only uses the sampling approach to simplify calculations in a large dataset setting. In this section, the k-medoid method is described since Clara is easily derived.
The algorithm has a structure similar to the k-means and in fact, in this section, we will show the fundamental differences. The main difference between k-means and k-medoids lies in the definition of centroids and distances. In fact, in the first case, the centroids are the barycenter of the points of the cluster, while in the second, they are points placed in a particularly central position of the cluster. In a few words, while the centroid is a point of space, the medoid (centroid of k-medoid) is a point among data. A positive side of this aspect is that it makes the algorithm less sensitive to outliers. In fact, if a new outlier point is present in a cluster, the centroid (barycenter) will be strongly influenced while the medoid could remain at the same point as before. Moreover, the distance used is often the Manhattan one, as opposed to the k-means case where Euclidean is preferred. However, in both algorithms, it is not excluded from the use of other dissimilarity measures. The algorithm passages are reported.
Let's consider a setup similar to the previous case. Let $X = \{x_1, \dots, x_N\}$ be the data points, $k \in [2, \dots N]$ the number of clusters one intends to divide data and $V = \{v_1, \dots, v_k\}$ the medoids of the clusters $\{C_1, \dots, C_k\}$.

- Randomly select k points that will be the k medoids at time 0.
- Calculate the distances through the dissimilarity measure between the points and the medoids and assign the points to the nearest medoid.
- For each element of a cluster, the following operation is performed. The medoid and simple data points roles are swapped. The intra-cluster dissimilarity between the new medoid and the rest of the cluster points is calculated. The new medoid will be the one with the smallest intra-cluster measurement with respect to the others.
- All points are reallocated according to the new definition of medoids. If there is no difference in the cluster arrangement, the algorithm stops, otherwise, it starts again from step two.

4.4.3 K-shape

The starting point of this algorithm is different from the previous ones. In fact, it aims to divide a set of time series into clusters (see article [PG15]).
This is a partitional algorithm that operates in a similar way to k-means but it is

applied to the time-series context. The reason why it has managed to become one of the most exploited algorithms for time-series clustering is for its excellent ability to obtain well-separated and homogeneous clusters and it scales linearly with the number of time-series windows. It uses distance measurement and a method of creating a representative series based on the shapes of the series. Both the distance and the prototyping function have been anticipated in [Subsection 4.2.3] and in [Subsection 4.1.3].
Given as input a set of time-series X and a number k of desirable clusters, the algorithm steps are the following.

- The time series are assigned to k random clusters.
- For each cluster, the centroid is calculated using the method explained in [Subsection 4.2.3].
- Each time series in X is assigned to the cluster relative to the nearest centroid by using as a distance the one described in [Subsection 4.1.3].
- The procedure is iterated until convergence.

4.5 Internal Cluster Validity Indices (CVI)

All the experiments reported in this thesis are unsupervised and therefore, it is not possible to find direct proof of the accuracy of cluster division. However, it is possible to take advantage of some parameters of goodness that give information about the quality of the clusters without considering external information. For that reason, all parameters considered are called *internal*. These consider connectivity, compactness, and separation of partitions. In this way, one can objectively compare clustering divisions. The parameters used in this thesis are implemented into the two packages [BPD+11] and [SESL17] and they will be discussed below.
In the following explanations, the number of clusters will always be K and clusters will be indicated as $C = \{C_1, \ldots, C_K\}$. *dist* indicates a generic distance and the term *average point* could vary depending on the subject of the clustering.

Connectivity

Connectivity is explained in the article [HKK05] and it is an indicator of connectedness between observations in the same cluster.
Let's indicate with N the number of observations in the dataset and with M the number of attributes that each observation has. We define also $nn_{i(j)}$ the j-th nearest neighbor of observation i. Last element to be described is $x_{i,nn_{i(j)}}$, which

is zero if i and j are in the same cluster while it is $1/j$ otherwise. The formula is

$$conn(C) = \sum_{i=1}^{N} \sum_{j=1}^{L} x_{i,nn_{i(j)}},$$

where L is the number of nearest neighbors to consider. The connectivity can assume values from 0 to infinity and should be minimized.

Silhouette Width

This indicator, as well as the following one, considers the intra-cluster variance and also the distance between clusters in one indicator. It is fully explained in the article [Rou87].
Let's define the elements that characterize the Silhouette Width formula. Let $n(C_p)$ be the cardinality of the p-th cluster and $C(i)$ the cluster containing the i observation. Then

$$b_i = \min_{C_p \in C \setminus C(i)} \sum_{j \in C_p} \frac{dist(i,j)}{n(C_p)}$$

is the average distance between observation i and the nearest cluster. Let's define a_i as the avarage distance between observation i and all other observations in the same cluster. The silhouette related to observation i is defined as

$$S(i) = \frac{b_i - a_i}{\max(b_i, a_i)},$$

while the Silhouette Width is the average of Silhouette values. It ranges between -1 and 1 and it should be maximized.

Dunn Index

This index summarizes the compactness and separation (opposing trends) of clusters in a unique indicator. It was described for the first time in [Dun74].
Let's define $diam(C_m)$ as the maximum distance between observations in C_m, so the Dunn index is defined as follows.

$$Dunn(C) = \frac{\min\limits_{C_k, C_l \in C, C_k \neq C_l} \left(\min\limits_{i \in C_k, j \in C_l} dist(i,j) \right)}{\max\limits_{C_m \in C} diam(C_m)}$$

Average Proportion of Non-overlap (APN)

This measure [DD03] highlights the stability of the clusters. In fact they involve removing every attribute at each repetition. In particular, this index measures

how the amount of observations placed in a given cluster varies, by eliminating the l-th column from time to time.
Let $C^{i,0}$ be the cluster of observation i using all the available features, while $C^{i,l}$ is the one obtained by exploiting all attributes but the l-th.

$$APN(C) = \frac{1}{MN}\sum_{i=1}^{N}\sum_{l=1}^{M}\left(1 - \frac{n(C^{i,l} \cap C^{i,0})}{n(C^{i,0})}\right)$$

The interval of this measure is between 0 which correspond to highly consistent results and 1.

Average Distance (AD)

This measure evaluates the avarage distance between observations contained in the same cluster evaluated in different ways. In a first step, with all feature involved and afterwards, without the l-th column.

$$AD(C) = \frac{1}{MN}\sum_{i=1}^{N}\sum_{l=1}^{M}\frac{1}{n(C^{i,0})n(C^{i,l})}\left[\sum_{i\in C^{i,0}, j\in C^{i,l}} dist(i,j)\right]$$

It assumes values between 0 and infinite and it should be minimized.

Average Distance between Means (ADM)

This index measures the distance between the cluster centers evaluated with the clustering algorithm before and after removing the l-feature. Let's define $\bar{x}_{C^{i,0}}$ the average point of all observations contained in the cluster which includes the i-th observation, considering all features. $\bar{x}_{C^{i,l}}$ is defined as before but excluding the l-feature in the clustering evaluation.

$$ADM(C) = \frac{1}{MN}\sum_{i=1}^{N}\sum_{l=1}^{M} dist(\bar{x}_{C^{i,0}}, \bar{x}_{C^{i,l}})$$

It takes values between 0 and infinite and it is preferred small.

Figure Of Merit (FOM)

This index evaluates the average intra-cluster variance of a deleted column by considering the clustering based on the remaining columns. This method is fully explained in [YHR01]. The mean error is evaluated as

$$FOM(l,C) = \sqrt{\frac{1}{N}\sum_{k=1}^{K}\sum_{i\in C_k(l)} dist(x_{i,l}, \bar{x}_{C_k(l)})},$$

where $x_{i,l}$ is the l-th feature of the i-th observation in cluster $C_k(l)$ while $\bar{x}_{C_k(l)}$ is the average of the same cluster. It takes values from 0 and infinity and small value is index of better performance.

COP index

This index, as the next three, is fully described in [AGM+13]. It describes the cohesion of the clusters by evaluating the ratio between the distance of all the cluster points with its centroid and the furthest neighbor distance. This index should be minimized.

$$COP(C) = \frac{1}{N} \sum_{C_k \in C} n(C_k) \frac{\frac{1}{n(C_k)} \sum\limits_{x_i \in C_k} dist(x_i, \bar{x}_{C_k})}{\min\limits_{x_i \notin C_k} \max\limits_{x_j \in C_k} dist(x_i, x_j)}.$$

Where the fraction in the summation represents intra/inter and $\bar{x}_{C_k}$ is the average point of all elements in cluster C_k.

Davies-Bouldin index (DB) and Modified Davies-Bouldin index (DB*)

This is one of the most used indexes in literature and it estimates cohesion and separation in the following way. The first thanks to the distance between the points and its centroids. The second is based on the distance between centroids.

$$DB(C) = \frac{1}{K} \sum_{C_k \in C} \max_{C_l \in C \setminus C_k} \left(\frac{S(C_k) + S(C_l)}{dist(\bar{x}_{C_k}, \bar{x}_{C_l})} \right),$$

where $S(C_k) = \dfrac{1}{n(C_k)} \sum\limits_{x_i \in C_k} dist(x_i, \bar{x}_{C_k})$.
In its variation the denominator has been changed into the minimum distances between centers.

$$DB^*(C) = \frac{1}{K} \sum_{C_k \in C} \frac{\max\limits_{C_l \in C \setminus C_k} (S(C_k) + S(C_l))}{\min\limits_{C_l \in C \setminus C_k} (dist(\bar{x}_{C_k}, \bar{x}_{C_l}))}$$

Both indices should be minimized.

Calinski-Harabasz index (CH)

This index is a ratio type index, where the two terms represent cohesion and separation. The first is calculated through the distance between points and corresponding centroids, while the second is the distance between centroids and the

global centroid.

$$CH(C) = \frac{N-K}{K-1} \frac{\sum\limits_{C_k \in C} n(C_k) dist(\bar{x}_{C_k}, \bar{X})}{\sum\limits_{C_k \in C} \sum\limits_{x_i \in C_k} dist(x_i, \bar{x}_{C_k})}$$

where $\bar{X}$ is the average of all observations. Large values are preferred.

Score Function (SF)

This last index is a summation-type one. Separation is evaluated by taking the distance between cluster centroids and global centroid, while cohesion takes the distance from each point and its centroid. The definition is the following.

$$SF(C) = 1 - \frac{1}{e^{e^{bcd(C)+wcd(C)}}}.$$

Now the two components of the formula are described.

$$bcd(C) = \frac{\sum\limits_{C_k \in C} n(C_k) dist(\bar{x}_{C_k}, \bar{X})}{NK},$$

$$wcd(C) = \sum_{C_k \in C} \frac{1}{n(C_k)} \sum_{x_i \in C_k} dist(x_i, \bar{x}_{C_k}).$$

This index should be maximized.

4.6 Ranking algorithm

In the next chapter, for each category of clustering algorithms, different K (number of clustering to split the dataset) and aggregation methods will be compared. When analyzing clusters through the goodness indicators reported in [Section 4.5], the results very often do not provide a definite winner with respect to all indices. This was the reason why a ranking algorithm has been employed. This algorithm is based on the results obtained on the individual indices. In fact, for each test, a ranking that takes into account how well the algorithm performs with certain characteristics is provided. The algorithm involved is already present in R and it is fully explained in the article [PDD09]. The distance involved to compare rankings is described in the first part of the section, while the proper algorithm will be explained afterward.
As the author of the article proposes, the ranking problem will be seen as an optimization problem where the goal is to find a "super-list" as close as possible to all the others. Given a set of m lists $L = \{L_1, \ldots, L_m\}$, the objective function can be written as

$$\Phi(\delta) = \sum_{i=1}^{m} w_i dist(\delta, L_i),$$

where δ is the proposed list of length $k = L_i$, while w_i is the importance of every list L_i. The optimization problem becomes

$$\delta^* = \text{argmin} \sum_{i=1}^{m} w_i dist(\delta, L_i),$$

so that the resulting δ^* minimizes the total distance with all other lists.

4.6.1 Spearman footrule distance

Let's define some actors involved in the following definitions. Let $M_i(1), \ldots, M_i(k)$ be the ordered scores associated to each list L_i, where $M_i(1)$ is the best score. Let $r^{L_i}(A)$ be the ranking of the item A in the list L_i. If A is not within the fists k items, the constant value $k + 1$ will be associated to $r^{L_i}(A)$. The ranking in this phase is given according the definition of "best" and "worst" for each case, i.e. the first position is given to A, if A assumes the best values as possible according to the ranking criterion. $r^{\delta}(A)$ is defined in the same way. The Spearman footrule distance is defined as

$$S(\delta, L_i) = \sum_{t \in L_i \cap \delta} |r^{\delta}(t) - r^{L_i}(t)|.$$

This distance is very simple and only compares the position of the elements in the lists, but does not take into account the differences based on "how far" two lists

are. This is why very often the weighted version of the previous distance is taken into account. Weighted Spearman's footrule distance, which more objectively defines the distance between lists, is defined as follows for the comparison between a generic L_i list and the δ list.

$$WS(\delta, L_i) = \sum_{t \in L_i \cap \delta} |M(r^{\delta}(t)) - M(r^{L_i}(t))| \times |r^{\delta}(t) - r^{L_i}(t)|.$$

In this way, if two items are very far apart, the weight that will be given is very relevant for the final result and vice versa.

4.6.2 Cross-Entropy Monte Carlo algorithm

The following algorithm is the core of the method. This is based on the considerations made and on the measure previously introduced. However, to continue, it is appropriate to define in a new way the ranking concepts.
Let's consider a matrix $(X)_{n \times k}$, and let's associate it to an ordered list of n elements through k positions. In this way, rows can sum at most to one, while columns must sum up to one. This matrix represents the position of the n-th element when a "1" appears in the k-th column.
Now, the solution space $\mathscr{X}$ is defined by all the possible X matrix and the objective function must be optimized in such space. Let's assume that X is a random variable, and let's define the probability mass function of a generic matrix x as $\mathbb{P}(x)$. The index of this new matrix will be indicated as $(v)_{n \times k} = ((p_{jr}))$. The conditions of the join distribution $\mathbb{P}(X = x)$ satisfy the following condition.

$$\mathbb{P}_v(x) \propto \prod_{j=1}^{n} \prod_{r=1}^{k} (p_{jr})^{x_{jr}} \times I\left(\sum_{r=1}^{k} x_{jr} \leq 1,\ 1 \leq j \leq n;\ \sum_{j=1}^{n} x_{jr} = 1,\ 1 \leq r \leq k \right)$$

The algorithm follows the next four steps.

- **Initialization:** At $t = 0$ the random distribution is initialized to a constant value $p_{jr}^0 = 1/n$, where the exponent indicates the instant of time. At this step every item could be included into the final ranking with same probability.
- **Sampling:** A sample of size N is drawn from $\mathbb{P}_{v^t}(x)$. The top-k lists δ_i's and the values of the objective function $\Phi(\delta_i)$ are evaluated at each time t. Now, let's define the integer part of a real number e as $[e]$. The $\Phi(\delta_i)$'s are sorted in ascending order and the ρ-quantile is evaluated ($y^t = \Phi_{([\rho N])}$).

- **Updating:** Parameters of the distribution are updated as

$$p_{jr}^{(t+1)} = (1-w)p_{jr}^{(t)} + w\frac{\sum_{i=1}^{N} I(\Phi(\delta_i) \leq y^t)x_{ijr}}{\sum_{i=1}^{N} I(\Phi(\delta_i) \leq y^t)},$$

  where w is a parameter introduced to avoid local maxima and x_{ijr} is the value at the jr-th position in i-th sample.

- **Convergence:** If the previous step does not modify the optimal list, the algorithm stops and the optimal value of Φ is found.

Chapter 5

Clustering techniques applied to off-road vehicle time series

The theory analyzed in the previous chapters will be exploited in this one to obtain clusters that will group univariate or multivariate series.
Although a different approach is reported in the various sections, some common concepts can be identified. First of all, to understand how much two time series are similar, measures that deal with shape-based characteristics will be involved (see [Section 4.1]). On the other hand, to illustrate a cluster of elements, a representative that summarizes the characteristics of the series will be drawn. It can be done exploiting the prototyping techniques illustrated in [Section 4.2].
In conclusion, clustering algorithms exploited in this section can be described through the following three parameters.

- **Type.** There are different methods of clustering but the most common are fuzzy, partitioning, or hierarchical ones.
- **Distance.** To measure the distance between elements and understand which is the relative positioning, a similarity measure is needed. In this section, shape-based and DTW similarity measures are considered.
- **Centroid evaluation method.** The clustering algorithms are based on the maximization of the distance between clusters whose representative is a centroid series. So, it is crucial to appropriately identify the representative of each cluster.

The objective of this chapter is to apply different methods to understand how they work in the proposed scenario, i.e. three series (SPNs) aligned with respect

to a unique clock. In this way, it will be possible to understand which type of aggregation or downsampling works better for each model. Later, in the concluding chapter, the results obtained in this section will be analyzed.

Two approaches will be addressed in this chapter. The first involves a two-step process: individual time-series are clustered according to their own properties and then an aggregation phase follows, where an optimal method to merge previous results. This will cover the first section.
The second section will involve methods that consider all three series at the same time.

5.1 Combination of univariate results

The first approach that has been tested is based on the identification of clustering within the individual SPN signals and then, combining the results, a general clustering is obtained. The process that follows involves, as in almost all cases from now on, all the eight datasets that have been determined by aligning the time series at different granularities.
The three models reported in [Table 5.1] have been exploited to compare the different aggregation methods.

Name	type	distance	centroid evaluation	norm
k-shape	partitional	SBD	“shape” method	1
Pam-based	partitional	DTW	pam	1
Dba-based	partitional	DTW	dba	2

Table 5.1. Features of the clustering algorithms that have been applied. The "Name" column is only reported for the sake of the reader’s understanding.

For each SPN-signal, algorithms presented in [Table 5.1] are applied to the eight datasets to obtain which one performs better. To evaluate the division performances, internal goodness indices (CVIs) are used. Among those mentioned in the previous chapter, those reported in [Table 5.2] have been selected.

Before conducting the experiments, one could expect that an optimal combination of (algorithm, aggregation, granularity) could emerge independently of the SPN considered.
However, a unique general solution did not emerge only based on the results of the CVI.
There are two main reasons.

CVI	Optimal
Silhouette index	to be maximized
Dunn index	to be maximized
COP index	to be minimized
Davies-Bouldin index	to be minimized
Modified Davies-Bouldin index	to be minimized
Calinski-Harabasz index	to be maximized
Score Function	to be maximized

Table 5.2. Cluster Validity Indices (CVIs) in the one-SPN clustering analysis.

Depending on the shape of the single SPN signal, the type of aggregation/down-sampling acts differently. For example, a method that has flattening problems with an already smooth SPN can be optimal for an irregular one. Moreover, depending on the complexity of the shapes of each signal, the clustering algorithm can have more or less difficulty in finding the representative elements of each cluster.

The CVIs output of a single dataset that passes through one clustering algorithm is a vector of seven values as reported in [Table 5.2]. However, for each SPN, eight datasets are compared and for each of them, three different clustering algorithms are used. So, the result is very difficult to interpret, especially because not all CVIs agree on a unique winning solution/method. For this reason, a method to get a more stable ranking was needed. The ranking algorithm that is explained in [Section 4.6] solves this problem.
After calculating the CVIs of all analyzed cases, they were merged into a matrix. The latter was used as the weight matrix, useful to calculate the Spearman footrule distances between rankings. This method allows therefore to establish which couple (algorithm, dataset) is better than others. This is possible without the constraint that the couple is the best according to all the indexes.

[Table 5.3] shows the best result obtained in the optimal list for each considered SPN.

SPN	number of clusters	dataset	method
190	2	Fourier, 0.5Hz	K-shape
524	2	Fourier, 0.5Hz	Dba-based
94	3	Fourier, 2Hz	Pam-based

Table 5.3. SPN optimal choice in the one-signal based analysis.

Since the analysis is unsupervised, it was necessary to use heuristic methods to validate the goodness of the clusters. The following graphical analyses have been inspected together with domain experts which have been validated what has been inferred. Before going any further, however, a small clarification on the use of colors seemed necessary. The time-series have been drawn with the addition of vertical lines indicating the cluster to which the left window belongs. Moreover, a one-to-one correspondence has been set between the cluster number and a specific color (see [Table 5.4]).

Number	0	1	2	3	4
Color	blue	red	green	pink	violet
Number	5	6	7	8	9
Color	grey	yellow	light blue	light green	dark green

Table 5.4. In the whole analysis this one-to-one correspondence between cluster number and associated color is used.

Engine speed windows (SPN 190) have been clustered clearly (see [Figure 5.1]). On one hand one can find windows characterized by a strong presence of high frequencies and absence of low ones. They are grouped in cluster 1. This means that they experience micro oscillations at constant levels. On the other hand (cluster 0), windows present the opposite situation, with the presence of strong variations in engine speed measurement [Figure 5.1]. In the same figure, the averages of the frequency responses are represented. Here, it is even more evident the difference between the two clusters highlighted before. Finally, a time-domain representation in a limited time slot is shown. Here all hypotheses are confirmed again.

Very similar behavior is shown in the figure which displays the results of the transmission gear (SPN 524). In cluster 0, shaken series are grouped, while flat windows are set as cluster 1. However, as one can see by the example reported in the second plot in [Figure 5.2], cluster 1 not only contains straight lines but also segments with slight oscillations. In fact, in the indicated plot, a not exactly flat green line is shown.
Since this signal has only five different measurements (the vehicle has five gears) when the pattern is stable on a single gear, there are no oscillations at all and therefore, the difference between the two clusters is more evident than before. In [Figure 5.2] all main results are shown.

The last univariate analysis involves the fuel pressure signal (SPN 94). Also in this case the cluster assignment mechanism is very similar to the previous ones. However, three behaviors are highlighted. In [Figure 5.3] main results are reported. Cluster 0 and 2 are characterized by fluctuating windows (strong presence of low frequencies) while number 1 is characterized by flat windows. Cluster number 0, however, presents fewer oscillations (also less powerful) than cluster 2 and this is confirmed once again by the graph which shows the windows in the time-domain and frequency-domain, the third and fourth image respectively.

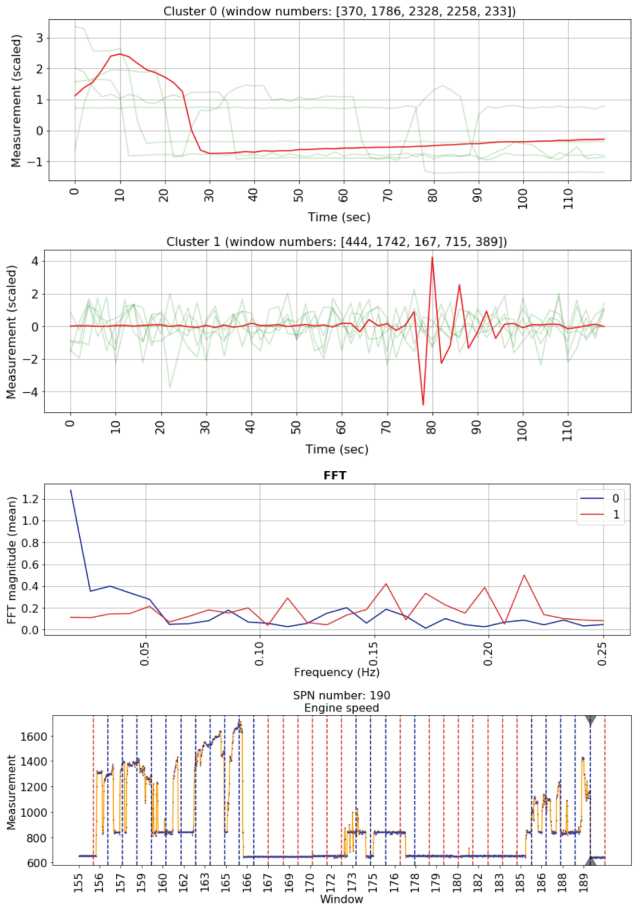

Figure 5.1. Set of plots that summarizes the result obtained from the best clustering (in terms of goodness indexes) for what concern the engine speed signal. In the first two plots, the centroid is drawn in red, while 5 random series belonging to the relative clusters are plotted in green. The third image represents the averages of the magnitude of the Fourier transform of the windows belonging to cluster 0 and 1. The plot at the bottom represents a time-domain plot of the signal which highlights the windows clusters. In this plot, the x-axis represents the window number.

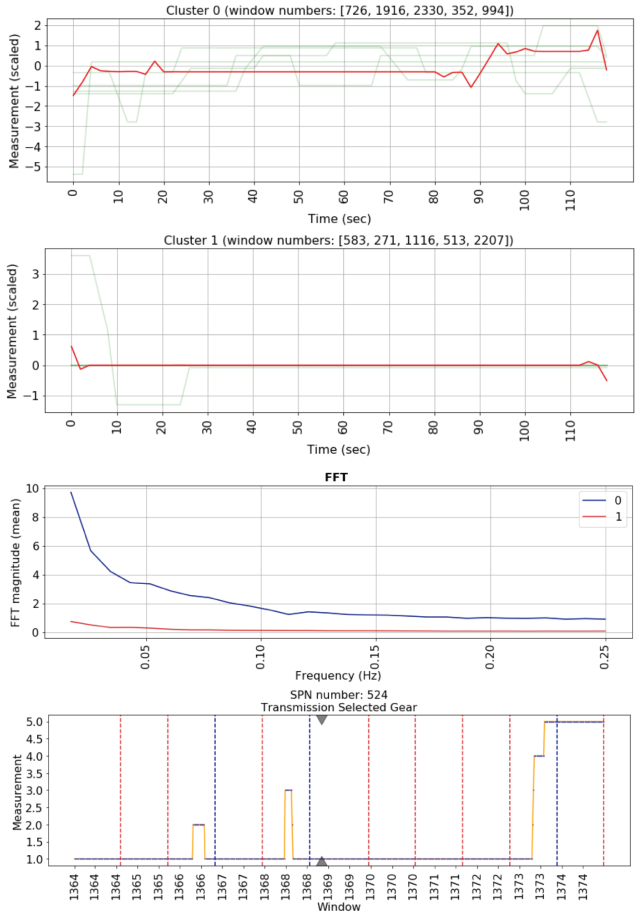

Figure 5.2. Set of plots that summarizes the result obtained from the best clustering algorithm (in terms of goodness indexes) with respect to the SPN 524. The first two plots represent in red the centroid and in green the random series belonging to the relative clusters. The clusters represented in these first two images are respectively 0 and 1. The third image represents the averages of the magnitude of the Fourier transform of the windows. The plot at the bottom represents a time-domain plot of the signal by highlighting the windows clusters. In this plot, the x-axis represents the window number.

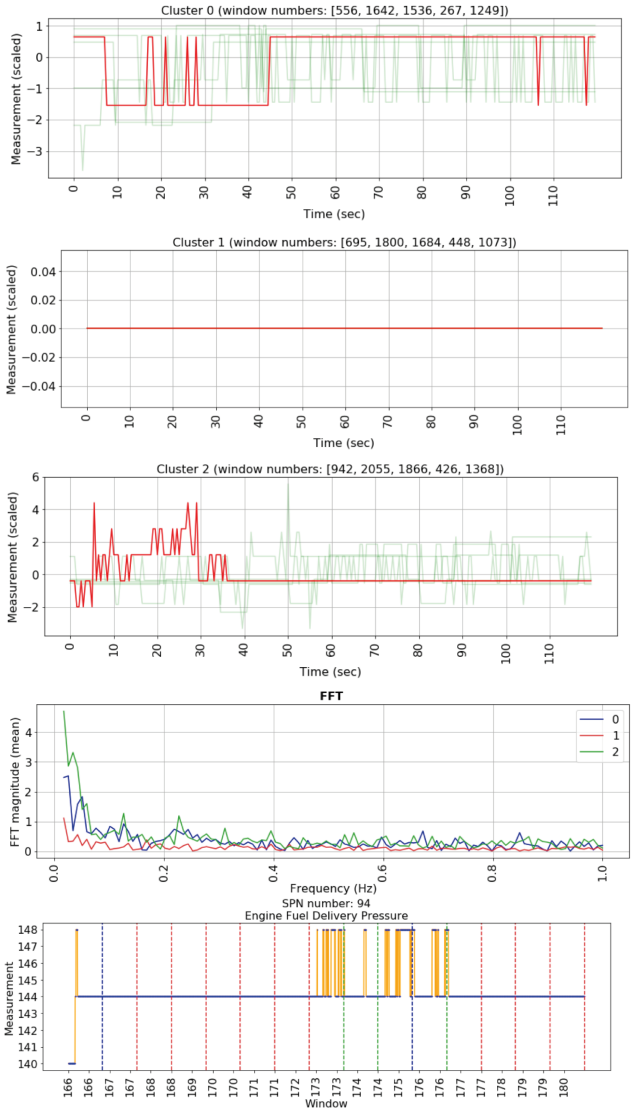

Figure 5.3. Set of plots that summarizes the result obtained from the best clustering algorithm (in terms of goodness indexes) with respect to SPN 94. The first three plots represent in red the centroid and in green the random series belonging to the relative clusters. The clusters represented in these first images are respectively 0, 1 and 2. The third image represents the averages of the magnitude of the Fourier transform of the windows belonging to the different clusters. The plot at the bottom represents a time-domain plot of the signal by highlighting the windows cluster. In this plot, the x-axis represents the window number. Only a few windows are considered in the example.

In the conclusion of this section, some analysis on the similarity of the clusters of the different signals (SPNs) is needed. Although belonging to different engine parts, the SPNs analyzed are all closely interconnected. This led to conducting a more in-depth analysis of the number of windows that are classified in the same way within the three signals. The result obtained is about 57%. As anticipated, this is not unexpected considering the strong connection of the characteristics of a machine. Even if the similar signals (in terms of simple correlation) have been removed, some similarities are present within the signals selected, since the subject involved is a fully connected vehicle.

After having interpreted the single results, the next step is to characterize each window in order to combine the characteristics of the windows of each SPN signal. For example, if a window has been labeled as 1 for all three signals, it will be labeled with the triplet (1,1,1). This means that the vehicle behavior is characterized by the different status of the SPNs considered.
In this way, $2x2x3$ clusters will be created.
The 12 resulting super-clusters are summarized in [Table 5.5].

General Cluster	524 cluster	190 cluster	94 cluster	number points
0	0	0	0	79
1	1	0	0	218
2	0	1	0	1
3	1	1	0	125
4	0	0	1	53
5	1	0	1	252
6	0	1	1	11
7	1	1	1	1258
8	0	0	2	52
9	1	0	2	170
10	0	1	2	1
11	1	1	2	112

Table 5.5. Final result of the single SPN approach.

This division identifies as the biggest cluster the one characterized by flat windows (1,1,1). It contains half of the data. All the other clusters are much smaller because they identify windows with rarer characteristics than the flat series in a vehicle moving in a test field. Other clusters are empty (or almost empty) because they identify combinations of incompatible single clustering characteristics. For

example, the cluster 10 (in the new configuration) indicates all windows characterized by rapid gear changes but flat engine speed. Two incompatible behaviors that never find a match.

5.2 Multivariate clustering application

In this section, the methods exploited in the previous section are proposed in a multivariate context. In the first part, clustering is based on multivariate DTW, while in the second part, a DTW-based aggregate similarity measure is derived to achieve the clustering goal. In the last subsection, a method that exploits the feature extraction has been developed.

5.2.1 DTW-based methods

The first approach is the one used in the previous section but extended to the multivariate time-series context.
In this case, the algorithm based on the shape distance (k-shape) was not exploited, due to the difficult generalization from the univariate to the multivariate case. Also the authors of [PG15] does not report this extension in their work or in the following ones.
In conclusion, the two models involved are summarized in [Table 5.6].

Model	type	distance	centroid evaluation	norm
Pam-based	partitional	DTW	pam	1
Dba-based	partitional	DTW	dba	2

Table 5.6. The table summarizes the features that distinguish the different clustering algorithms applied.

The execution process that has been carried out is the same as in the previous case; the eight datasets were processed through clustering algorithms summarized in [Table 5.6]. Then, the CVIs are evaluated. This time the CVIs involved are the ones reported in [Table 5.7].

After that, all results of CVIs analysis have been processed by the ranking algorithm, which has decreed as the best way to divide data the one reported in the [Table 5.8]. The algorithm has divided the data in a very unbalanced way as we can see from table [Table 5.9].

The results obtained are shown in the same fashion as before. [Figure 5.4] and [Figure 5.5] summarize all results.

CVI	Optimal
Silhouette index	to be maximized
Dunn index	to be maximized
COP index	to be minimized
Davies-Bouldin index	to be minimized
Modified Davies-Bouldin index	to be minimized
Calinski-Harabasz index	to be maximized
Score Function	to be maximized

Table 5.7. CVIs in DTW-based approach.

number of clusters	dataset	method
2	Mean, 0.5Hz	Pam-based

Table 5.8. SPN optimal choice in the DTW-based multivariate analysis.

label of cluster	number of windows
0	114
1	2218

Table 5.9. Number of windows for each cluster in the DTW-based multivariate analysis.

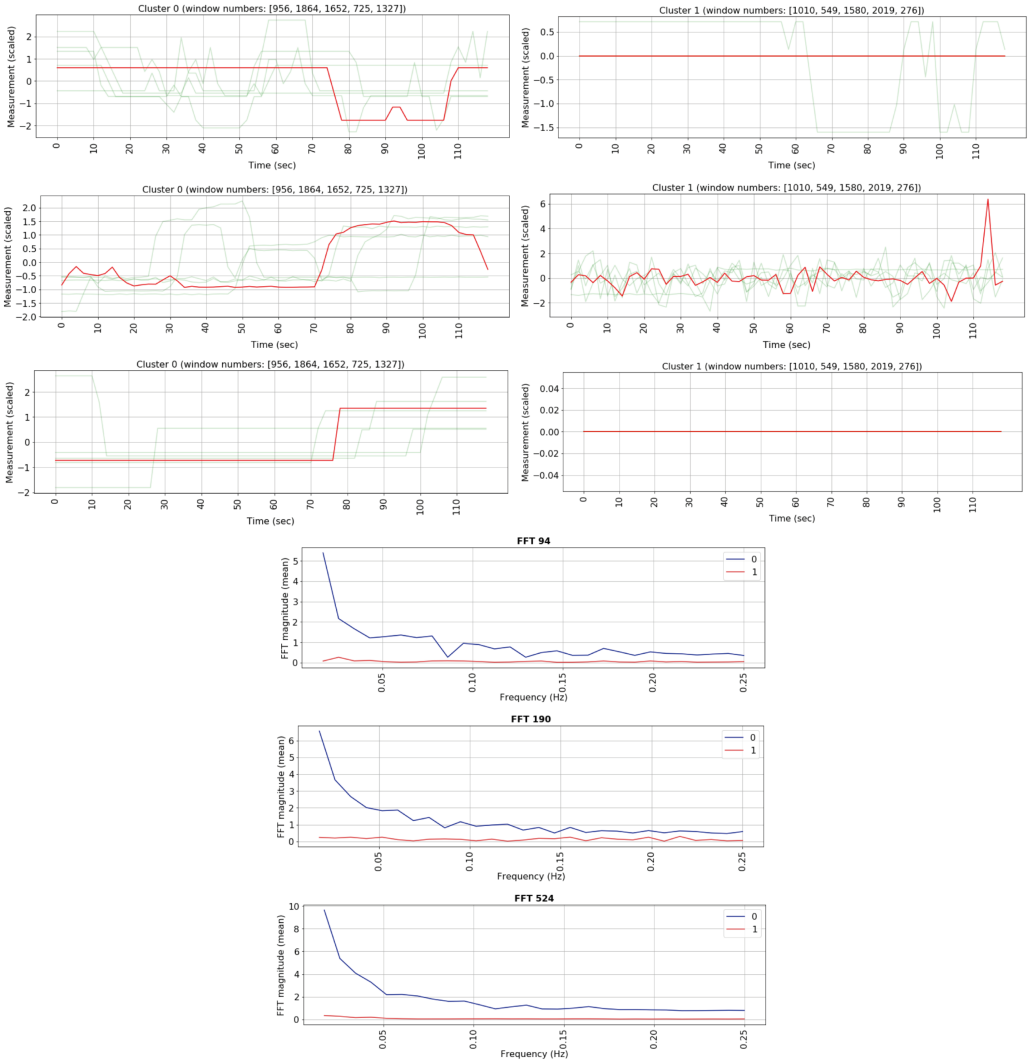

Figure 5.4. Multivariate DTW-based algorithm results. In the first 6 images, each row represents the SPN considered (from the top to the bottom: SPN 94, 190 and 524), while columns represent the cluster number (in order from left to right: cluster 0 and 1). In red the prototype for each group is plotted while, in green, 5 random elements for each cluster are reported. The last 3 images represent the average of FFT magnitude among all windows of the same cluster (in order from the top to the bottom: SPN 94, 190, and 524).

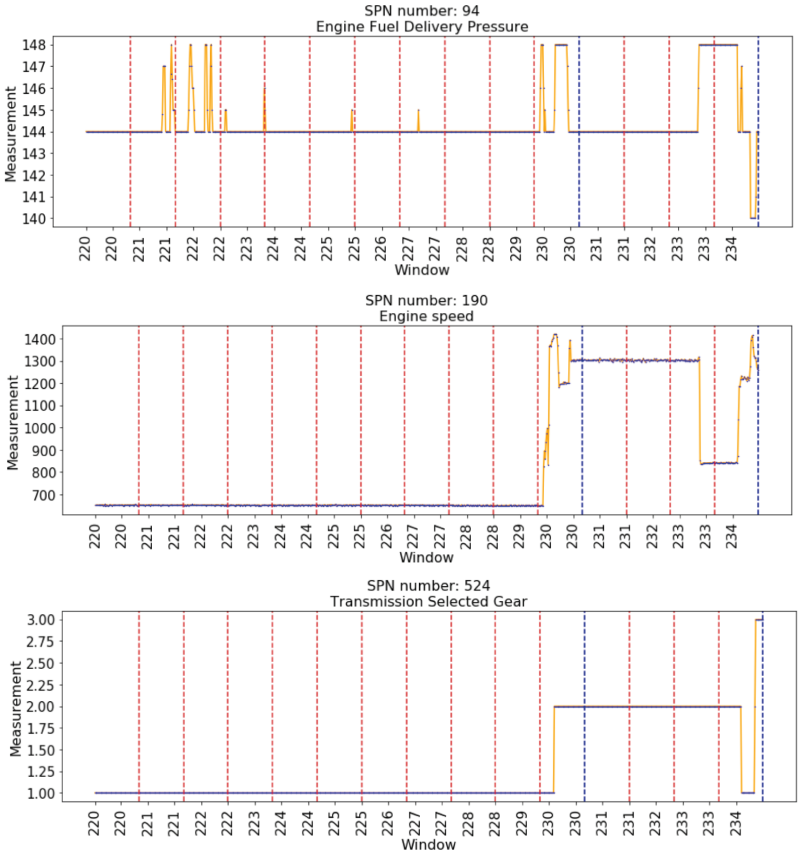

Figure 5.5. Multivariate DTW-based algorithm results. For each SPN, a plot is drawn. The three series are plotted in the time domain ranging in the window-slot $(220 - 234)$. Clusters are represented by the color of the respective label (see [Table 5.4]).

The two clusters that have been formed have a clear characterization for each of the three signals in the dataset. In this case the result is very unbalanced (see [Table 5.9]); there are 114 windows classified as cluster 0 and the rest as cluster 1. Cluster 0 is characterized by strong variations while cluster zero is characterized by smoother shapes. For further comments or comparisons with the previous method, the reader is invited to refer to the conclusions chapter.

5.2.2 Aggregation-based approach for multivariate clustering

In this section, two methods allow to combine the 3 distance matrices obtained through the DTW (one for each SPN). These methods are fully described in [Subsection 4.1.2] and they can be summarized in the M and C operations by considering two multivariate series x and y.

$$M(x, y) = \sum_i DTW(x_i, y_i)$$

and

$$C(x, y) = \max\{DTW(x_i, y_i) : i = 1, .., n\}.$$

The data analysis process starts as in the previous cases. The eight multivariate series, aligned according to the univocal clock, are spitted in the 2332 windows obtained previously. The windows are transformed into null mean and unitary variance multivariate series and then, they move towards the next step. For each dataset the matrix of the distances is created in the two ways described previously, to be used by the clustering algorithm. For this section, only the traditional PAM method is used to evaluate the centroid (see [Table 5.10]).

Model	type	distance	centroid evaluation	norm
"max" model	partitional	$\sum_i DTW(x_i, y_i)$	PAM	1
"sum" model	partitional	$\max\{DTW(x_i, y_i)\}$	PAM	1

Table 5.10. "max" / "sum" models.

The reason why these two aggregations have been derived, deserves an explanation and an in-depth examination. The explanation is reported in the final chapter where the three methods based on the multivariate approach are compared.
After processing the datasets in the clustering algorithms CVIs tables are evaluated. Since the same indexes of the previous case are used, they are not reported here (see [Table 5.7]). From this analysis, the optimal ranking algorithm has established that the optimal solution is the one reported in [Table 5.11] and [Table 5.12].

number of clusters	dataset	distance
2	Fourier, 0.5Hz	"max" meth.

Table 5.11. Optimal choice for the "max" / "sum" method.

Label of cluster	number of windows
0	627
1	1705

Table 5.12. Number of windows for each cluster in the best "max" / "sum" method.

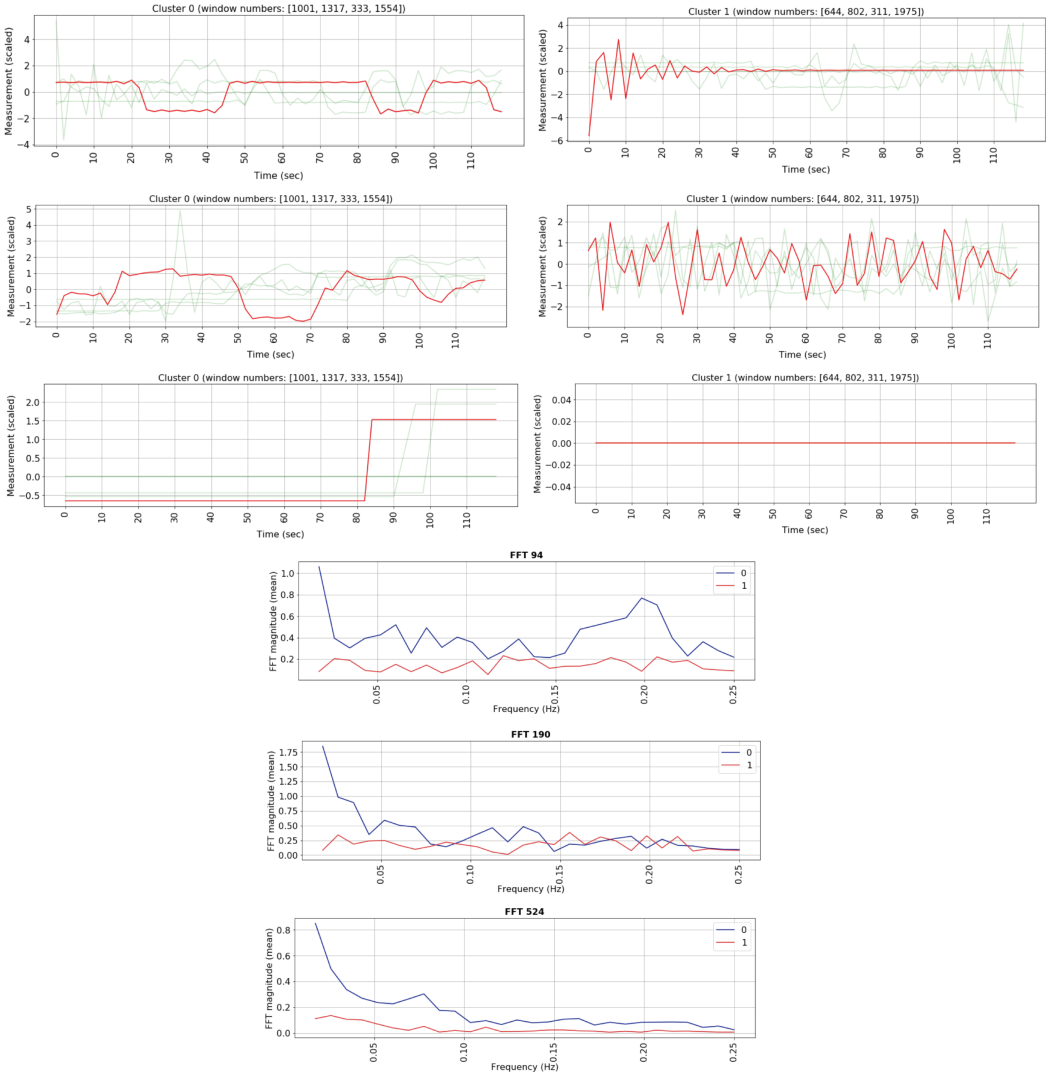

Figure 5.6. Aggregation-based clustering results. In the first 6 images, each row represents the SPN considered (in order from the top to the bottom: SPN 94, 190 and 524), while columns represent the cluster number (in order from left to right: 0 and 1). In red the prototype for each group is plotted, while in green, 5 random elements for each cluster are drawn. The last 3 images represent the average of the FFT magnitude among all windows of the same cluster (in order from the top to the bottom: SPN 94, 190 and 524).

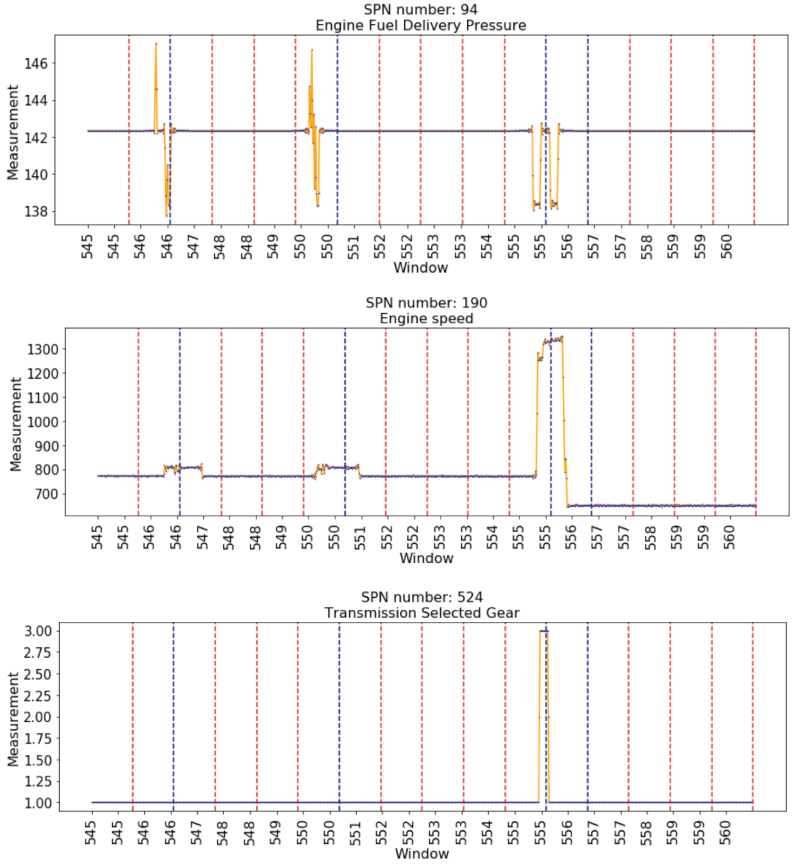

Figure 5.7. Aggregation-base clustering results. For each SPN a plot is drawn. The three series are plotted in the time-domain ranging in the window slot $(545 - 560)$. Clusters are represented by the color of the respective label (see [Table 5.4]).

The results on clustering are very similar to the previous ones, but the proportions changed. In fact, in the previous case about 100 windows were classified with 0 label, while now about 600. From the [Figure 5.6] and [Figure 5.7] the following evaluations on the clusters can be made. In cluster 1, generally flatter series are grouped. On the contrary, cluster 0 has a strong presence of low and medium frequencies. However, in cluster 0, there are not only windows characterized by a strong presence of low frequencies with respect to all SPN. In fact, as one can see by the images, even if a single SPN window is far away from the others (with respect to the DTW distance), that window is labeled as 1. This behavior can be easily seen in the time-domain. For example, one can see windows 546 and 550 where the gear of the machine is constant.

5.2.3 Manual feature extraction for clustering purposes

This subsection deals with clustering through manually extracted features.
This approach is generally applied in the biomedical context where the analysis of signals is mainly based on the frequency response of certain body indicators. These are often used to correlate a series of microscopic pulses to macroscopic behaviors. In other cases, instead, given a priori knowledge about a specific biological frequency, one can understand behaviors or actions much more hidden and less obvious, for example those at the brain level.
Many articles concerning these methods are present in the literature of biomedical engineering. Specifically, this section is inspired by [ZWYG18] for what concerns the applications and [BZ11] for the theoretical part.

First of all, it is necessary to specify how the starting dataset was created. Each SPN signal has been aligned with its own rate (see [Table 5.13]) and then, the resulting signals have been divided into the 2-minute windows previously identified. Each window is identified by a label that identifies it (in each signal) in a unique way. In particular, all window labels have a one-to-one relationship with the labels of the previous sections. The starting time-series have been aligned to their rate to keep the data as authentic as possible (without any loss of information) in order to identify every fluctuation. In conclusion, the dataset exploited to extract the features is composed of 3 vectors of different lengths according to the rates of the various SPNs.

SPN	Rate
94	2 HZ
190	50 HZ
524	10 HZ

Table 5.13. These are the rates of the SPNs involved in the feature based approach.

From the single-SPN dataset, 7 features have been derived for each SPN to build the final dataset. On the other hand, each row of the final dataset will represent one of the 2332 windows.
The first feature evaluated for each window and for each SPN is the average in the time-domain. After that, the window average has been removed from each window to not affect the other features.
Unfortunately, there is no a priori knowledge about specific bands of vehicle signals. So, the spectrum has been divided into equal parts, obtaining three groups for each window in the frequency-domain: high, medium, and low frequencies.
The two functions that have been used to derive the other 6 features are the following: the average of the squared magnitude of the components and the maximum magnitude for each group of frequencies. More formally, let $X_i^k(f)$ be the n coefficients of the Fourier transform relative to the k frequency band, with $k =$ low, medium, high. The dataset has been created by evaluating

$$\frac{1}{n}\sum_{i=1}^{n}|X_i^k(f)|^2$$

and

$$\max\{|X_i^k(f)|^2 \quad \forall i \in \{0,\dots,n\}\}.$$

In the end, a total of seven features have been extracted for each SPN, resulting in a total of 21 features. Remember that the number of total windows is 2332 and therefore the final matrix obtained is $2332x21$.
Afterward, all features were analyzed with respect to the correlation index, thanks to which important considerations have been drawn (see [Figure 5.8]).

In general, all features seem to be very correlated but especially within the individual SPNs, the correlation is very high. A method to extract information without repetitions has been exploited to reduce the number of features. In fact, the traditional algorithmic techniques suffer from the so-called curse of dimensionality, preferring as few features as possible (see [AHK01]). In this case, it is even useless to consider a high dimensionality, since some information is redundant.
However, as it is not a supervised problem, it was not possible to select features

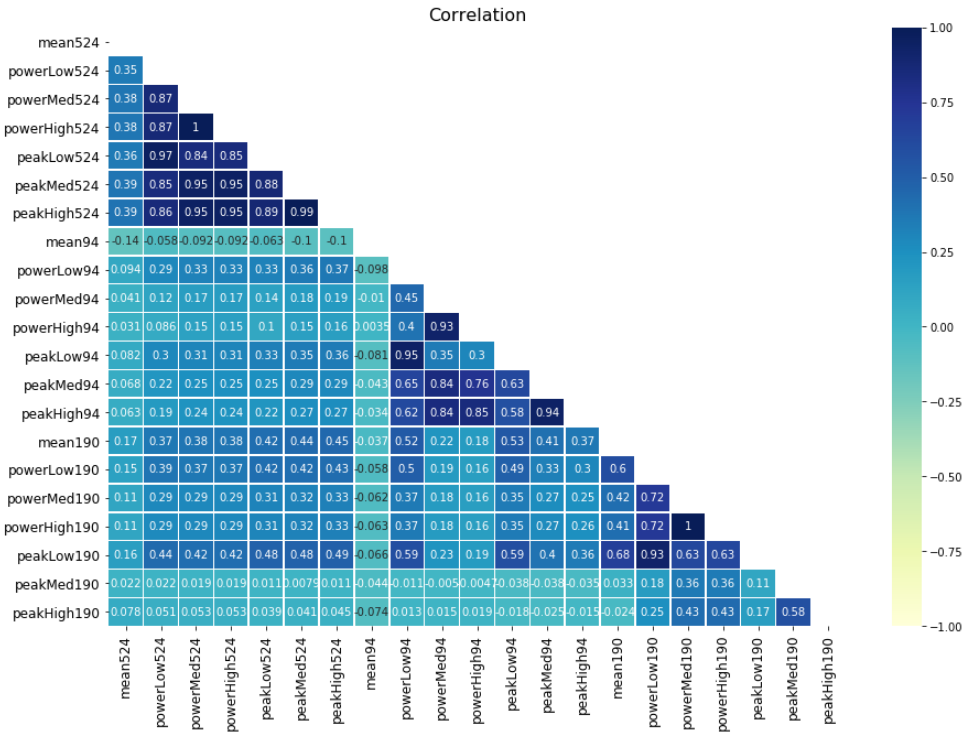

Figure 5.8. Manual feature extraction clustering. General correlation matrix.

according to the relationship with a given response variable.
So, in this case, the matrix was reduced to its main components by applying the PCA method.
This method is an important unsupervised learning technique, which discovers the most informative directions; i.e. the ones along which the dataset varies the most. A short explanation of the mathematical foundation is reported below.

Let's consider $x_i \in \mathbb{R}^n, i = 1, \ldots, m$ data points. Let's consider the baricenter $\bar{x}$ and the matrix with centered data points $\tilde{X} = [\tilde{x_1}, \ldots, \tilde{x_m}]$ where $\tilde{x_i} = x_i - \bar{x}$.
The algorithm allows finding the normalized direction $z \in \mathbb{R}^n, ||z||_2 = 1$ in data space such that the variance of the projection of the centered data points is maximal.
One can indicate the components of the centered data along direction z as

$$\alpha_i = \tilde{x_i}^T z, \qquad i = 1, \ldots, m.$$

According to the previous definitions, the mean-square variation of the data along direction z can be expressed as

$$\frac{1}{m} \sum_{i=1}^{m} \alpha_i^2 = \sum_{i=1}^{m} z^T \tilde{x_i} \tilde{x_i}^T z = z^T \tilde{X} \tilde{X}^T z.$$

The problem now can be formalized as following,

$$\max_{z \in \mathbb{R}^n} z^T \tilde{X} \tilde{X}^T z, \qquad s.t.\ ||z||_2 = 1.$$

To solve the problem is useful to recall the singular value decomposition (SVD), which states that every matrix $A \in \mathbb{R}^{n,m}$ can be factorized as

$$A = U \tilde{\Sigma} V^T,$$

where $V \in \mathbb{R}^{n,n}$ and $U \in \mathbb{R}^{m,m}$ are orthogonal matrices and $\tilde{\Sigma} \in \mathbb{R}^{n,m}$ is a matrix having the first $r = rank(A)$ diagonal entries positive and decreasing in magnitude, and all other entries zero. In this way it is possible to rewrite the problem according to the decomposition of

$$\tilde{X} = U_r \Sigma V_r^T.$$

The direction of the largest data variation is $z = u_1$, the first column of U_r, and the mean-square variation along this direction is proportional to the square of its eigenvalue. Successive principal axes can be found by removing the first principal components, and by applying another time the previous procedure.

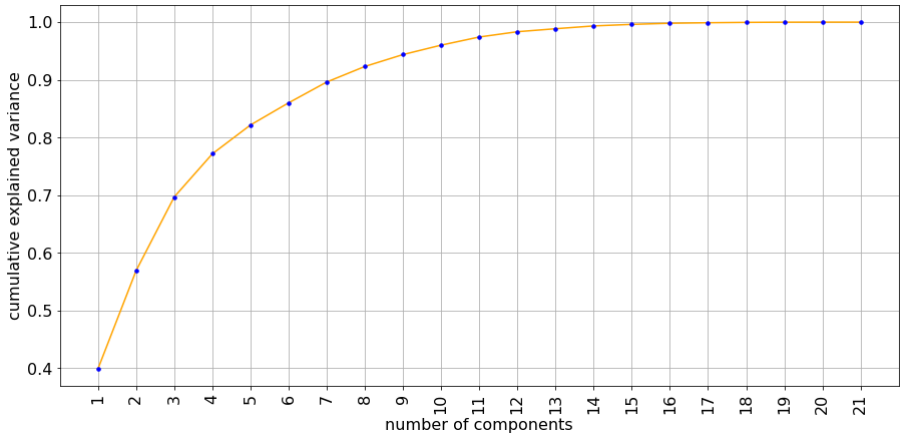

Figure 5.9. Graph of cumulative variance in the case of manual feature extraction clustering.

By exploiting this method, it was possible to capture more than 85% of the variance with 6 principal components (instead of 21) [Figure 5.9].

From here on, the clustering process starts. The data obtained, in fact, have been processed by k-means (see [Subsection 4.4.1]) and Clara (see [Subsection 4.4.2]) algorithms, for a number of clusters between 2 and 10. To understand which method is the most appropriate in this case, goodness indexes have been evaluated, and the choice of the optimal number of clusters was derived. Indices involved in this case are summarized in [Table 5.14].

CVI	Optimal
Silhouette index	to be maximized
Dunn index	to be maximized
Connectivity index	to be minimized
Average proportion of non-overlap (APN)	to be minimized
Average distance (AD)	to be minimized
Average distance between means (ADM)	to be minimized
Figure of merit (FOM)	to be minimized

Table 5.14. CVIs in the case of manual feature extraction clustering.

However, also in this case, indices were not directly comparable as it was not possible to identify a clear winner from them. Index results are reported in [Figure 5.10] and [Figure 5.11].

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
APN	kmeans-3	kmeans-2	kmeans-6	kmeans-5	kmeans-10	kmeans-8	kmeans-9	clara-6	kmeans-7	clara-3	clara-2	clara-4	kmeans-4	clara-5
AD	kmeans-10	clara-10	kmeans-9	clara-8	clara-9	kmeans-8	clara-7	kmeans-7	clara-6	kmeans-6	clara-5	clara-4	kmeans-5	clara-3
ADM	kmeans-10	kmeans-8	kmeans-9	kmeans-7	kmeans-3	clara-3	kmeans-6	clara-2	kmeans-2	clara-6	clara-5	kmeans-4	clara-4	clara-9
FOM	kmeans-10	kmeans-8	kmeans-9	kmeans-7	clara-9	clara-8	clara-10	clara-7	kmeans-6	clara-6	kmeans-5	kmeans-4	clara-5	kmeans-3
Connectivity	kmeans-2	kmeans-3	kmeans-4	clara-2	clara-3	kmeans-5	clara-5	kmeans-6	kmeans-7	clara-4	clara-9	kmeans-9	kmeans-8	clara-6
Dunn	kmeans-2	kmeans-3	kmeans-4	kmeans-10	clara-2	kmeans-9	kmeans-8	kmeans-5	clara-3	kmeans-6	kmeans-7	clara-4	clara-5	clara-6
Silhouette	kmeans-2	kmeans-3	kmeans-4	clara-2	clara-3	kmeans-5	kmeans-6	kmeans-7	kmeans-8	kmeans-9	kmeans-10	clara-8	clara-9	clara-6

Figure 5.10. Result of the clustering algorithm in the manual feature extraction approach. For each line, it is reported the ranking of the cluster algorithms according to the CVI considered.

	1	2	3	4	5	6	7	8	9	10
APN	0.11414433	0.13925226	0.15064016	0.15649180	0.16078733	0.16201796	0.17167669	0.18424601	0.18867452	1.972752e-01
AD	2.98517473	3.01174746	3.05461092	3.08002046	3.08065874	3.14549664	3.24236605	3.25316925	3.30463788	3.443354e+00
ADM	0.65720027	0.65850086	0.66566453	0.73863606	0.74752881	0.85943057	0.94375781	0.99733786	1.05783526	1.157968e+00
FOM	1.27224434	1.28199202	1.29486603	1.32797293	1.36801892	1.37458816	1.37637367	1.38934131	1.45378951	1.455387e+00
Connectivity	12.59603175	116.78650794	125.82142857	139.65833333	247.74444444	292.94484127	296.62341270	326.18809524	331.47658730	3.520333e+02
Dunn	0.08370102	0.03408392	0.03408392	0.01898304	0.01881514	0.01622851	0.01519738	0.01163957	0.01046731	9.170974e-03
Silhouette	0.66677864	0.57392464	0.57383309	0.51141935	0.38115264	0.37943314	0.36789631	0.36712307	0.34000956	3.386198e-01

Figure 5.11. Result of the clustering algorithm in the manual feature extraction case. For each line, there is the value of the cluster algorithm, according to the CVI considered. This table is referred to [Figure 5.10].

For this reason, it was necessary to involve the ranking algorithm [Section 4.6]. At the end of the process, it determines the winning ranking; the k-means algorithm with two clusters was ranked in the first position (see [Table 5.15] and [Figure 5.12].

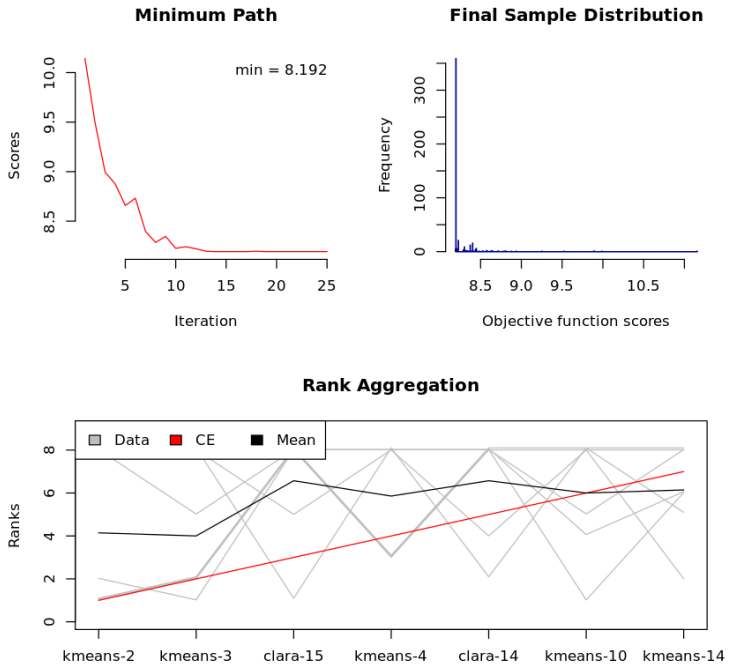

Figure 5.12. Graphical result of the ranking algorithm in the manual feature extraction case.

[Figure 5.13] represents the result by plotting the first three main components, which together explain almost 70% of the variance. To better understand the graph, let's keep in mind the convention of representing each point with the cluster color in [Table 5.4]. In this graph, it is evident that the algorithm has been divided data into two unbalanced parts. The first one is very concentrated on the left side of the plot and its windows have low PCs values (in absolute value). Instead, the second scattered group (the minority) is characterized by higher values. Some quantitative features of the two clusters are reported in [Table 5.15].

To get a more complete overview, however, it is necessary to go into the frequency-domain. Cluster 1 is composed of series that on average have more visible oscillations and have a strong presence of middle and low frequencies. The opposite happens with cluster 0 ([Figure 5.14]). This behavior is well visible even in the time-domain, where the reader can observe that the windows classified with the label 1 have many more variations compared to the other case([Figure 5.15]).

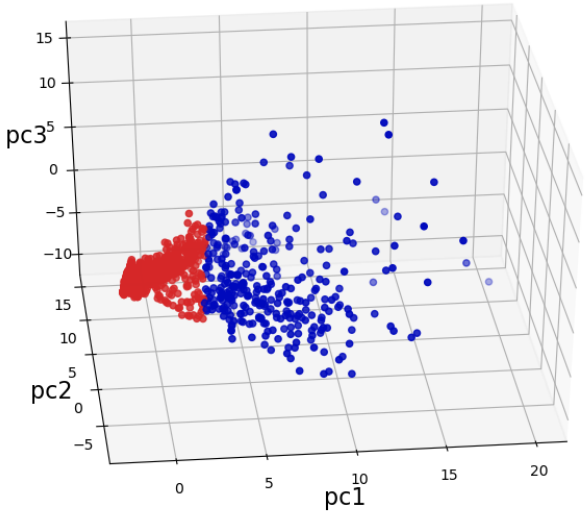

Figure 5.13. Graphical result of the first 3 PC in the manual feature extraction case.

Cluster	number	baricenter	color
0	331	(6.6, 15.1, -8.1)	blue
1	2001	(0.2, 3.3, -2.6)	red

Table 5.15. Best result in the manual feature extraction approach.

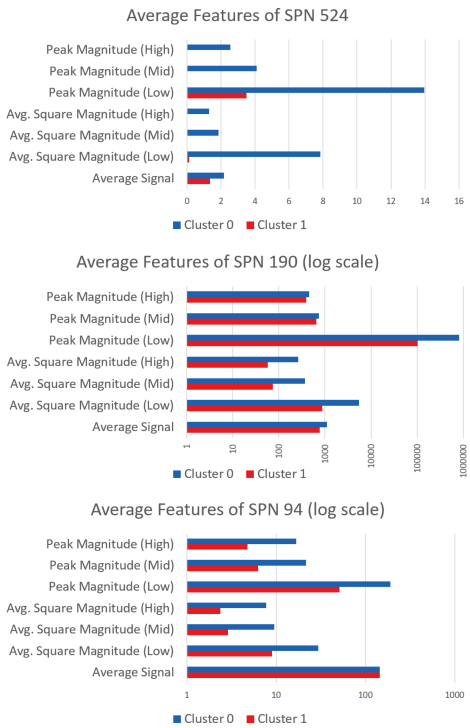

Figure 5.14. Feature average for each SPN and for each cluster.

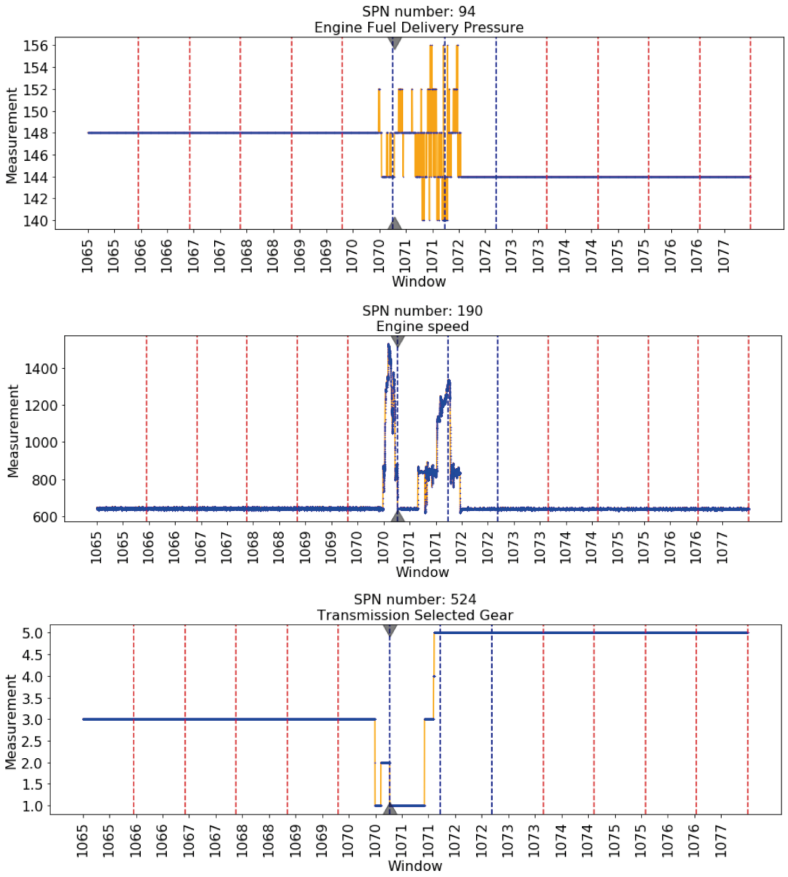

Figure 5.15. Example of time-domain clustering results. The three central windows are clustered together, while the others are mainly flat and their label is set to 1.

Chapter 6

Conclusions

In this chapter, reflections on the results obtained in the previous chapter are reported.
In the first part, criticalities and favorable points that distinguish all methods are collected. Subsequently, an optimal method is proposed, and finally, ideas on future developments, which could help to achieve a better result, are suggested.

The first observation concerns the granularity and the optimal type of aggregation/downsampling to use for this analysis. Although the analyzed datasets are always the eight explained in [Subsection 3.2.1], solutions that take advantage of a dataset with an intermediate/high level of aggregation (low frequency) have been preferred (see granularity level of the winner clustering methods in the previous chapter). These aggregations lead to a flattening of the series, but at the same time, clusters are more easily identifiable and get higher goodness measurements. In fact, an oscillatory but almost flat signal could be perceived as a completely different series with a higher granularity, while the aggregated time window could assume the most common shape among those present in the dataset. In conclusion, for the analysis carried out (and so related to agricultural machinery) an intermediate/high granularity is suggested. Logically this advice should not be taken at face value. In fact, some isolated situations, such as SPN 94 in the single SPN approach, need more resolution to understand the real differences between the windows, and therefore a preliminary analysis on the type of aggregation is still advisable for future analysis.
Moreover, significant differences in clustering results by using aggregation by average or downsampling have not been highlighted. Very often the two methods were equivalent in terms of goodness indexes if compared at the same granularity level. Therefore no preference is expressed.

Turning to the particular cases of the various algorithms used, the following conclusions can be drawn.
The starting algorithm of the previous chapter is the single signal based. This method surely has a negative side; it cannot be generalized. In fact, taking more signals of reference than three (as in this case), too many details in the division could be identified, not recognizing the macro behaviors of the vehicle. This approach, therefore, is very precise but not scalable.
On the contrary, a generic issue present in all dataset-oriented methods is the opposite; the optimal choice flattens too much the results.

Before drawing the conclusions, let's look for the best method among the dataset-oriented ones. The first one proposed is described in [Subsection 5.2.1]. This method is extremely selective for windows classified with the label 0. In fact, it can identify shaped windows, only if all three components of the multivariate series have a strong presence of low frequencies. On the contrary, in the more numerous cluster 1, the algorithm generally combines extremely flat windows with ones in which not all 3 signals have this behavior.
This problem led to the development of a measure that took into account variations in the individual SPN signal. For that reason the [Subsection 4.1.2] was developed and subsequently exploited in the application of [Subsection 5.2.2].
The operations of the maximum and the sum, in fact, highlight as anomalous a window even if only one of the three variables of the multivariate series has anomalous behavior compared to the other signals. In this way, it is possible to highlight at different levels without compression of the results. An example is given to clarify this point. Let's focus on three consecutive random windows. Let's call them window 0,1 and 2. Let's assume that only the first window has a singularity about SPN 94 and therefore this window series is very far from the second and third window series for what concerns the SPN 94. In the second window, instead, let's assume a rare behavior happening to SPN 190, which determines a big DTW dissimilarity with respect to the window series of the first and third series. Let's assume now that the SPN 524 is flat for all the periods. Under these hypotheses, if the max or sum aggregation function is considered, the first and second windows will be far away from the third window even though it is caused by different elements.
This leads to a correct result from the point of view of the diversity of the windows. The negative aspect is the flattening of the results due to the impossibility to understand which SPN causes the distance. In fact, there can be many ways in which two windows can be labeled in the same way. This second method also suffers from a lack of interpretability of the results. It is not possible to establish a priori what is the reason for the clustering but it must be based on analysis of

the clustered series after the algorithm has acted.
This led to the development of the third method. The third method involves manually extracted features that can give useful information on the structure of the time series windows. Moreover, it partially solve the issues raised previously, since it provides a way to immediately interpret the results by using the starting feature as indicators.
This last sentence is only partly true, because the process exploits a PCA, which "mixes the features". However, these were derived from real features that can be read after the clustering analysis.
This last algorithm seems to be the most reliable also from the point of view of CVIs. Not all indices can be exploited for the final comparison because, for each section, the most appropriate ones have been used. The CVIs result can be found in [Table 6.1]. By the numbers, it is clear that the feature method can better divide clusters and make them more cohesive within themselves.

Algorithm	Dunn	Silhouette
Feature - based	0.1	0.67
DTW - based ([Subsection 5.2.1])	0.1	0.46
DTW - based ([Subsection 5.2.2])	0.03	0.07

Table 6.1. CVIs of the best dataset-oriented methods identified.

In conclusion, a method that takes the positive side of both approaches is recommended. In fact, the dataset can be initially processed by the dataset-oriented method that exploits the features extraction. This method seemed more stable and interpretable than the others.
Then, signals of particular interest can be chosen, and they are exploited to apply the single signal oriented clustering method. In this way, specific groups will be created based on the shapes of these specific signals. The recommended number of selected signals does not exceed 3 but, in any case, an extremely high number of final clusters is not appropriate due to the scarce utility.
In the end, a merge of the two results can be done. The macro-categories are kept in mind but at the same time, the shape dissimilarity between the single signal windows can be used to identify a more in-depth analysis.
A future study would find out which SPN values are the best to be considered in the univariate approach compared to the initial set. It might even be a better choice to analyze other variables with respect to the ones used for the second step. In fact, in this way, the link between the univariate and multivariate clustering approach would be completely deleted. At the same time, there is a risk of obtaining incompatible results. By using the same variables for the univariate (a

part of them) and multivariate approach, it is more likely to obtain two comparable but almost overlapping results, as in this case. Besides, these analyses will be possible starting from a richer dataset in terms of the number of variables and considering a vehicle that performs routine operations rather than functional tests.

In conclusion, other investigations on multivariate methods for time series clustering will be necessary to find a method that can obtain more refined divisions already in the first step. This way the second step may no longer be necessary.

Bibliography

[AGM+13] Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, JesúS M PéRez, and IñIgo Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013.

[AHK01] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.

[BPD+11] Guy Brock, Vasyl Pihur, Susmita Datta, Somnath Datta, et al. clvalid, an r package for cluster validation. *Journal of Statistical Software (Brock et al., March 2008)*, 2011.

[BZ11] Katarzyn J Blinowska and Jaroslaw Zygierewicz. *Practical Biomedical Signal Analysis Using MATLAB®*. CRC Press, 2011.

[CC08] Jonathan D Cryer and Kung-Sik Chan. *Time series analysis: with applications in R*. Springer Science & Business Media, 2008.

[CT65] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[DD03] Susmita Datta and Somnath Datta. Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19(4):459–466, 2003.

[Dun74] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[Gio09] Toni Giorgino. Computing and visualizing dynamic time warping alignments in R: The dtw package. *Journal of Statistical Software*, 31(7):1–24, 2009.

[HKK05] Julia Handl, Joshua Knowles, and Douglas B Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[HPL02] Steve Corrigan HPL. Introduction to the controller area network (can). *Application Report SLOA101*, pages 1–17, 2002.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[Jai18] Brijnesh J Jain. Semi-metrification of the dynamic time warping distance. *arXiv preprint arXiv:1808.09964*, 2018.

[KR90] Leonard Kaufman and Peter Rousseeuw. Finding groups in data: An introduction to cluster analysis., 1990.

[LZPK18a] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. Matrix profile x: Valmod-scalable discovery of variable-length motifs in data series. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1053–1066, 2018.

[LZPK18b] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. Valmod: A suite for easy and exact detection of variable length motifs in data series. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1757–1760, 2018.

[Mak80] John Makhoul. A fast cosine transform in one and two dimensions. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):27–34, 1980.

[MC09] Andrew V Metcalfe and Paul SP Cowpertwait. *Introductory time series with R*. Springer, 2009.

[Mül07] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[PDD09] Vasyl Pihur, Susmita Datta, and Somnath Datta. Rankaggreg, an r package for weighted rank aggregation. *BMC bioinformatics*, 10(1):62, 2009.

[Pei83] Wang Peizhuang. Pattern recognition with fuzzy objective function algorithms (james c. bezdek). *SIAM Review*, 25(3):442, 1983.

[PG15] John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1855–1870, 2015.

[PKG11] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[RNS85] Enrique Vidal Ruiz, Francisco Casacuberta Nolla, and Hector Rulot Segovia. Is the dtw "distance" really a metric? an algorithm reducing the number of dtw comparisons in isolated word recognition. *Speech Communication*, 4(4):333–344, 1985.

[Rou87] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[SE17] Alexis Sardá-Espinosa. Comparing time-series clustering algorithms in r using the dtwclust package. *R package vignette*, 12:41, 2017.

[SESL17] Alexis Sarda-Espinosa, Maintainer Alexis Sarda, and TRUE LazyData. Package 'dtwclust', 2017.

[VKG14] Martin Vetterli, Jelena Kovačević, and Vivek K Goyal. *Foundations of signal processing*. Cambridge University Press, 2014.

[YHR01] Ka Yee Yeung, David R. Haynor, and Walter L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

[ZWYG18] Bobo Zhao, Zhu Wang, Zhiwen Yu, and Bin Guo. Emotionsense: Emotion recognition based on wearable wristband. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 346–355. IEEE, 2018.