Universidad de las Américas Puebla

Engineering School
Department of Computing, Electronics and Mechatronics

Doctorate in Intelligent Systems

Multiresolution Analysis for Transiting Exoplanet Identification Using Machine Learning

By

Miguel Angel Jara Maldonado

A dissertation presented in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Advisor: Dr. Vicente Alarcón Aquino

Co-Advisor: Dr. Roberto Rosas Romero

Santa Catarina Mártir, San Andrés Cholula, Puebla

May 2022

Abstract

Since the launching of the *Kepler* satellite in 2009, the discovery rate per year of exoplanets (i.e. planets found outside the Solar System) has increased meaningfully with more than $5,000$ exoplanet confirmations up to now. Exoplanet discoveries are important to understand the composition and evolution of the universe. Even more, searching for exoplanets also opens the possibility to look for life in other planets. The amount of data that have to be processed in order to perform such a discovery is immense. Nowadays, there exist several machine learning approaches, although they are not yet unerring. The underlying aim of this thesis is to develop a machine learning model based on multiresolution analysis capable of identifying if a signal detected in time series of star flux, called light curves, corresponds to an exoplanet transit or not.

First, we propose a pipeline that helps to understand the process of exoplanet discovery. Such pipeline is divided in data acquisition, data preprocessing, exoplanet detection, and exoplanet identification. Next, we compare some of the best performing machine learning models that have been applied to exoplanet identification in the literature; such as convolutional neural networks and random forests. Also, we tested preprocessing the light curves using multiresolution analysis to improve the performance of the models. Three different techniques were used, namely the discrete wavelet transform, empirical mode decomposition and its ensemble alternative. We created two datasets of synthetic exoplanet transits for the experiments. Our results show that the models classify better when using multiresolution analysis. For instance, the convolutional neural network increases its accuracy from 91.46% and 97.68% to 97.13% and 99.32%. Furthermore, our experiments show that the discrete wavelet transform helps to decrease the execution time of the models because it reduces the length of the input vectors. For example, the random forests classifier improved from 10.26 and 9.42 to 1.18 and 1.16 seconds, and the convolutional neural network improved its testing time from 46.74 and 54.17 to 31.59 and 22.93 seconds.

Finally, we have proposed a new machine learning model that uses multiresolution analysis as its core component; which we termed WAvelet-Based Broad LEarning System (WABBLES). It was tested using one of the light curve datasets aforementioned and a benchmark dataset of breast cancer, the latter to prove that the model can be used for any classification problem. The proposed model obtains better identification results than the traditional models that were tested in this work. On the one hand, for exoplanet identification, our model attained an accuracy of 99.01%, while the other models obtained an accuracy lower than 98.6%. On the other hand, for breast cancer detection, our model obtained an F-Score of 96.23% while the rest of the models did not obtain more than 94.8%.

Resumen

Desde el lanzamiento del satélite *Kepler* en el año 2009, el número de descubrimientos de exoplanetas (planetas encontrados fuera del Sistema Solar) ha incrementado notablemente con más de $5,000$ exoplanetas confirmados hasta la fecha. Su estudio es importante para entender la composición y evolución del universo. Además, estudiar exoplanetas provee la posibilidad de buscar vida fuera de la Tierra. La cantidad de información a procesar para lograr tales descubrimientos es abrumadora. Hoy en día existen diversas propuestas de aprendizaje de máquina para realizar estas tareas, aunque aún cuentan con sus limitaciones. El propósito de esta tesis es desarrollar un modelo de aprendizaje de máquina basado en el análisis multiresolución, para analizar las series de tiempo de luz estelar, llamadas curvas de luz, e identificar si las señales detectadas corresponden a tránsitos de exoplanetas.

Primero, presentamos el método general a través del cual se descubren los exoplanetas. Dicho método se divide en la adquisición de datos, su preprocesamiento, detección e identificación de exoplanetas. Luego, comparamos varios algoritmos de aprendizaje de máquina reportados en la literatura para la identificación de exoplanetas; tales como redes convolucinoales y árboles aleatorios. También presentamos experimentos utilizando análisis multiresolución para preprocesar las curvas de luz y así obtener mejores resultados de identificación. Se probaron tres técnicas: la transformada discreta wavelet, la descompocisión empírica de modos y su variante en conjunto. Hemos creado dos conjuntos de datos con tránsitos sintéticos de exoplanetas para los experimentos. Los resultados muestran que el rendimiento de los modelos mejora gracias al análisis multiresolución. Por ejemplo, la red convolucional incrementa su precisión de 91.46 % y 97.68 % a 97.13 % y 99.32 %. Además, hemos disminuido el tiempo de ejecución de los modelos utilizando la transformada discreta wavelet, la cual reduce la longitud del vector de entrada. Por ejemplo, el clasificador de bosques aleatorios mejoró de 10.26 y 9.42 a 1.16 y 1.18 segundos, mientras que la red convolucional mejoró su tiempo de pruebas de 46,74 y 54,17 a 31,59 y 22,93 segundos.

Finalmente, proponemos un nuevo modelo de aprendizaje de máquina que utiliza análisis multiresolución como componente principal; llamado Sistema de Aprendizaje Amplio Basado en Wavelets (WABBLES). El modelo fue probado para la identificación de exoplanetas y detección de cáncer de mama para probar que puede ser utilizado para cualquier problema de clasificación. Nuestro modelo obtiene mejores resultados que los modelos tradicionales que fueron probados. Por una parte, para la identificación de exoplanetas, nuestro modelo obtuvo 99.01 % de precisión, mientras que los otros modelos obtuvieron porcentajes menores a 98.6 %. Por el otro lado, para la detección de cáncer de mama, nuestro modelo obtuvo un F-Score de 96.23 %, mientras que el resto de los modelos no pudo superar el 94.8 %.

Acknowledgements

I want to express my gratitude to my advisor, Dr. Vicente Alarcon Aquino, for supporting me on every step of this project. He has guided me through the vast ocean of knowledge, thus avoiding me to get lost in it. Each feedback that he has given me was a guiding light with the objective of making me mature on my way to science. Also, I want to thank Dr. Roberto Rosas Romero, my co-advisor, for always being available to answer my questions with his useful and perceptive critical thinking. Without a doubt, this project would have been impossible without both of you.

I have no words to thank my beloved wife, Andrea. She has witnessed all the effort that has been put into materializing this work. –You have always believed in my capabilities and you have spread on me the will to continue until the end–. I also want to thank my family for believing in me and supporting me in the hard times.

I am immensely thankful for all the support that my colleagues and teachers have given to me. They have always been honest and critical at each opportunity raised to revise my work. I admire that you have always found the right words to advise me and make me grow as a better version of myself. I have no doubt that I have had the opportunity to develop myself in an enviable work environment thanks to you.

Finally, I thank the Consejo Nacional de Ciencia y Tecnlogía (CONACYT) of Mexico and the Universidad de las Américas Puebla (UDLAP) for granting me the financial and administrative support to study this doctorate degree.

The nitrogen in our DNA, the calcium in our teeth, the iron in our blood, the carbon in our apple pies were made in the interiors of collapsing stars. We are made of starstuff.

– Carl Sagan.
Cosmos

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence.

BATMAN BAsic Transit Model cAlculatioN.

BFLS Box-Fitting Least Squares.

bior2.4 bi-orthogonal 2.4.

BJD Barycentric Julian Date.

BLS Broad Learning System.

cA Approximation Coefficient.

cD Detail Coefficient.

CHEOPS CHaracterising ExOPlanets Satellite.

CNN Convolutional Neural Network.

coif5 coiflet 5.

CoRoT Convection, Rotation and Planetary Transits space observatory mission.

CWT Continuous Wavelet Transform.

db1 daubechies 1.

db5 daubechies 5.

DL Deep Learning.

DWT Discrete Wavelet Transform.

EEMD Ensemble Empirical Mode Decomposition.

EMD Empirical Mode Decomposition.

ESA European Space Agency.

FFNN Feed Forward Neural Network.

FN False Negatives.

FP False Positives.

GL Gregory-Loredo Bayesian method.

HHT Hilbert-Huang Transform.

IMF Intrinsic Mode Function.

JWST James Webb Space Telescope.

KNN k-Nearest Neighbors.

LC Light Curve.

LS Least Squares.

MAST Mikulski Archive for Space Telescopes.

ML Machine Learning.

MLP Multilayer Perceptron.

MRA Multiresolution Analysis.

MRW Multidimensional Radial Wavelon.

MSE Mean Squared Error.

NASA National Aeronautics and Space Administration.

NB Naïve Bayes.

PDF Probability Density Function.

RF Random Forests.

SNR Signal-to-Noise Ratio.

STFT Short-Term Fourier Transform.

SVM Support Vector Machine.

SWT Stationary Wavelet Transform.

sym5 symlet 5.

TESS Transiting Exoplanet Survey Satellite.

TN True Negatives.

TP True Positives.

WABBLES WAvelet-Based Broad LEarning System.

WFIRST Wide-Field Infrared Survey Telescope.

WVM Wavelet Map.

Notation

$\mathbb{R}$	Set of all Real Numbers
$\mathbb{Z}$	Set of all Integer Numbers
ω	Angular frequency
$\psi(t)$	Mother Wavelet Function
$^{\circ}C$	Degrees Celsius
a	Wavelet Dilation Parameter
au	Astronomical Unit
b	Wavelet Translation Parameter
f	Frequency in hertz
Hz	Hertz
K	Kelvin degrees
kg	Kilograms
km	Kilometers
m	Meters
M_*	Star Mass
M_p	Planet Mass

P	Orbital Period
R_*	Star Radius
R_J	Jupiter radius
R_p	Planet Radius
$R_\oplus$	Earth Radius
s	Seconds
T	Period
t	Time
v	Speed

Chapter 1

Introduction

1.1. Motivation

During the 1990's, astronomers Aleksander Wolszczan and Dale Frail [Wolszczan and Frail, 1992] validated the discovery of a planet orbiting the pulsar *PSR B1257+12* outside the Solar System. This discovery was a definitive answer to the question: *are there any planets outside the Solar System?* in a time when, as stated in [Yaqoob, 2011], there was too much skepticism regarding this topic due to the lack of proofs. Later, in 1995, astronomers Michael Mayor and Didier Queloz [Mayor and Queloz, 1995] detected the exoplanet *51 Pegasi b* by using a technique called radial velocity. The discovery of *51 Pegasi b* was an important breakthrough because it was the first time that an exoplanet was found orbiting a star in the main sequence. The term *exoplanet* stands for extrasolar planet and it refers to any planet found outside the Solar System. The study of exoplanets is important for several reasons, as stated in [Burke et al., 2014]:

- Obtaining statistical information about the universe.
- Extending our knowledge of the creation of the Solar System.
- Searching for habitable planets outside the Solar System.

- Searching for evidences of life outside the Solar System.

Exoplanet research developed rapidly after the discovery of *51 Pegasi b*. Different sky survey missions have been performed since with the aim of retrieving the observable light of stars on a determined area of the sky. One of the goals of such missions is to find exoplanet transits (when a host star is periodically hidden by a planet, which causes its stellar light to be dimmer), to look for exoplanets. The information conveyed by such missions is then turned into a time series that contains the changes of the light flux obtained from a given star; which is called a Light Curve (LC). The brightness values obtained in the LC, such as the one shown in Figure 1.1, vary as the exoplanet orbits its host star. LC also allow one to extract certain parameters, such as the beginning of ingress ($t1$), end of ingress ($t2$), beginning of egress ($t3$), end of egress ($t4$), transit length, and transit depth. LCs are carefully examined in order to look for evidences of events that could correspond to an exoplanet transit signal. Then, the detected signals are vetted to ensure that they actually correspond to an exoplanet, avoiding spurious detections.

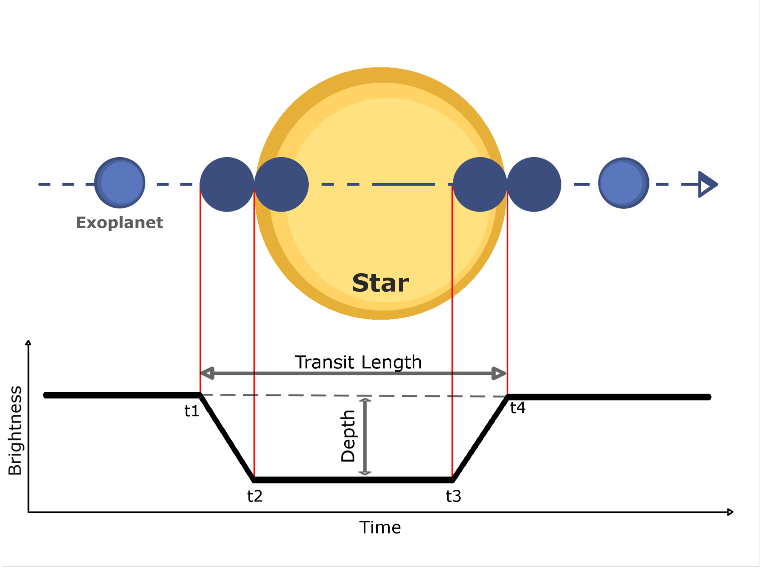


Figure 1.1: Example of a LC.

A successful example of the projects that have been dedicated to the search of exoplanets was the *Kepler* satellite. It recollected data under two missions, the first one from 2009 to 2013, where it only pointed at the Cygnus constellation until it presented a failure and it was adapted as the *K2* mission [Howell et al., 2014]. This second mission aimed at different regions but it was limited by the failure in one of the reaction wheels, which resulted in less accurate pointing.

Nevertheless, this second mission lasted from 2014 to 2018, when the satellite was definitely decommissioned. The data obtained by *Kepler* has helped to detect more than $6,000$ planetary candidates (unconfirmed signals that are likely related to exoplanets), from which, most of the confirmed exoplanets have been discovered, with more than $2,600$ confirmations (from the more than $5,000$ total discoveries up to may of 2022). Other space missions such as the Hubble space telescope, and Convection, Rotation and Planetary Transits space observatory mission (CoRoT), have provided useful information for the exoplanetary study. Furthermore, the importance of the exoplanetary research continues to be on the scope of the National Aeronautics and Space Administration (NASA), and the European Space Agency (ESA) missions. Some of the most recent space telescope missions are the Transiting Exoplanet Survey Satellite (TESS) [Ricker et al., 2014] satellite launched in 2018 to discover exoplanets, the CHaracterising ExOPlanets Satellite (CHEOPS) [Beck et al., 2017] satellite launched in 2019 to characterize exoplanets that have been detected using the radial velocity technique, and the James Webb Space Telescope (JWST) which was launched in the year 2021 with the aim of characterizing the atmospheres of the confirmed exoplanets. There are several future missions to come, for example the Wide-Field Infrared Survey Telescope (WFIRST) among others.

One of the main constraints of this research area is that the examination of LCs is time-consuming and demands considerable effort. As pointed in [Shallue and Vanderburg, 2018], the first lists of candidates were generated by manual vetting of planet candidates. Later, many Machine Learning (ML) techniques were proposed in order to identify the transiting exoplanet signals. The main reasons to develop ML algorithms for exoplanet identification are:

- The automation of a task that requires astronomers to spend time and effort.
- To allow scientists to identify exoplanets when the Signal-to-Noise Ratio (SNR) is too low to easily identify them, thus enabling to find smaller exoplanets.
- To reduce the false positive rates in the identification process.
- To profit from the capability of computational solutions to deal with great quantities of information without losing performance (for example, the *Kepler* mission has provided

us with more than 3 million files).

In order to understand how to achieve these goals, the next Section discusses several ML approaches that have been used in exoplanet research.

1.2. Related Work

Identifying transiting exoplanet signals in LCs poses several challenges. When using the transit technique (discussed in Section 2), false positive rates, and features with noise become a serious problem. Also, the necessity of dealing with datasets that contain such quantities of information, requires a robust and scalable approach. According to [Pearson et al., 2017], *the ideal planet detection algorithm should be fast, robust to noise and capable of learning and abstracting highly non-linear systems.* For this reason, ML approaches (which are derived from the Artificial Intelligence (AI) perspective) have been proposed by many authors.

The Box-Fitting Least Squares (BFLS) algorithm presented in [Kovacs et al., 2002] is an exact Least Squares (LS) solution that relies on detecting box-like events in stellar LCs. This approach allows one to detect periodic transit events in noisy time series, though it requires a good compromise between high resolution, and short execution time. Additionally, its optimization function has multiple minima, which means that the best solution is not necessarily guaranteed.

A decision tree partitions the space of attributes several times, until the attributes are considered to be partitioned enough. The works in [Coughlin et al., 2016], [Catanzarite, 2015], [Koch et al., 2010], [McCauliff et al., 2015], [Armstrong et al., 2018] and [Sturrock et al., 2019], use decision trees, in order to obtain the most representative characteristics from LC time series data. Once obtained, the decision tree is capable of determining if the source of the signal belongs to a planetary candidate, or other sources; as well as determining which are the most useful features.

A Convolutional Neural Network (CNN) is used in the works of [Shallue and Vanderburg, 2018], [Pearson et al., 2017], [Ansdell et al., 2018], [Dattilo et al., 2019], [Chaushev et al., 2019], [Yu et al., 2019], [Osborn, H. P. et al., 2020], [Pearson, 2019], [Priyadarshini and Puri, 2021] and [Schanche et al., 2018] (the last two combine CNNs with other models) to identify, detect, or even predict the parameters of exoplanetary transit signals. CNNs allow one to exploit the spatial correlation of the data. The aforementioned works are capable of detecting Earth-like exoplanets in noisy time series data by extracting their own features. One of the main limitations of these works is that their preprocessing steps often insert noise in the data; which causes problems during the identification process such as spurious identifications. Also, the nature of these algorithms makes the parameter optimization difficult because they require to test large hyperparameter grids to achieve the ideal configuration.

The works presented in [Pearson et al., 2017], [Masciadri and Raga, 2004], [Bravo et al., 2014], [Carter and Nathan Winn, 2009], [Grziwa and Pätzold, 2016], make use of Multiresolution Analysis (MRA) for different purposes, namely, to preprocess the LC data and then use it as input for an ML algorithm, to analyze the LC for automatic exoplanet identification or to determine the properties of a transit or its host star. On the one hand, the main advantage of these approaches is that, through the use of wavelets, they are capable of extracting the most significant features from the LCs, enabling scientists to detect shallow transits. In contrast, some instances only consider stationary noise problems whereas the actual noise is usually non-stationary.

Finally, [Aigrain and Favata, 2002] and [Carpano, S. et al., 2003], present a Bayesian approach based on the Gregory-Loredo Bayesian method (GL) to detect planetary transits due to terrestrial planets. It consists in computing the likelihood of the given data by using the transit parameters. These works have the advantage that they can approximate LCs with arbitrary shapes; and they also have the capability to use the information to reconstruct the signal (i.e., determine the characteristics of the exoplanet). In contrast, the main limitation of these approaches is that computing time scales linearly with the number of points in the LCs, and this implies that the more samples are fed, the more time consuming the algorithm will be.

Furthermore, the noise present in the transit signals is one of the most problematic causes of identification errors. In order to reduce instrumental and astrophysical noise from the transit signals, some works present a preprocessing step, such as using unsupervised learning to discard noise [Mislis et al., 2018], among other ML methods that are out of the scope of this work, which are presented in [Jara-Maldonado et al., 2020b]. Even though these methods reduce the quantity of missed detections, they are not yet unerring. Also, as mentioned in [Petigura et al., 2013], small Earth-like planets, which are of great interest due to their potential capabilities of being habitable, tend to be difficult to spot. The reason for this is that they are commonly found on LC signals highly contaminated by noise. For these reasons, an MRA approach is proposed in this thesis, which is capable of obtaining information of interest at different frequency resolutions.

1.3. Aim of Thesis

The aim of this thesis is to improve the transiting exoplanet identification rates by using MRA for a resolution-level-adapted inspection of LC time series data. MRA is advantageous to the exoplanet research field because it allows one to analyze signals at different resolution levels. This makes it possible to obtain characteristics of the signal that otherwise could be lost due to the noise present in the LCs. Therefore, the underlying purpose of the work presented in this thesis is to develop a MRA algorithm that is capable of identifying exoplanet transits in noisy LC time series. This study is based on the hypothesis that: *a machine learning model based on multiresolution analysis is capable of overmatching the transiting exoplanet identification accuracy performance of the state-of-the-art models applied to noisy light curve classification.*

1.3.1. Specific Objectives

The specific objectives of the underlying work reported in this thesis are:

- To select existing LC modeling tools publicly available that can be used to assess the exoplanet identification performance of different ML models.

- To choose MRA techniques that can be applied to LCs.
- To assess the accuracy and time performance obtained by different Machine Learning (ML) models, using LCs preprocessed using the MRA techniques chosen. This is done with the aim of identifying the best MRA technique for exoplanet identification.
- To propose and implement a ML model based on MRA capable of sensitively identifying exoplanet signals within noisy LCs.
- To compare the accuracy performance of the proposed WAvelet-Based Broad LEarning System (WABBLES) model against other models reported in the literature.

1.4. Contributions

The main contribution of this work relies on the proposed MRA approach for exoplanet identification. By using the different decomposition levels available in the transiting signal, it will be possible to extract the most significative pieces of the LC, thus providing an advantageous set of features in comparison to the works reported in the literature. Among the different contributions that this model will provide are:

- The design and implementation of a new ML model based on the Broad Learning System (BLS) and MRA for exoplanet identification, which we termed WABBLES.
- A comparative framework of different MRA and ML techniques that can be applied to exoplanet identification. This comparison is based on performance metrics such as identification accuracy and execution time.
- A guideline to apply different MRA techniques to preprocess LCs. It is proven in this work that these techniques enhance the identification performance of the state-of-the-art ML models. The guideline is presented along with the results obtained by three different MRA techniques that were applied to LCs, namely the Discrete Wavelet Transform (DWT),

Empirical Mode Decomposition (EMD), and Ensemble Empirical Mode Decomposition (EEMD) techniques.

- A proposal to improve the robustness of the ML models to different sources of noise in the input data through the use of MRA techniques.

- The application of some ML models that had not been used yet for exoplanet identification, such as the BLS algorithm.

- A pipeline for synthetic LC dataset creation and how it can be used to train and compare diverse ML models.

- A pipeline to be considered in order to automate the complete exoplanet discovery process.

- The proof of the potential of MRA techniques applied to exoplanet detection and identification.

1.4.1. Socio-economic Impact

Using an MRA-based ML model for discovering new exoplanets in data that otherwise is hard to analyze (due to different phenomena such as noise, astrophysical false positives, feeble transits, etc.) will enable to update the current exoplanet database with new information. This is important because:

- It will allow the creation of automated methods based on MRA for exoplanet identification and other astronomical areas of study. As explained in [Thompson et al., 2015], *as the size and complexity of astronomical data increases, the analysis of these data sets will need to become increasingly automated.*

- Having a wide variety of exoplanets on which astronomers may conduct further study will enable scientists to better understand the formation of the universe, galaxies, and the Solar system. [Yu et al., 2021].

- According to [Yaqoob, 2011], in nearly 5 billion years from now, our Sun will exhaust its main fuel income, i.e. hydrogen. This will probably result in a great expansion of the Sun in which the Earth would become engulfed. This situation along with some others (such as population growth) that threaten the survival of the human race pose a new motivation for finding exoplanets in the habitable zone. The first step to achieve such a feat is generating a catalogue of exoplanets in the habitable zone and to characterize those exoplanets. This knowledge could be used for future generations that may have the required technology to travel to distant worlds.
- Discovering terrestrial planets in the habitable zone (where liquid water may exist), increases the probabilities of discovering life outside planet Earth. This would provide a definitive answer to the question: *Are we alone?*
- Finally, mining in space is a promising business for the future. As stated in [Vázquez, 1981], there already exist different international regulations for space exploration as well as for freedom of space exploitation.

1.5. Thesis Outline

The work reported in this thesis is organized as follows.

In Chapter 2, an overview of exoplanet research is presented. The main techniques used to identify exoplanets are reviewed, as well as some general properties of exoplanets. Also, a background of MRA techniques and ML is provided.

Chapter 3 presents the implementation of different ML models applied to exoplanet identification. Also, a comparison of the performance of such models is reported here. This chapter also includes the results of using MRA techniques to preprocess LCs before employing them as inputs for different ML models, along with the comparison of those results.

The proposed WAvelet-Based Broad LEarning System (WABBLES) model is presented in Chapter 4. This chapter describes the architecture of the algorithm. The advantages and limitations of this approach are also discussed in this section.

Chapter 5 presents the conclusions and future research directions of this work.

Finally, all the publications related to this work are listed in Appendix A.

Chapter 2

Theoretical Background

Discovering exoplanets is a complex task. First, there are several techniques that can be used to detect and identify an exoplanet. Once an exoplanet is discovered, the characterization of its properties is another challenge. It commonly involves making indirect measurements that could require a prior knowledge of its host star properties. In this chapter, a general overview of exoplanets is given, which presents the main properties of exoplanets. Then, the main techniques used for exoplanet detection are discussed. Finally, this chapter introduces some of the basic concepts of MRA and ML. It should be noticed that the ML approaches referenced in Chapter 1, as well as the MRA approach proposed in this work are exclusively focused on detecting and identifying exoplanets through the transit technique.

2.1. Exoplanet Overview

Exoplanets are planets found outside the Solar System. Studies with the aim of discovering new exoplanets have recently started to give satisfying results. One of the main goals of exoplanetary missions such as *Kepler*, is determining which planets could host life. A habitable planet can be defined as a planet capable of sustaining life [J. Lissauer, 2000]. Under the assumption that organisms need water to survive, the *habitable zone* is defined by the Encyclopedia Britannica[1] as *the orbital region around a star in which an Earth-like planet can possess liquid water on its surface and possibly support life.* For this to be possible, an exoplanet has to be close enough to the star for water not to be frozen, but far enough to ensure that the temperature allows for liquid water. The Habitable zone is exemplified in Figure 2.1, where an exoplanet is found at a suitable distance from its host star to sustain life. Even more, as mentioned in [Seager and Bains, 2015], looking for life in the habitable zone is based on the premise that life in the whole universe is similar to life as we know it; but life could exist in exoplanets found outside the habitable zone. Nevertheless, up to now no proof of life has been found outside the planet Earth.

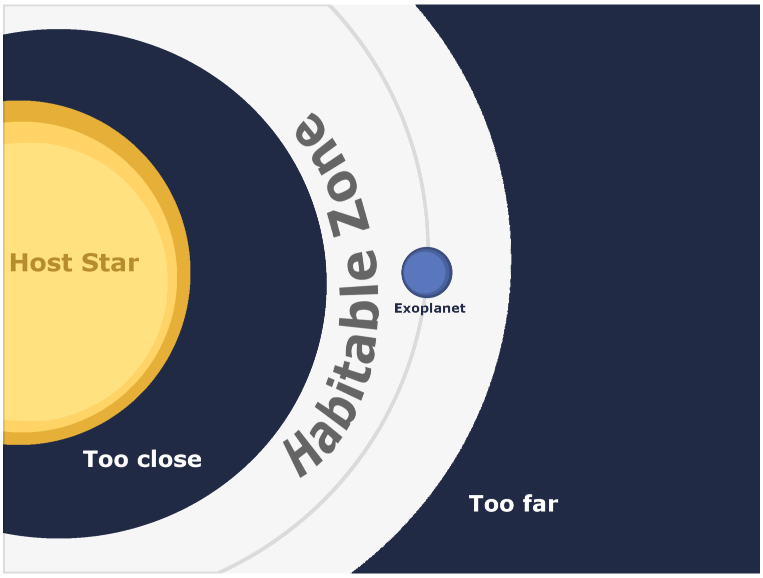


Figure 2.1: The habitable zone is found not so far nor so close to the star to allow for liquid water to be formed.

According to [Yaqoob, 2011], obtaining different measurements from exoplanets is a difficult

[1] `https://www.britannica.com/science/habitable-zone` by Jack J. Lissauer

process because it cannot be directly achieved. Estimating measurements of the properties of an exoplanet first requires to calculate some parameters from its host star. On the one hand, the most fundamental parameters from the host star to be measured are its mass (M_*), radius (R_*), and distance to Earth. On the other hand, the most important exoplanet parameters are mass (M_p), radius (R_p), and temperature (which is also important for determining its capacity to support liquid water). Other exoplanet properties used to determine life sustainability are the distance of the exoplanet to its host star and atmospheric data such as its chemical composition. Table 2.1 presents some of the properties of exoplanets along with their common metric units according to [Yaqoob, 2011] and the NASA planetary fact sheet[2]. Notice that all these properties are limited to estimates, and, thus, do not imply that exact measurements are obtained. Also, different detection techniques allow astronomers to estimate different properties, which means that some properties can only be observed by using a certain detection method.

2.2. Exoplanet Detection Methods

The breakthrough discovery of *51 Pegasi b* in 1995 was performed by using the radial velocity method, which consists in analyzing the Doppler shift effect that results from the mutual gravity of the host star and the exoplanet. A stellar wobble is produced as the star moves around the barycenter of the system. This can be measured by observing the differences in wavelength from the star's spectrum. This method has been previously used in binary star astronomy, as mentioned in [Wright, 2017], where a binary system consists on two stars orbiting each other. The main limitation of this approach is that stellar wobbles can be difficult detect due to *jittering* and because they require very precise instruments. Table 2.2 shows the percentage of exoplanets that have been confirmed using each of the different detection techniques. The information of this table was retrieved from the NASA exoplanet website[3].

Another method that has proved to be very effective for exoplanet detection is the transit technique. As demonstrated in Figure 2.2 (generated using information from the NASA Ex-

[2] `https://nssdc.gsfc.nasa.gov/planetary/factsheet/`

[3] Source: `https://exoplanets.nasa.gov/`. Consulted in May 18, 2022.

Table 2.1: Indirectly Obtainable Exoplanet Properties.

Property	**Metric Unit**
Mass.	*Kilograms* (in some cases it is compared to the mass of the Earth or another planet).
Radius.	In Jupiter radius R_J (for giants) and Earth radius $R_\oplus$ (for small ones).
Temperature.	K or $^\circ C$.
Density.	kg/m^3.
Chemical composition and associated atmosphere.	-
Distance of the exoplanet to its host star.	Typically expressed in astronomical units (au).
Surface gravity.	m/s^2.
Inclination angle of the orbit.	Degrees.
Orbital eccentricity.	There are no units for eccentricity. 0 means a perfect circle orbit and as the eccentricity comes close to 1 it means that the elliptical orbit is more elongated.
Orbital period.	Days.
Dimensions of the orbital ellipse.	Major and minor axis typically expressed in au.

oplanet Archive[4]), most of the known exoplanets have been discovered by using this method; despite the statistically restricting conditions that have to be met for a transit to occur. Transits are similar events to solar eclipses. In a solar eclipse, when the moon is exactly between the observer and the Sun, the light of the Sun that is received on Earth appears dimmer. A transit in contrast, happens when an exoplanet (or a planet) passes between the observer and the star the aforementioned planet orbits. Some of the parameters that may be estimated by using the transit method are the planet radius R_p, star radius R_*, orbital period P, and some characteristics of the planet. The main limitations of this method are that it can only find planets with orbits passing between the observer and the star, as well as its vulnerability to false positives caused by multiple noise factors.

[4] NASA Exoplanet Archive: `https://exoplanetarchive.ipac.caltech.edu/index.html`

Table 2.2: Percentage of exoplanet confirmations according to the method used.

Method	Detection percentage
Transit method.	76.6%.
Radial Velocity.	18.4%.
Microlensing.	2.6%.
Direct Imaging.	1.2%.
Others.	1.2%.

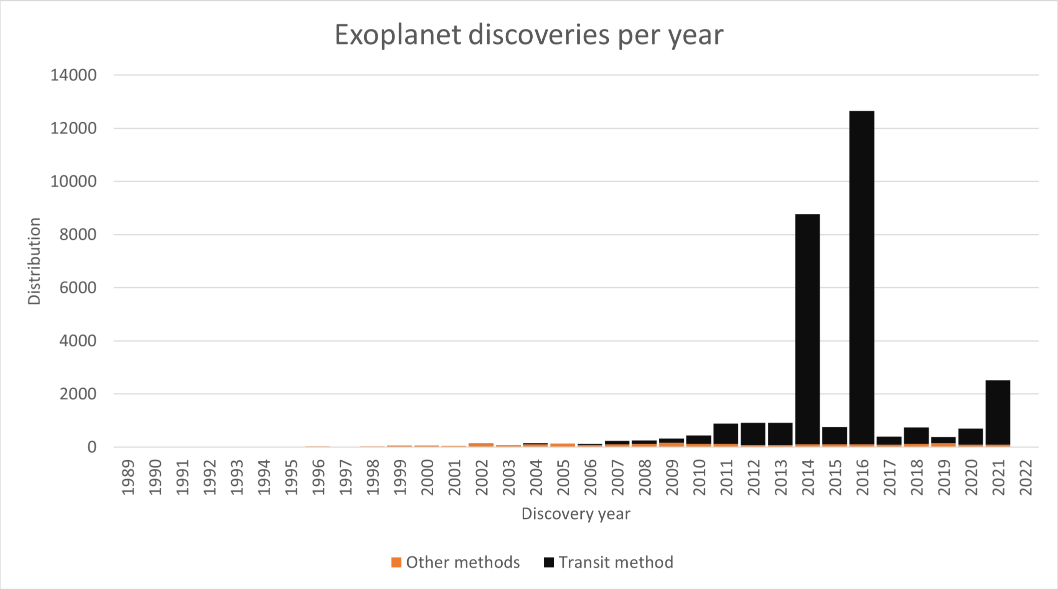


Figure 2.2: Distribution of exoplanets found up to 2020. The dark area corresponds to the number of exoplanets found using the transit search method, while the orange area corresponds to those exoplanets found using any other detection method. This figure was generated using the data from the NASA Exoplanet Archive.

The transit method uses LCs, which are time series with the quantity of light in a certain filter/band/wavelength range from each star. The light fluxes of the LCs are obtained by space-borne and ground-based instruments. If a transit occurs, a drop in the level of brightness will be shown in the LC. An example of a real LC is presented in Figure 2.3, where the x-axis represents a measure of time called Barycentric Julian Date (BJD). The y-axis represents the brightness of the star, where flux values are scaled from the minimum to the maximum value. The LC from Figure 2.3 was generated by using the data from a planetary system around the star *HIP 41378*, stored in the the MAST archive[5], as explained in the Transiting Light Curve Tutorial[6].

[5] MAST Archive: `http://archive.stsci.edu/kepler`

[6] Transiting Light Curve Tutorial: `https://www.cfa.harvard.edu/~avanderb/tutorial/tutorial4.html`

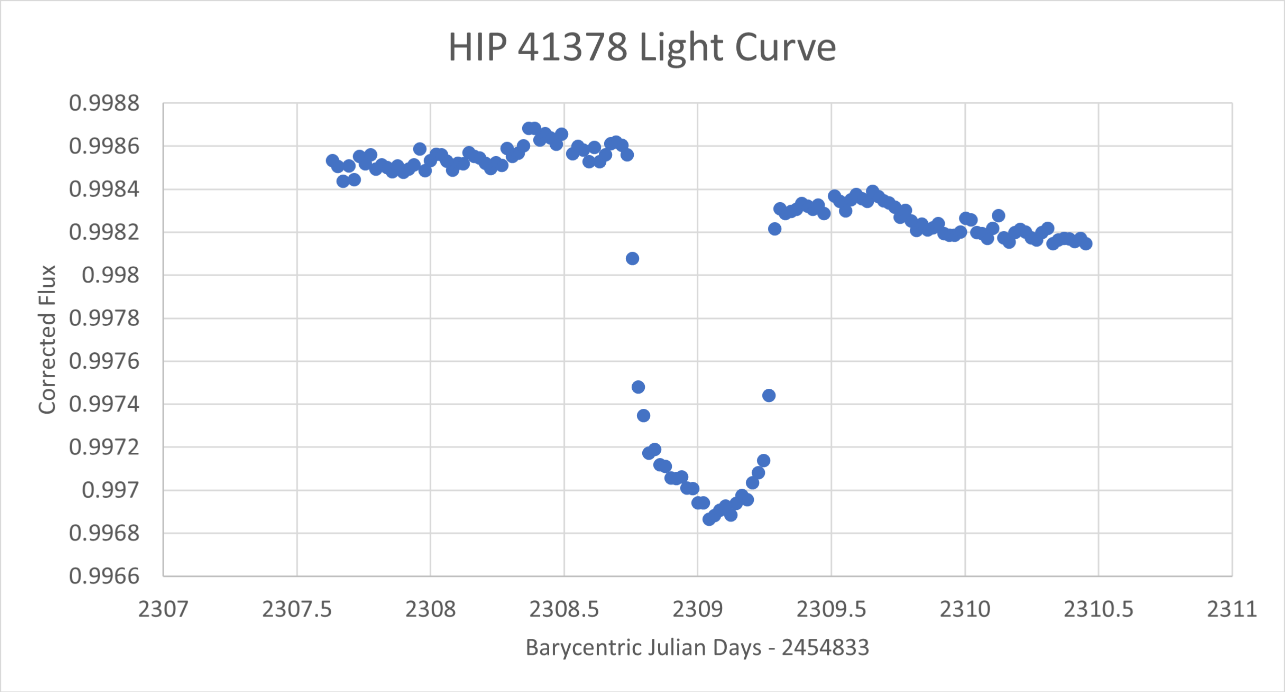


Figure 2.3: Light Curve (LC) showing a transit of the exoplanet HIP 41378 f. Source: MAST archive.

The distortion in light trajectories caused by massive objects, such as stars or planets, can change the direction of light, generating a gravitational lensing effect on the light of a star as described in [Treu et al., 2012]. The microlensing method looks for short-time star brightness rises caused by the gravity of an exoplanet, which works as a massive lens. The main limitations of this method are that the required alignment of the star with the planet and the observer is unlikely to happen, and astronomers cannot predict where or when the lensing events will occur.

A further technique used for exoplanet identification is direct imaging, where the planet and stars are well resolved. Direct imaging is a promising technique for the future generation of giant telescopes. Finally, other techniques, such as astrometry, are also used; although exoplanet identifications using these methods are scarce. Finally, Figures 2.4 and 2.5 shows a relation between the confirmed exoplanets and the method that detected them. The detections are shown according to the planet radius or mass (M_p or R_p) and orbital period (P). The figure was generated using the tools from the NASA Exoplanet Archive[4] (May, 2022).

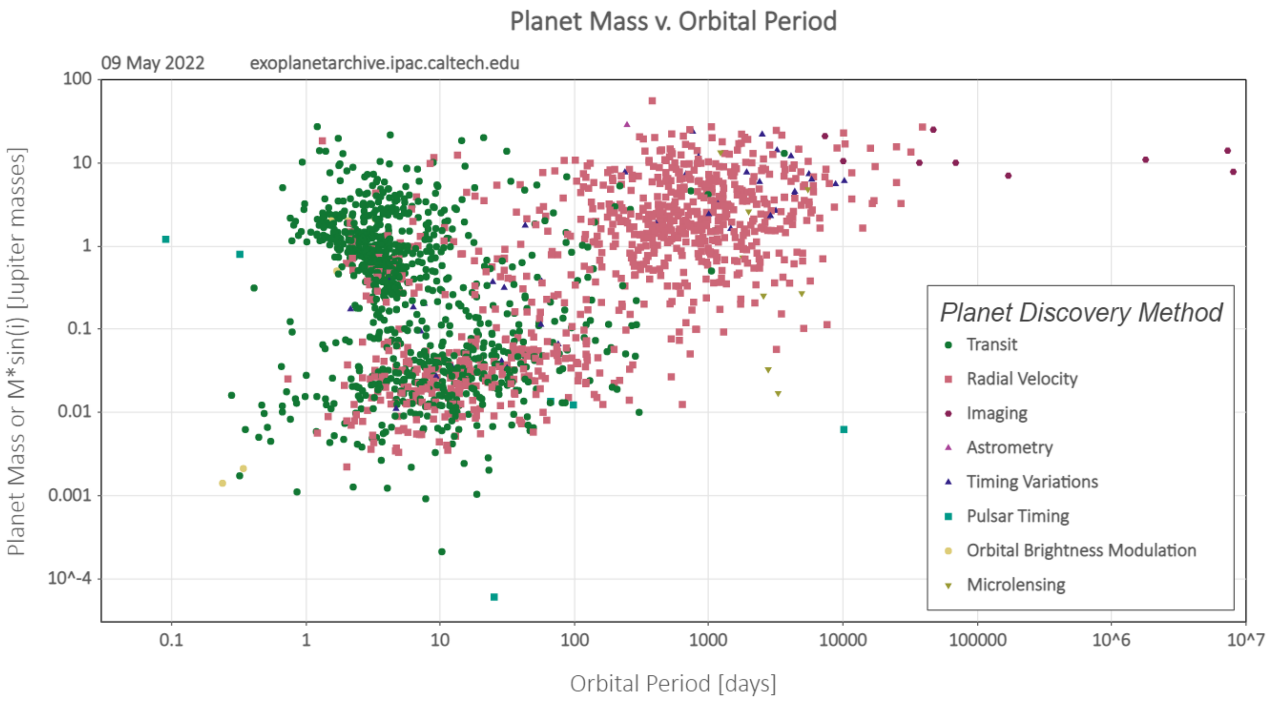


Figure 2.4: Planet mass vs. orbital period of the confirmed exoplanets according to the detection method used. Source: NASA Exoplanet Archive.

2.3. Multiresolution Signal Processing

According to [Graps, 1995], MRA allows one to divide signals into different frequency components, and then analyze these components with a resolution that matches in scale. In order to describe the MRA theory in more detail, a brief overview of the fundamentals of signal analysis is provided next.

A period is the time taken to complete one full cycle, in the case of waves, it is the time necessary for one complete vibration to be achieved [de Mayo, 2015]. As an example, the period of a planet T of radius R_p, that travels around a completely round orbit at speed v, is the time it takes to go all the way around its orbit, and it is calculated by using Eq. (2.1) [Newton and Henry, 2000].

$$T = \frac{2\pi R_p}{v} \tag{2.1}$$

Furthermore, the frequency of a wave is the number of vibrations occurring per unit of time.

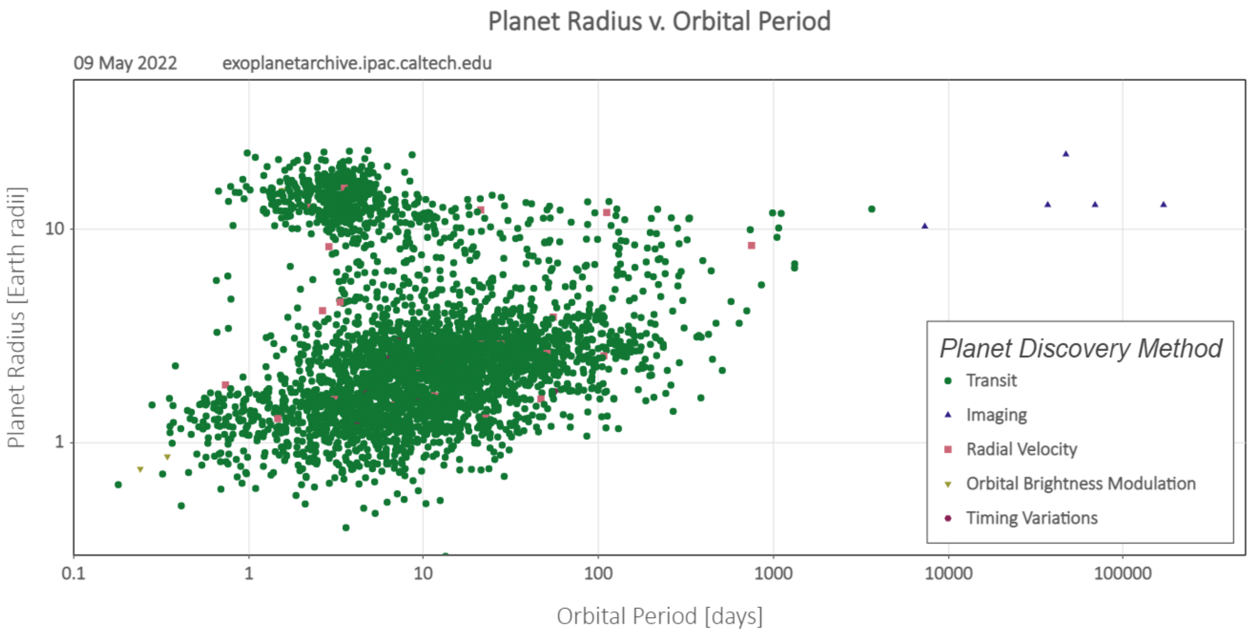


Figure 2.5: Planet radius vs. orbital period of the confirmed exoplanets according to the detection method used. Source: NASA Exoplanet Archive.

In the previous example, the frequency would be the number of times that the planet completes its orbit, per unit of time. Frequency can be calculated through the use of Eq. (2.2) [de Mayo, 2015].

$$f = \frac{1}{T} \tag{2.2}$$

where T is the period measured in seconds and the frequency f is often expressed as Hertz (Hz).

From a historical perspective, in the 19th century, Joseph Fourier contributed much to frequency analysis. He introduced the Fourier series, whose aim is to express a periodic function $f(t)$ (i.e. a function whose values are repeated at regular intervals), with period T, as a linear combination of sines and cosines as Eq. (2.3) [Graps, 1995] shows.

$$f(t) = a_0 + \sum_{k=1}^{\infty} \{a_k cos(kt) + b_k sin(kt)\} \tag{2.3}$$

where t is time; and a_0, a_k, and b_k are calculated using Eq. (2.4) [Graps, 1995].

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(t)dt, \quad a_k = \frac{1}{\pi}\int_0^{2\pi} f(t)cos(kt)dt, \quad b_k = \frac{1}{\pi}\int_0^{2\pi} f(t)sin(kt)dt \tag{2.4}$$

In summary, in Eq. (2.3), each term of the sum represents a harmonic of the function. The Fourier Transform $F(w)$ presented in Eq. (2.5) [Bravo et al., 2014], allows one to determine how much influence has each frequency in the signal; nonetheless, it provides no information of how the frequency is related to time.

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-2\pi iwt}dt \tag{2.5}$$

where ω is the frequency, and t represents time. Due to the fact that the Fourier transform provides no information of how frequency is varied over time, Dennis Gabor [Gabor, 1947] modified the Fourier transform to create the Short-Term Fourier Transform (STFT), or Gabor transform. It consists in dividing the signal into segments of a fixed length, small enough to obtain a stationary signal, to then apply the Fourier transform. The STFT is defined in Eq. (2.6).

$$STFT(b, f) = \int_{-\infty}^{+\infty} x(t)g^*(t-b)e^{-i2\pi ft}dt \tag{2.6}$$

where $*$ stands for the complex conjugate and $g(t)$ is the analysis window. The window $g(t-b)$ is a localised function that is varied over time to compute the transform at several b positions. Still there remained a limitation, the STFT does not provide which frequency components exist at any given time, i.e., it is only possible to know which frequency band exists at a given time interval. This resulted in a contradiction between the size of the window and the type of resolution searched: A wide window provides better frequency resolution but poor time resolution, whereas a narrow window furnishes a better time resolution and a poor frequency resolution. This trade-off is known as the Heisenberg uncertainty principle, and it is applied to time-frequency information [Bravo et al., 2014].

2.3.1. Wavelets

Wavelets provide an useful tool for analyzing non-stationary and non-periodic signals. They are oscillatory signals of short duration, whose energy is finite or concentrated in a determined interval of time. According to [Daubechies, 1992], the wavelet transform depends on scale (i.e. frequency) and time, thus it can be used for time-frequency localization. Also, using the wavelet transform allows one to obtain the different frequency components of a signal, to then study each component with a resolution that matches to each scale. At lower resolutions, the details belong to larger structures, while at higher resolutions, more detailed information of the signal is obtained. As an analogy, Amara Graps compares lower and larger resolutions to zooming in and out in a forest [Graps, 1995]. By using a lower resolution (i.e. a large “window“), one would be able to see a forest, but by using a higher resolution (i.e. a small “window“), it would be possible to see each tree with more detail.

To be considered as a wavelet, a function has to comply with certain criteria. In the space of all the square integrable functions $L^2(\mathbb{R})$, a function $\psi(t) \in L^2(\mathbb{R})$ is a wavelet if and only if it complies with the *admissibility condition*, which establishes that ψ must take both positive and negative values, and that its Fourier transform Ψ is zero at the zero frequency. The admissibility condition implies that the Fourier transform $\Psi(f)$ of the function satisfies the relation shown in Eq. (2.7) [Daubechies, 1992].

$$\int_{-\infty}^{+\infty} \frac{|\Psi(f)|^2}{|f|} df < +\infty \tag{2.7}$$

The admissibility condition also implies that the wavelet is a pass-band because the Fourier transform of $\Psi(f)$ vanishes at the zero frequency, meaning that the average value of the wavelet in the time domain must be zero; as shown in Eq.(2.8) [Chui, 1997].

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \Leftrightarrow \Psi(0) = 0 \tag{2.8}$$

Another condition that the function must comply in order to be considered as a wavelet is that its total energy must always be equal to the unity. In certain cases, the function must also comply with the orthogonality property. The number of vanishing moments in the function $\psi(t)$ is related to Eq. (2.8). Concretely, the number of vanishing moments v of a wavelet is given by Eq. (2.9) [Mallat, 2008], which also establishes that a wavelet with v vanishing moments is orthogonal to polynomials of degree $v-1$, i.e. the smoothness (differentiability) of $\psi(t)$ is related to the number of vanishing moments.

$$\left.\frac{\delta^k \Psi(f)}{\delta f^k}\right|_{f=0} = \int_{-\infty}^{+\infty} t^k \psi(t) dt = 0, \text{ for } 0 \leq k < v \text{ and } k \in \mathbb{Z} \tag{2.9}$$

In the wavelet theory, a prototype function, which is called mother wavelet $\psi(t)$, is used to define the daughter wavelets $\psi_{a,b}(t)$. These functions are dilated or translated versions of the mother function built by applying the b translation and a dilation (scale) parameters to the mother wavelet, as Eq. (2.10) shows.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \qquad a, b \in \mathbb{R}, a \neq 0 \tag{2.10}$$

where $\mathbb{R}$ is the set of all real numbers, a is the dilation parameter, b the translation parameter, the constant number $\frac{1}{\sqrt{a}}$ is an energy normalization factor so that the transformed signal will have the same energy at every time scale, and t represents time. According to [Daubechies, 1992], small frequency ranges are covered with large values of the scaling parameter a, and high frequency ranges are covered with small values of a. Changing the value of the translation parameter b allows one to center the time localization.

2.3.2. Continuous Wavelet Transform

The Continuous Wavelet Transform (CWT) operates at all possible scales and translations, i.e. the dilation and translation parameters are varied continuously over $\mathbb{R}$, with a different from

0. The formula for the CWT is shown in Eq. (2.11) [Daubechies, 1992].

$$CWT_f(a,b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}dt = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right)dt \qquad (2.11)$$

where the $a \in \mathbb{R}^+$, $a \neq 0$ and $b \in \mathbb{R}$ are the dilation and translation parameters respectively.

Finally, an example of a continuous wavelet (called Morlet wavelet) is presented in Figure 2.6.

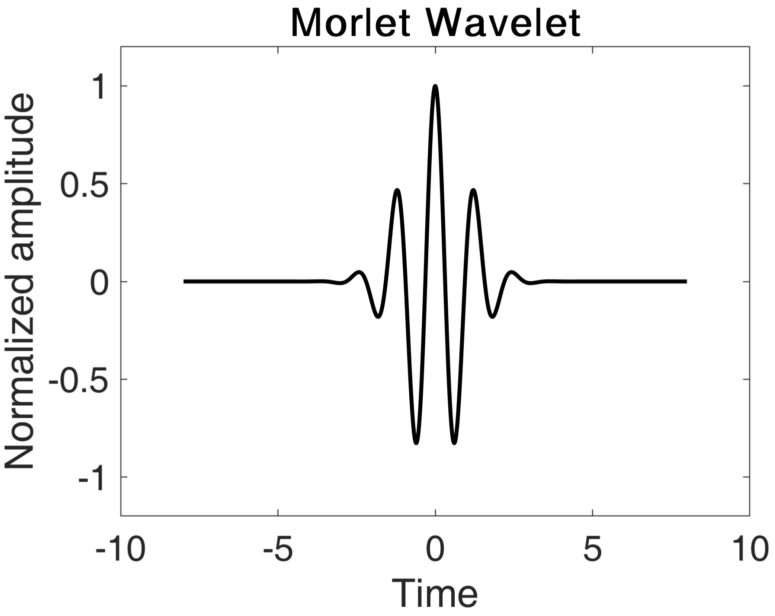


Figure 2.6: Morlet wavelet.

2.3.3. Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) uses a specific set of scales and translations, which means that a and b take discrete values. For a, the positive and negative powers of one fixed dilation parameter $a_0 > 1$ are chosen and are denoted as a_0^m. The discretization of the translation parameter b depends on m. For high frequency wavelets, the number of translations will be greater because smaller steps are required to cover the entire range of time. Conversely, for low frequencies, the translation steps will be greater. Since the width of the wavelet depends on a_0^m, b is discretized by $b = n\, b_0\, a_0^m$, where $b_0 > 0$ is fixed, and $n \in \mathbb{Z}$. The formula for the

DWT is shown in Eq. (2.12).

$$DWT_f(j,k) = a_0^{\frac{-j}{2}} \int_{-\infty}^{\infty} f(t)\,\psi\big(a_0^{-j}t - n\,b_0\big)dt, \qquad j,k \in \mathbb{Z} \tag{2.12}$$

where $\mathbb{Z}$ is the set of all integer numbers, $a = a_0^m, b = nb_0a_0^m$, and $a_0 > 1$ and $b_0 > 1$ are fixed [Foster, 1996].

The dilation and translation parameters are shown in Figure 2.7, which compares the time-frequency grid between STFT and wavelets. In the left grid, all windows have the same size, whereas in the right grid, there are different frequency and time resolutions. Also, under the right grid, there are two wavelets that correspond to the highlighted areas in the wavelet grid, these wavelets exemplify two dilation (a_0 and a_1) and two translation (b_0 and b_1) parameters of the wavelet.

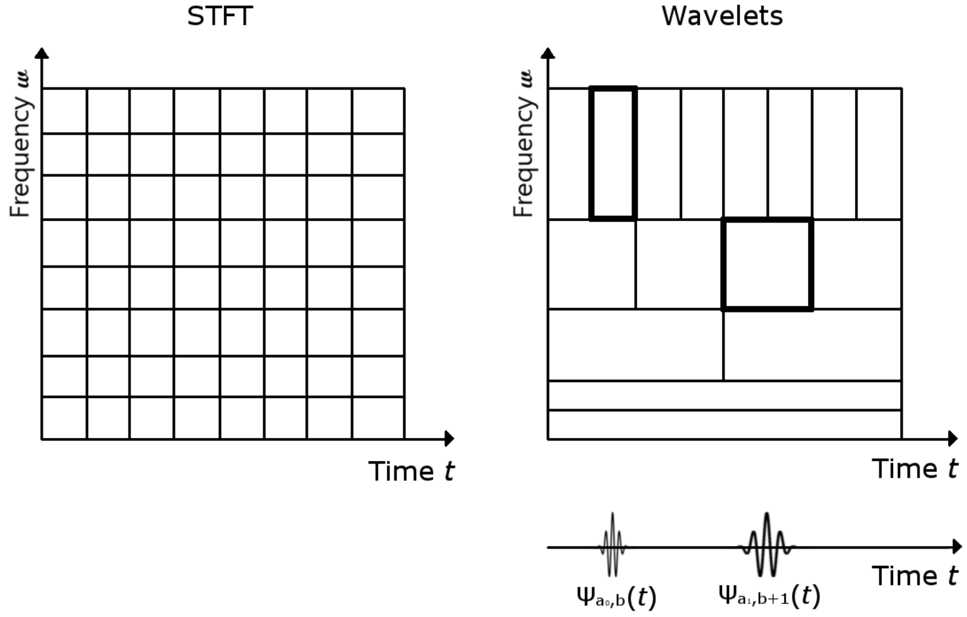


Figure 2.7: STFT vs wavelet time-frequency tiles.

2.3.4. Multiresolution Analysis

According to [Mallat, 1989], multiresolution approximations consist in sequences of embedded vector spaces $(V_j)_{j\in\mathbb{Z}}$ for approximating $L^2(\mathbb{R})$ functions. Even more, the wavelet function $\psi(t)$

can be approximated from any multiresolution approximation when $(\sqrt{2^j}\psi(2^j t - k))_{(k,j)\in\mathbb{Z}^2}$ is an orthonormal basis of $L^2(\mathbb{R})$. Specifically, the sequences of sub-spaces $(V_j)_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$ must follow the properties enlisted next:

(I) $V_j \subset V_{j+1} \quad \forall j \in \mathbb{R}$,

(II) $\bigcup_{j=-\infty}^{\infty} V_j$ is dense in $L^2(\mathbb{R})$ and $\bigcap_{j=-\infty}^{+\infty} V_j = \{0\}$,

(III) $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1} \quad \forall j \in \mathbb{Z}$,

(IV) $f(x) \in V_j \Rightarrow f(x - 2^{-j}k) \in V_j \quad \forall k \in \mathbb{Z}$,

(V) There exists a function $\phi(t) \in V_0$ such that $\{\phi_{j,n}(t) = 2^{-j/2}\,\phi(2^{-j}t - n) : j, n \in \mathbb{Z}\}$ is an orthonormal basis of V_0.

The first property denotes the successive subspaces used to represent the different resolutions or scales. The second property guarantees the completeness of the subspaces, and ensures that when the resolution is incremented, the approximated function converges into the original signal. The third property denotes that V_{j-i} is composed by all the rescaled versions of V_j. The fourth property implies that translations over a function are part of the same space as the original function. Finally, the fifth property establishes the $\phi(\cdot)$ scaling function.

2.3.5. Stationary Wavelet Transform

One of the main disadvantages of the DWT is that is loses information during the decimation step of each decomposition level. Even more, the reduced length of the data may limit the number of decomposition levels to be applied to the signal. For instance, a CNN architecture may require its inputs to have a minimum length due to the pooling layers of the network. The number of decomposition levels to be allowed in DWT would be limited by the minimum length of the inputs of the CNN. The Stationary Wavelet Transform (SWT), from [Nason and Silverman, 1995], does not decimate the coefficient sequences at each decomposition level. Instead, it pads each high-pass and low-pass filter with zeros. In this way, the gaps left by the decimation step are filled. For this reason, there is no information loss, and the length of the data is not altered. The zero padding process is performed by recursively defining Eq. 2.13 [Nason and Silverman, 1995].

$$a^{j-1} = H^{J-j} a^{j} \text{ and } b^{j-1} = G^{J-j} a^{j} \tag{2.13}$$

where H and G are high-pass and low-pass filters respectively, which have been alternated with sequences of zeroes defined by a Z operator, so that, for all integers j, $(Zx)_{2j} = x_j$ and $(Zx)_{2j+1} = 0$.

2.3.6. Empirical Mode Decomposition

According to [Zeiler et al., 2010], the Empirical Mode Decomposition (EMD) method allows non-linear analysis for complex non-stationary time series. As stated in [Mandic et al., 2013], EMD iteratively decomposes the signal as a sum of every Intrinsic Mode Function (IMF) obtained. IMFs are created by considering the coexisting simple oscillatory modes locally present and superimposed in a signal $x(t)$. [Fontugne et al., 2017] state that IMFs are a finite set of non-stationary and oscillatory components with decreasing frequencies. The process of EMD is shown in Figure 2.8. The first step is to look for the IMFs by putting the signal through an iterating process called sifting. In this process, the upper and lower envelopes of the original signal are obtained, averaged, and subtracted from it. Then, if the result complies with two conditions, it is stored as an IMF and the process is repeated again with the residual of the IMF subtracted from the signal to look for the next IMF. The two conditions needed for an IMF are that in the whole data set the number of extrema and the number of zero crossings must be equal or differ at most by one, and, at any data point, the mean value of the local maxima envelope and the local minima envelope must be zero. If the result is not an IMF, then the process is repeated in the same way as if the IMF was found, but the IMF is not stored. This process is repeated until the residue is a monotonic function (i.e. a function with only one extremum).

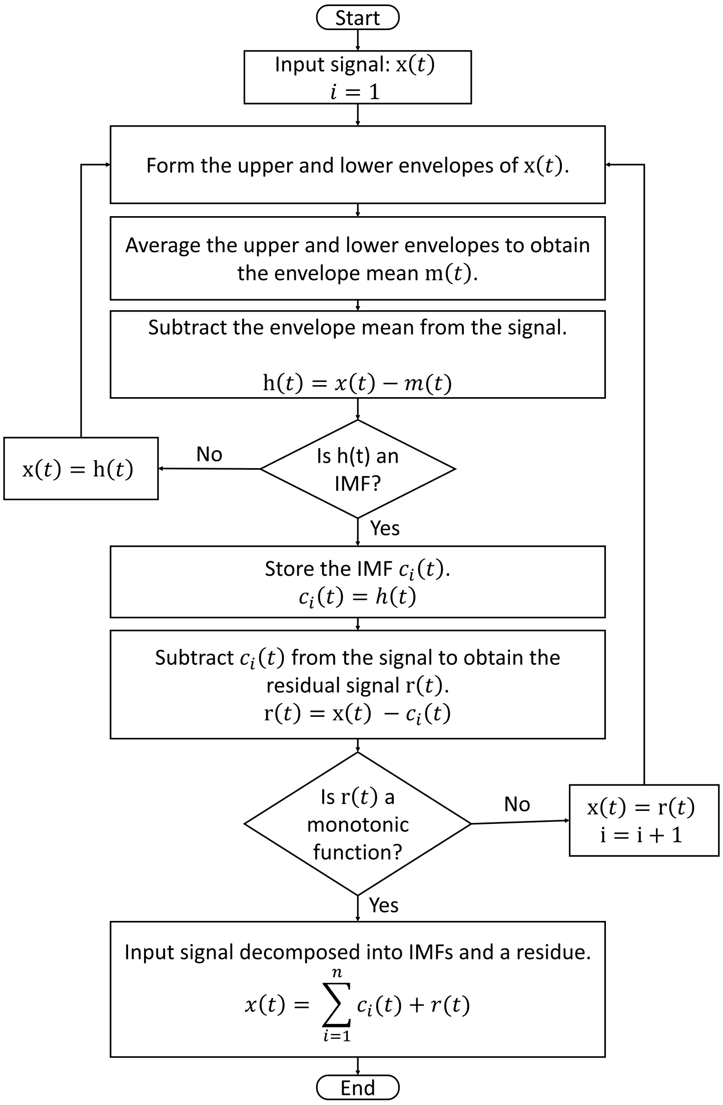
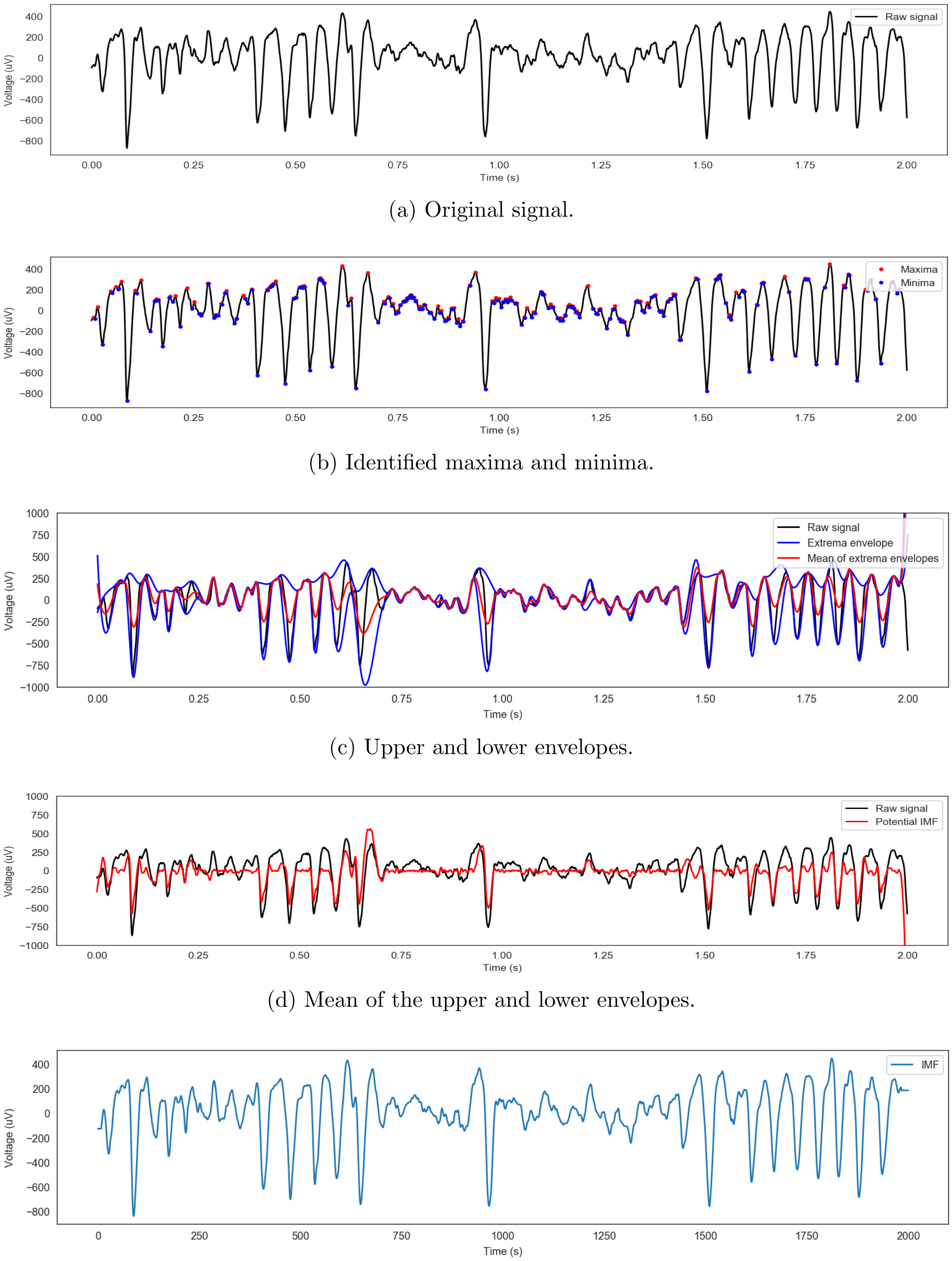

Start
Input signal: x(t)
i = 1
Form the upper and lower envelopes of x(t).
Average the upper and lower envelopes to obtain the envelope mean m(t).
Subtract the envelope mean from the signal.
h(t) = x(t) − m(t)
Is h(t) an IMF?
No
x(t) = h(t)
Yes
Store the IMF c_i(t).
c_i(t) = h(t)
Subtract c_i(t) from the signal to obtain the residual signal r(t).
r(t) = x(t) − c_i(t)
Is r(t) a monotonic function?
No
x(t) = r(t)
i = i + 1
Yes
Input signal decomposed into IMFs and a residue.
x(t) = Σ_{i=1}^{n} c_i(t) + r(t)
End

Figure 2.8: Flow chart of the EMD process (from [Jara-Maldonado et al., 2020a]).

The sifting process is illustrated in Figure 2.9, where the maxima and minima of a voltage signal are identified to later calculate their mean value and subtract it from the original signal to generate an IMF. Once the IMFs are obtained, further analysis may be performed by using the Hilbert spectral analysis, which, according to [Huang and Wu, 2008], would result in the Hilbert-Huang Transform (HHT). EMD avoids the trend removal step, which often causes low-frequency terms injection in the resulting spectra that could generate unwanted false positives.



(a) Original signal.

(b) Identified maxima and minima.

(c) Upper and lower envelopes.

(d) Mean of the upper and lower envelopes.

(e) Intrinsic Mode Function (IMF).

Figure 2.9: Empirical Mode Decomposition (EMD) sifting process applied to a voltage signal.

2.3.7. Ensemble Empirical Mode Decomposition

One of the main limitations of EMD is that it has to consider the entire data. The algorithm is inefficient for huge amounts of data and for online applications because of the high computational and memory cost. For this reason, the work in [Fontugne et al., 2017] proposes the Online EMD method, which uses a sliding window to compute the local IMFs. According to [Huang and Wu, 2008], another limitation that the EMD method presents is mode mixing. This occurs when a single IMF is composed by signals of widely disparate scales, or when a signal of a similar scale resides in different IMF components. As a solution to this problem, a method that relies on noise-assisted data analysis called Ensemble Empirical Mode Decomposition (EEMD) was proposed. In EEMD, the true IMF components are defined as the mean of an ensemble of trials that convey the signal with added white noise of finite amplitude. The process of EEMD is shown in Figure 2.10. In this figure, it is shown that this technique requires one to generate several signals by adding white Gaussian noise to the original x signal. Then, the IMFs are looked for as with the EMD technique. Finally, all the IMFs of each mode are averaged to obtain the definitive IMFs of the signal. Each IMF will represent a different mode, starting from mode one to the last mode found. The stopping criterion is the same as in the EMD process, namely when the residuum between the last IMF and the signal is a monotonic function.

The main differences between wavelets and the EMD based techniques are that EMD does not use a predefined basis, which allows the algorithm to deal with nonlinear data. Another difference is that the lack of theoretical base for EMD represents a need for a mathematical foundation that can allow researchers to make an unified general conclusion on the validity of the empirical results obtained so far by the EMD and EEMD methods; whilst the wavelet theory mathematical foundation has been deeply studied (e.g. in [Daubechies, 1992]).

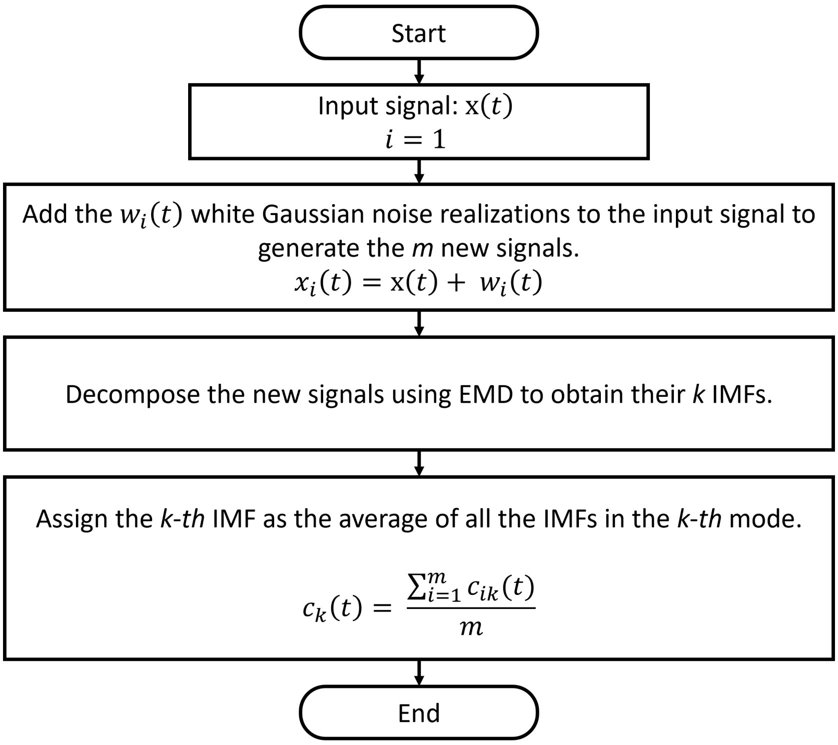

Start
Input signal: $x(t)$
$i = 1$
Add the $w_i(t)$ white Gaussian noise realizations to the input signal to generate the m new signals.
$x_i(t) = x(t) + w_i(t)$
Decompose the new signals using EMD to obtain their k IMFs.
Assign the *k-th* IMF as the average of all the IMFs in the *k-th* mode.
$c_k(t) = \frac{\sum_{i=1}^{m} c_{ik}(t)}{m}$
End

Figure 2.10: Flow chart of the Ensemble Empirical Mode Decomposition (EEMD) process (from [Jara-Maldonado et al., 2020a]).

2.4. Machine Learning Models

One of the main purposes of ML is to reduce the effort and time required to perform a certain task. By doing so, it is also possible to process the immense quantities of data that some problems, such as exoplanet identification, require. Automation leads to less resources consumption and allows humans to focus on more difficult tasks that have not yet been automated. Throughout the history of ML, many models have been proposed, each with its own advantages and limitations. Different problems may require different algorithms, though there are certain proposals that have good performance in a wide range of situations. The ML models that have been used in this work are briefly explained in this section.

The RFs algorithm perform different predictions using decision trees. These trees classify a signal by using the data points of such signal as input features. Each decision tree divides the input feature space into binary branches. The branches are created until no more divisions are possible or a threshold value has been reached. For further detail refer to [Statistics and

Breiman, 2001].

One of the most common Deep Learning (DL) algorithms is the CNN. CNNs convolve the input vector with a series of kernels to learn the local features of the input signal. In general, CNNs are composed by three mechanisms, namely the local receptive fields that organize the inputs into feature maps, kernel weight sharing among neurons, and subsampling, which is used by the CNNs to reduce the size of the features. In most cases, the final layer of a CNN is a fully-connected layer which is responsible of generating the classification result. More information about CNNs can be found in [Bishop, 2006].

The SVM algorithm maps the classes in a feature space and then searches for an optimal margin to draw a decision boundary that best separates the classes in the feature space. The margin is optimized using the hyper-parameters of the SVM, which are the soft margin constant C and the parameters of the kernel function chosen. Another algorithm that uses a decision boundary to separate the classes is the LS model. The purpose of LS is to minimize the Mean Squared Error (MSE) between the inputs and the function that traces the decision boundary. Once calculated, the decision boundary can be used to classify a new input because each side of the decision boundary corresponds to a different class. Multilayer Perceptrons or Feed Forward Neural Network (FFNN)s are based on the functioning of the human brain. It uses layers of neurons that are interconnected by synaptic links. Those links are enhanced or inhibited by an activation function (e.g. the sigmoid function) and the weights of the link. In this way, the MLP is capable of adapting its knowledge base to non-linear problems by automatically adjusting such weights. Furthermore, each layer of neurons may not have the same activation functions and weight values as the others, guaranteeing that the inputs undergo a series of transformations. The last layer of the network determines the class of the input vector. More information about SVMs, MLPs and LS can be found in [Bishop, 2006, Pearson et al., 2017, Verma et al., 2014, Haykin, 2009].

The Wavelet MLP used in [Pearson et al., 2017] applies the DWT to the inputs in order to preprocess them. The cAs and cDs obtained from the first level of decomposition of the second order Daubechies wavelet are obtained and concatenated to conform the new input vector, which is classified by an MLP.

The Ridge classifier works in a similar way to the LS classifier. It is useful for the classification of binary problems. The Ridge classifier introduces a regularization term used by the cost function of the LS algorithm. Another ML model used in this work is the KNN classifier. It clusters similar input features to determine the class to which they belong to. To classify a new input, it calculates the distance between the input and the features that have already been clustered. Once all the distances are known, the inputs are assigned to the cluster to which they were closer to. Finally, the Naïve Bayes (NB) model estimates the Probability Density Function (PDF) of each class and then calculates the probability of membership of the new inputs. For further details about the Ridge classifier, KNN and NB refer to [Theodoridis and Koutroumbas, 2008].

2.5. Discussion

Exoplanetary research has rapidly developed over the last few years. In Chapter 2, several exoplanet detection techniques have been discussed as well as the properties that each of these techniques allow one to describe. Despite the statistically low number of exoplanets that can be discovered using the transit technique (because of the geometric conditions required to observe the transits, see [Jara-Maldonado et al., 2020b]), the undeniable success of this method suggests that it has the potential to detect a wider number of exoplanets. Computational solutions are an appropriate approach to enhance this technique because they are capable of analyzing enormous quantities of data with less effort and time consumption. As it has been shown in this chapter, astronomy is not a new domain for signal processing and MRA. In [Bravo et al., 2014], for example, Wavelet Map (WVM)s are used to determine which frequencies are predominant in the signal and at which instant they exist or not. This allows one to identify several physical phenomena in stars (e.g., rotation period, changes of active regions on the star due to growth or decay of spots, etc.) from their LCs. For this reason, an ML model that takes advantage of MRA is proposed for the transit technique, which depends on analyzing signals presented as a time series of brightness values.

Chapter 3

Proposed Multiresolution Analysis Pipeline for Light Curve Preprocessing

We have created two datasets of simulated LCs containing exoplanet transits. These datasets have been used to measure the level of improvement of several ML models using MRA techniques to preprocess the LCs before using them as inputs for the ML models. The creation of the datasets is explained in this chapter, as well as the configuration of the ML models tested. Finally, we present a comparison of the results obtained with each of the MRA techniques tested. Such techniques are the DWT, EMD, and EEMD.

3.1. Datasets Creation

We created two datasets that consist of 10,000 simulated LC samples each. Half of the LCs on these datasets have simulated transits, and they have been labeled as such; while the other half does not contain transits. The LCs contain 15,000 datapoints each. Both datasets contain four types of transit models, namely the linear, uniform, quadratic, and non linear models. This allows one to diversify the geometry of the transits for a better representation of real data. To simulate the transits, we used the BAsic Transit Model cAlculatioN (BATMAN) model proposed by [Kreidberg, 2015]. This model consists of a python package that can be configured to simulate planetary transits with different parameters. An example of a simulated transit is shown in Fig. 3.1, as it can be seen in this figure, the LC produced by the BATMAN model is centered on the transit event. The ranges of the parameter values used to simulate the transits are presented in Table 3.1, which was extracted from [Jara-Maldonado et al., 2020b]. The parameters used to simulate the transits were extracted from the data of 140 real exoplanet features that were discovered with the transit method. In total, we created 560 exoplanets because we simulated each of the 140 exoplanets using the four different limb darkening models. These values were retrieved from the NASA Exoplanet Archive[1], and correspond to parameters of exoplanets reported in the Q1-Q17 Kepler Data Release 24 (*DR24*) [Coughlin et al., 2016]. Finally, Fig. 3.2 shows a histogram that represents the different durations of the simulated transits, and Fig. 3.4 shows the distribution of the simulated radii and periods.

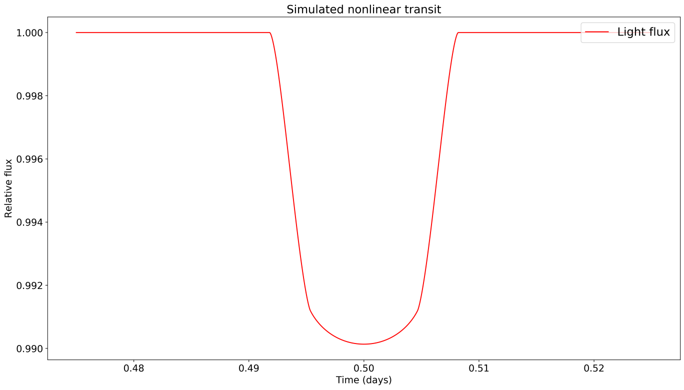


Figure 3.1: Transit simulated with the BATMAN model from [Kreidberg, 2015].

[1] NASA Exoplanet Archive: https://exoplanetarchive.ipac.caltech.edu/

Table 3.1: Transit simulation parameters from [Jara-Maldonado et al., 2020b].

Parameter Name	**Range of values**
Orbital period (P).	0.0253 – 46.69 days.
Orbit semi-major axis (a/R_*).	0.0058 – 0.2535 au.
Orbit inclination (i).	78.3 – 96.5 deg.
Planet radius (R_p).	0.063 – 1.98 Jupiter radius.
Orbit eccentricity (e).	0 – 0.53.
Stellar radius (R_*).	0.12 – 2.59 Sun radius.
Transit depth (R_p^2/R_*^2).	0.0085 – 3.23%.
Transit duration.	0.0253 – 0.4113 days.
Argument of periastron (ω).	90 deg.
Mid transit time (t_0).	75 days.
Transit resolution.	150 datapoints.
Limb darkening model.	Uniform, linear, quadratic and nonlinear.
Limb darkening coefficients (u_1, u_2, u_3, u_4).	[empty], [0.5], [0.5, 0.1], [0.5, 0.1, 0.1, -0.1].

The first dataset is called the Real-LC dataset. Its LCs are based in real LCs from the Mikulski Archive for Space Telescopes (MAST) that contained periodic events marked as non-transiting planets. Then, we de-trended the LCs by using the spline method from [Shallue and Vanderburg, 2018]. We added the simulated transits to half of the LCs. To ensure that the transits were periodic signals, we copied the transit over the LCs so that they were repeated in the same period as the one of the exoplanet that they were based on. Thus, the resulting dataset is balanced between the positive and negative labels. The method followed to create the Real-LC dataset is depicted in Figure 3.4.

The second dataset is called the 3-median dataset, and it consists of a dataset where all the data is synthetic. It was created using the LC simulation equation shown in Eq. (3.1) [Pearson et al., 2017]. This equation allows one to generate noisy LCs with quasi-periodic systematic

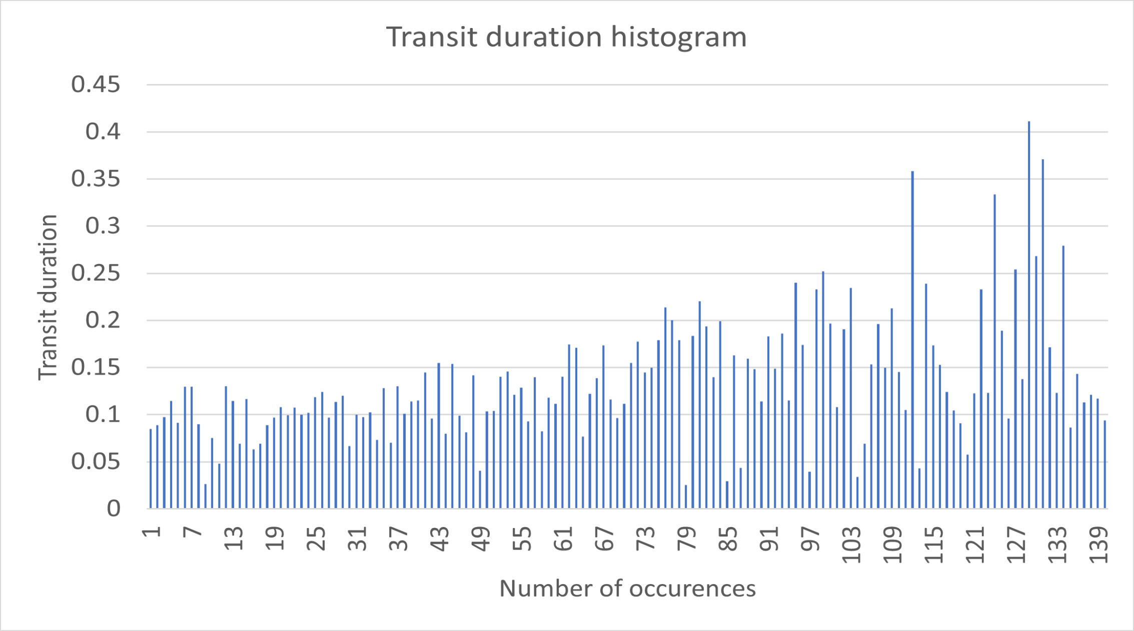


Figure 3.2: Histogram of the different transit durations used for the simulations.

trends, as it is the case of the LCs from certain space telescopes such as the *Kepler* satellite.

$$
\begin{gathered}
t' = t - t_{min} \\
A(t') = A + A \sin\left(\frac{2\,\pi\,t'}{P_A}\right) \\
\omega(t') = \omega + \omega \sin\left(\frac{2\,\pi\,t'}{P_\omega}\right) \\
F_{tr}(t) * \mathcal{N}\left(\frac{R_p^2}{R_*^2}/\sigma_{tol}\right) * \left(1 + A(t') \sin\left(\frac{2\,\pi\,t'}{\omega(t')} + \phi\right)\right)
\end{gathered}
\tag{3.1}
$$

where $F_{tr}(t)$ is the transit signal simulated using the BATMAN model from [Kreidberg, 2015], t are the time instances from the time series, t_{min} is the first time instance of the time series, σ_{tol} is the noise parameter, A is the amplitude of the simulated stellar variability, ω is the period of oscillation, ϕ is the phase shift, $\mathcal{N}$ is a Gaussian distribution with a mean of 1 and standard deviation of $(R_p^2/R_*^2)/\sigma_{tol}$, P_A and P_ω allow one to configure the frequency and amplitude of the simulated variability, and R_p^2/R_*^2 is the normalized radius ratio between the planet and the

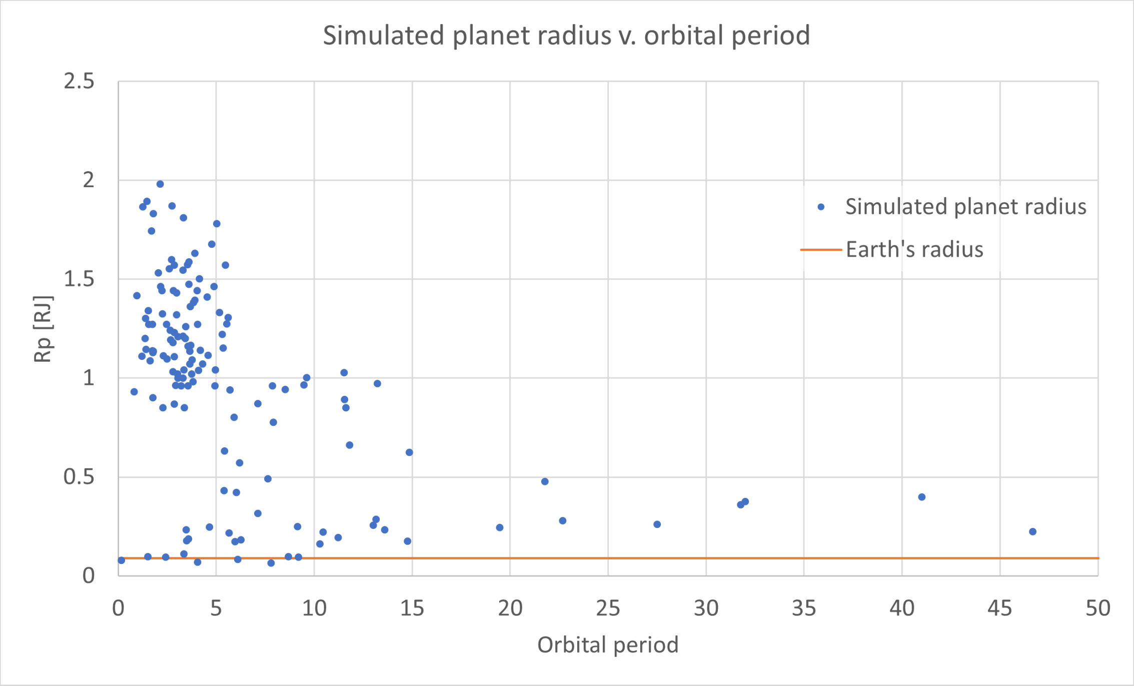


Figure 3.3: Distribution of the periods and radii of the simulated exoplanets.

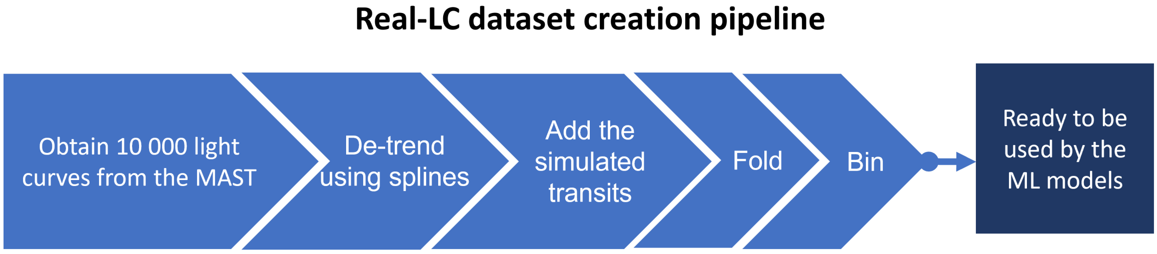

Real-LC dataset creation pipeline
Obtain 10 000 light curves from the MAST
De-trend using splines
Add the simulated transits
Fold
Bin
Ready to be used by the ML models

Figure 3.4: Real-LC dataset creation pipeline.

star. Once that the LCs were created, we added the simulated transits to half of those LCs. The other half consists of LCs without the added simulated transits. In this case, we used Eq. (3.1) without the transit signal $F_{tr}(t)$, as reported by [Pearson et al., 2017]. The parameter grid used to create the $10,000$ LCs is presented in Table 3.2 from [Jara-Maldonado et al., 2020b], which shows the parameter values used with Eq. (3.1).

Once the LCs were generated, we applied a 3 median filter to them, so the trends could be reduced, and we used the result of the filter as the final LC. The methodology followed to create the 3-median dataset is presented in Figure 3.5

Afterwards, as it can be seen in Figures 3.4 and 3.5, we folded and binned both datasets. An example of an LC before being folded and binned is shown in Figure 3.6.a. Before undergoing

Table 3.2: Noisy light curve simulation parameters from [Jara-Maldonado et al., 2020b].

Parameter Name	Range of values
Noise parameter (σ_{tol}).	$0.25, 0.75, 1.25, 1.75, 2.25, 2.75, 3, 10.$
Wave amplitude (A).	$0.025, 0.05, 0.1, 0.2.$
Wave period (ω).	$6./24, 12./24, 24./24.$
Phase offset (ϕ).	$0.$
Amplitude variability period (P_A).	$-1, 1, 100.$
Wave variability period (P_ω).	$-3, 1, 100.$
Cadence (dt).	1 minute.

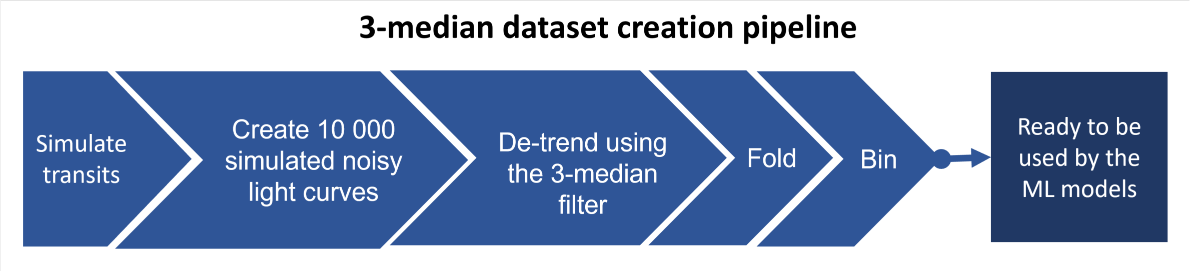
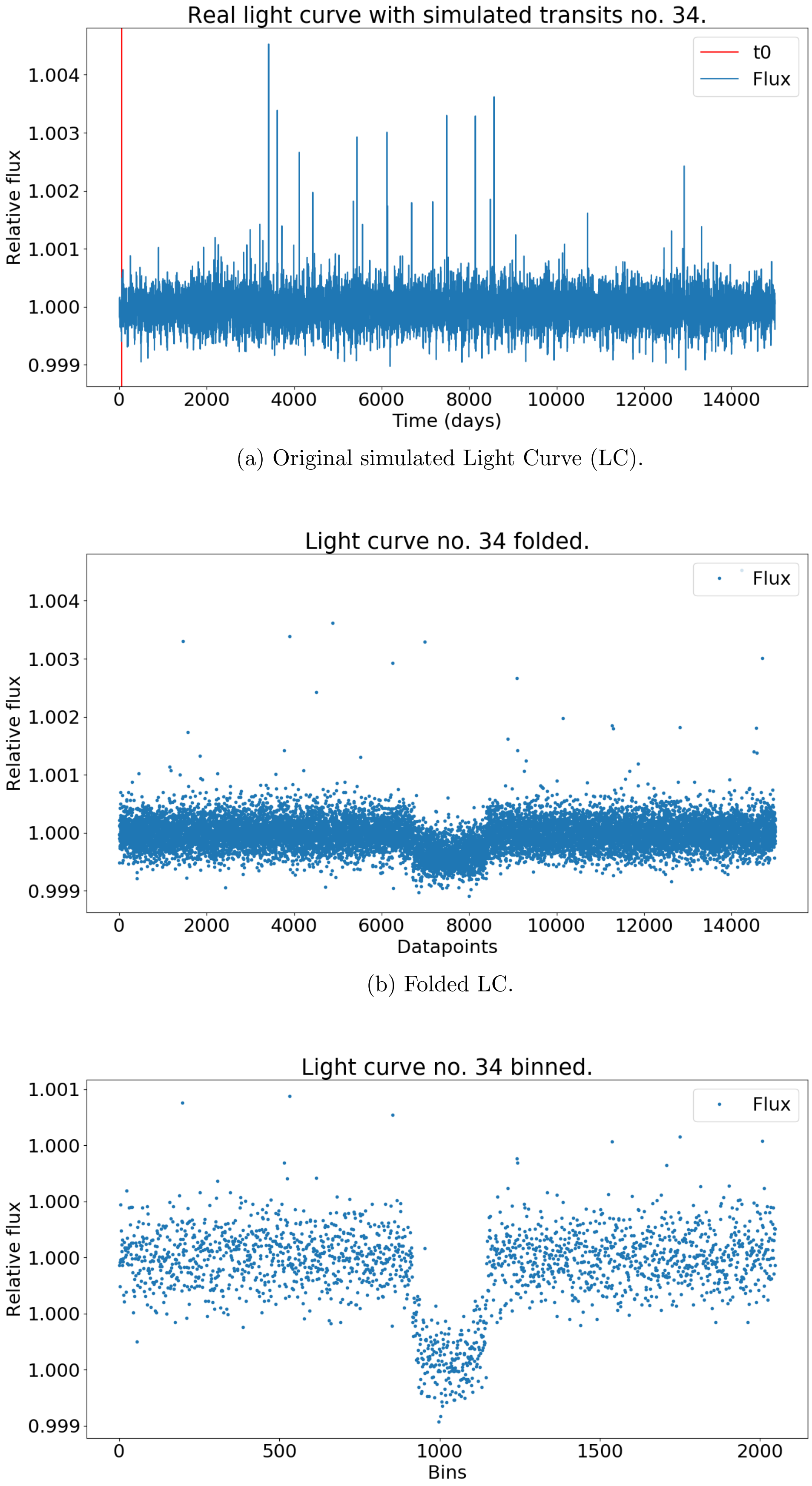

3-median dataset creation pipeline
Simulate transits
Create 10 000 simulated noisy light curves
De-trend using the 3-median filter
Fold
Bin
Ready to be used by the ML models

Figure 3.5: 3-median dataset creation pipeline.

those two steps, all the LCs have very different shapes and lengths which complicates the exoplanet identification. The phase folding step consists in overlapping all the datapoints of the LC using the transit event as the center. This enhances the possible transit signal and centers it for a better identification. An example of a folded LC is presented in Fig. 3.6.b. In some cases, the examined event can be very evident and it can be seen as a major dim in the light flux in the middle of the LC. Notice that aside from exoplanet transit events being analyzed, there could also be other sources present in the LC that could belong to another transit within the same LC. Nevertheless these other sources are not centered because they do not correspond to the event that is being analyzed in this example.

Finally, the binning step is used to reduce the dimensionality of the dataset by grouping the values in a limited number of bins. Fig. 3.7 explains the construction of one bin: the bins are created by calculating the mean of all the n points found inside a bin. We used $2,048$ bins; in other words, the final length of the LCs is $2,048$ datapoints, and each bin is then represented by the mean of all the values inside that bin. An example of a binned LC is shown in Fig. 3.6.c, which contains the same LC as the one presented in Fig. 3.6.b with the difference that



(a) Original simulated Light Curve (LC).

(b) Folded LC.

(c) Binned LC.

Figure 3.6: Example of a simulated LC.

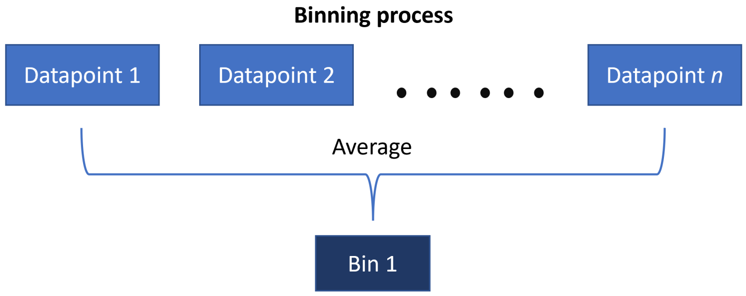


Figure 3.7: Light Curve (LC) binning process. Every datapoint contained within a bin is averaged to represent that bin.

it is now binned. As it can be observed, the number of datapoints has been reduced to $2,048$. Having the same number of datapoints in all the LCs is important because the LCs will be used as inputs for the ML models and the number of inputs to be introduced to the models has to be specified as part of the setting of the models.

3.2. Experimental Setup

We performed experiments using different MRA techniques to preprocess the LCs. The preprocessed LCs were then used as inputs for ML models and their performance was measured and compared with the results obtained without preprocessing the LCs. In total, we tested three different MRA preprocessing techniques, namely the DWT, EMD, and EEMD techniques. In this subsection we describe the configuration of the tests and the setting of the ML models used. Furthermore, all the experiments were performed using a computer with an Intel Core i7-7700 HQ CPU, 16.0 GB of RAM, Windows 10 operative system of 64 bits, and a NVIDIA GeForce GTX 1060 graphics card.

The first set of experiments consisted in using the DWT to preprocess the LCs. For these tests, we compared the performance of a CNN, LS, RF, NB, and SVM architectures and different MLP settings. The configurations of the models are presented in Tables 3.3 and 3.4, where all the parameters of each of the aforementioned models are shown respectively. In the case of the CNN and DWT, the configuration of the model depends on the number of decomposition levels because of the down-sampling process. This is because the CNN requires to have a minimum

Table 3.3: Machine learning models setup.

Machine Learning Model	**Parameters**
CNN.	▪ Batch size: 64. ▪ Number of epochs: 50. ▪ We used the Adam optimization algorithm. ▪ Step size (α): 10^{-5}. ▪ Exponential decay rates: $\beta_1 = 0.9$, $\beta_2 = 0.999$. ▪ $\epsilon = 10^{-8}$ (this is used to avoid dividing by zero during the parameters update). ▪ Loss function: categorical cross-entropy.
MLP.	▪ Learning rate: 0.001. ▪ L2 penalty parameter: $\alpha = 0.0001$. ▪ Maximum iterations: 200. ▪ Tolerance: 0.0001. ▪ Number of iterations without change: 10 (the training stops before the 200 iterations if the score is not improved by at least the tolerance value for 10 epochs). ▪ Solver function: lbfgs. ▪ The Sigmoid/Relu MLP(5, 2) have two hidden layers with five and two neural units. One uses the sigmoid function as activation (Sigmoid MLP(5, 2)), while the other uses the relu function for activation (Relu MLP(5, 2)). ▪ The Sigmoid/Relu MLP(1024) only have one hidden layer of 1024 neural units. We did this accordingly with the description of [Alarcon-Aquino and Barria, 2006], which says that the number of hidden units should be *approximately equal to half the sum of the number of input and output units* (i.e. the half of 2048 inputs and one output). ▪ The Sigmoid/Relu MLP(64, 32, 8, 1) consisted of four hidden layers with 64, 32, 8, and 1 neural units accordingly. These last two MLPs are based on the description of [Pearson et al., 2017], although the model reported in that work used a different learning rate.
LS.	Linear regression classifier without fitting the intercept.

number of inputs due to the max-pooling downsampling process. The architectures of the CNN model with DWT are shown in Figure 3.8, where it can be observed that the settings depend on the number of decomposition levels chosen. Moreover, the DWT experiments are reported in [Jara-Maldonado et al., 2020b] and are briefly described next.

Table 3.4: Machine learning models setup.

Machine Learning Model	**Parameters**
RF.	▪ Number of trees in the forest: 10 without limit of expansion (the branches could expand until becoming leaves). ▪ Split quality criteria: the Gini impurity criteria. ▪ Minimum number of samples to split a node: 2. ▪ Minimum samples on a leaf: 1. ▪ We built the trees by bootstrapping. .
NB.	Gaussian NB with no specified prior probabilities of the classes.
SVM.	▪ Regularization parameter: 1.0. ▪ Kernel: Radial base function. ▪ Scale kernel coefficient. ▪ Tolerance: 0.001. ▪ No limit of iterations.
KNN.	▪ Number of neighbors (k): 3, 5, 7, and 9. We chose these values to evaluate the performance of the model given different k values. ▪ Nearest neighbors algorithm: brute-force search. This search was used since the length of the inputs was not too large.
Ridge classifier.	▪ Regularization strength (α): 1.0. We chose this value to avoid a strong regularization, while still reducing the variance of the estimates. ▪ The intercept was computed to add flexibility to the model. ▪ No normalization was applied because the LCs were already normalized during the folding and binning steps. ▪ Both classes had the same weights to avoid prioritizing one class. ▪ The LS solver was used to compare our results with the LS results presented in [Jara-Maldonado et al., 2020b].

The DWT experiments consisted in testing the performance of the models using different wavelets, decomposition levels, and coefficients to preprocess the LCs. We compared those results with the ones obtained using the LCs without any modification of the inputs. In the case of the DWT, we used the cDs alone or the cAs alone in six different decomposition levels as inputs (from the first to the sixth level of decomposition). The wavelet selection was based in looking for a wavelet that could better represent the transit signal because, as mentioned in [Alarcon-Aquino and Barria, 2009], the wavelet should be selected based on how

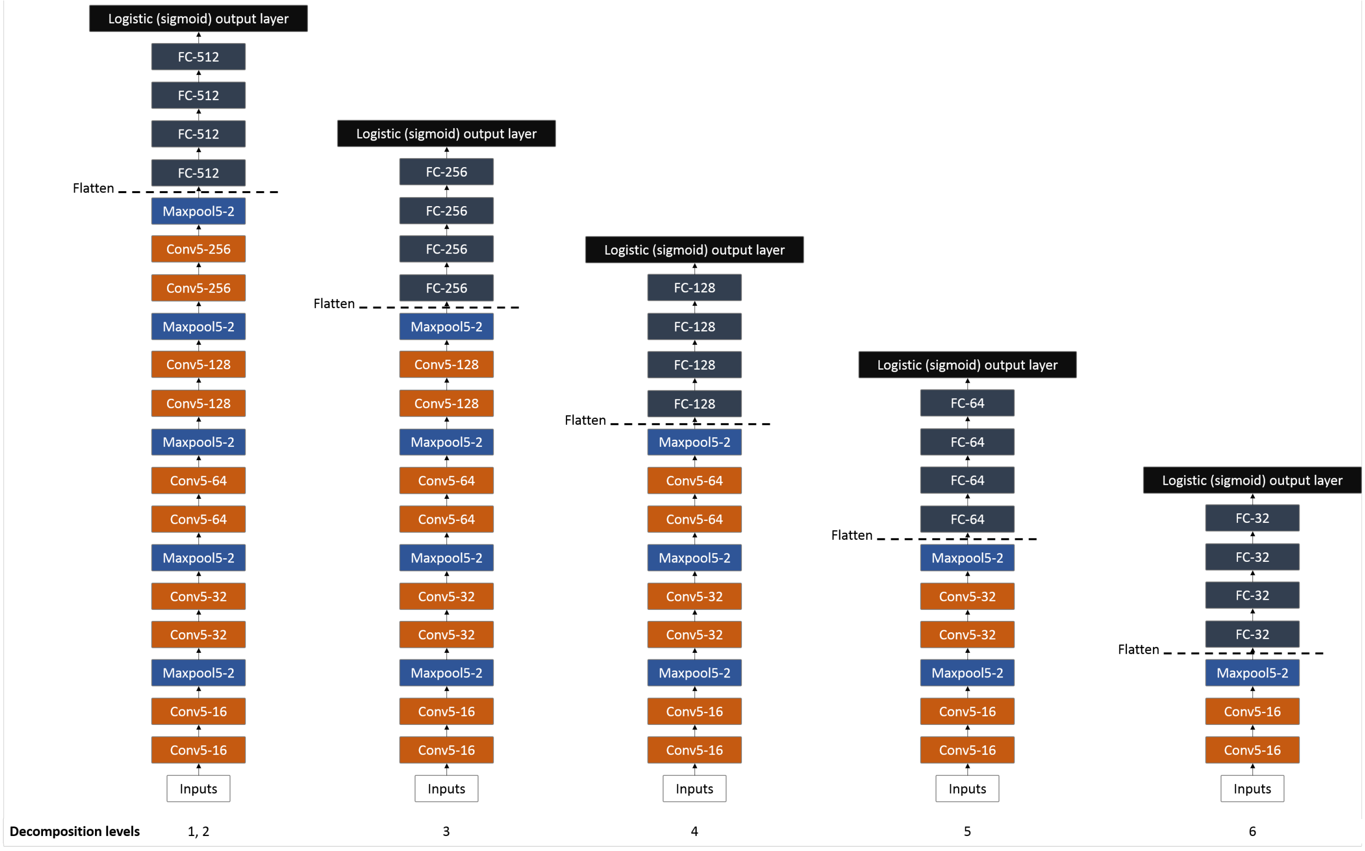


Figure 3.8: Architecture of the Convolutional Neural Network (CNN) according to each decomposition level of the Discrete Wavelet Transform (DWT). Convolutional layers are described as Conv[kernel size]-[filters], max pooling layers as Maxpool[pool size]-[strides], and Fully Connected (*FC*) layers as FC-[number of units]. The number of DWT decomposition levels is found beneath its related architecture. Figure from [Jara-Maldonado et al., 2020b].

well it adapts to the event to be analyzed. For this reason we chose wavelets with different vanishing moments. Moreover, we chose orthogonal and bi-orthogonal wavelets for comparison purposes. The wavelets used were the daubechies 1 (db1), daubechies 5 (db5), symlet 5 (sym5), coiflet 5 (coif5) and bi-orthogonal 2.4 (bior2.4) wavelets. Furthermore, we chose six different decomposition levels to evaluate if different transit signals could be present at different scales, i.e. to assess if the transits display different features at each scale. Because of the nature of the DWT, each different level of decomposition reduced the size of the time series by a half, which means that the deeper the level of decomposition the faster that the model is, but this also means that some information could be lost during the down-sampling process. The LC DWT process is shown in Figure 3.9, which indicates that the original LC signal is decomposed in cAs and cDs at each level of decomposition.

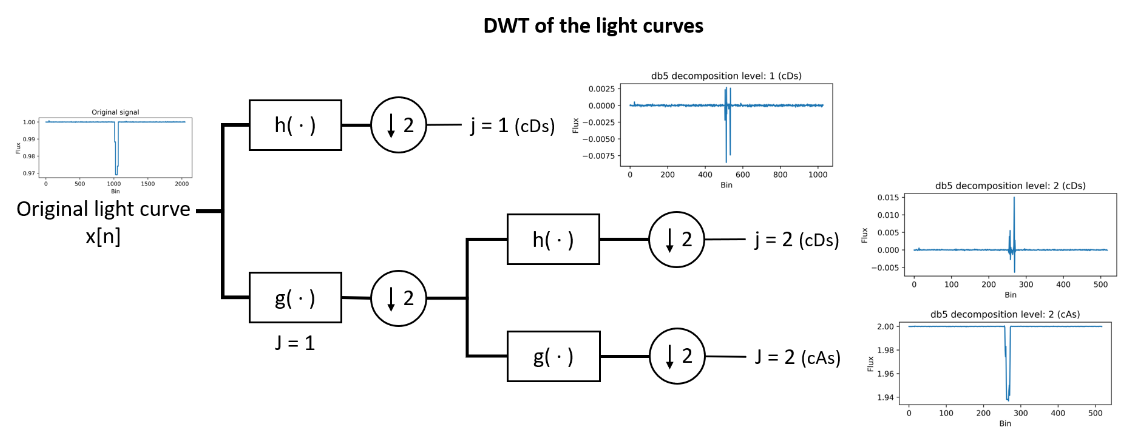


Figure 3.9: Wavelet or Detail Coefficient (cD) and scaling or Approximation Coefficient (cA) are obtained by applying a series of high-pass filters ($h(\cdot)$) and low-pass filters ($g(\cdot)$) to the original Light Curve (LC) signal $x[n]$. Also, j denotes the cDs, while J denotes the cAs; and they indicate the number of decomposition levels.

The next set of experiments consisted in using the EMD and EEMD techniques to preprocess the LCs. As in the case of the DWT, the results obtained by the preprocessed EMD and EEMD inputs and the LCs without preprocessing were compared to understand if MRA is useful for exoplanet identification. In this case, the models tested were a CNN, RF, KNN, and a Ridge classifier. The configurations of these models are also presented in Tables 3.3 and 3.4. The EMD and EEMD experiments are reported in [Jara-Maldonado et al., 2020a]. Each model presented was tested using the different IMFs obtained by applying EMD or EEMD to preprocess the LCs to use them as inputs, and those results were also compared to the results obtained by the models without applying MRA to the LCs.

For the EMD technique, we extracted all the IMFs of each LC using the *PyEMD* Python package. We tested up to 10 different modes of EMD. To test each mode, we used the nth IMF obtained from each LC (similar to what was done with the different decomposition levels of the DWT). The resulting inputs for the ML models consisted on time series of IMFs with 2048 samples. For those cases where EMD did not provide enough IMFs to correspond with the mode being tested we used the deepest IMF retrieved. For those cases we used the last IMF available. The EMD-based processing of the LCs took approximately 105 minutes for the 3-median dataset and approximately 65 minutes to preprocess the whole Real-LC dataset. An example an LC preprocessed using EMD is shown in Figure 3.10, where the plot has been centered in the transit of the simulated exoplanet.

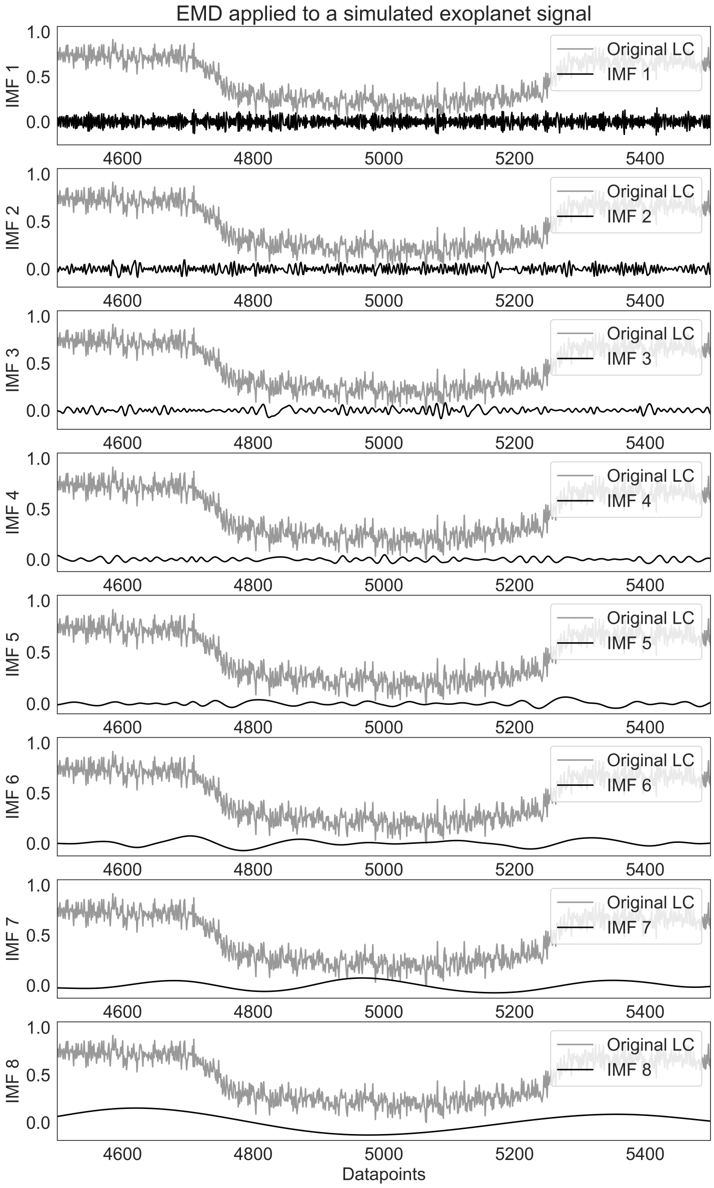


Figure 3.10: Normalized Intrinsic Mode Function (IMF) of the first eight modes of the Empirical Mode Decomposition (EMD) technique applied to a simulated exoplanet transit. The Light Curve (LC) has been centered on the transit for a better visualization.

In the case of the EEMD technique, we used the EEMD Matlab implementation presented in [Torres et al., 2011][2] to obtain the EEMD IMFs of the LCs. In the same way as it was done with EMD, we decomposed each LC with 1,000 iterations. Also, we set the number of realizations to 500 (which is the number of realizations of Gaussian white noise added to the data. This number of realizations is based on the configuration used in [Torres et al., 2011]). We used the IMFs to test the performance of the models with the Python programming language. Similar

[2] EEMD Matlab function `https://github.com/ron1818/PhD_code/tree/master/EMD_EEMD`

to what happened with the EMD, not every LC produced enough IMFs to correspond with the mode being analyzed. For those cases we also used the last IMF available. Preprocessing the LCs with EEMD took approximately 238 minutes with the 3-median dataset and approximately 139 minutes with the Real-LC dataset. An example an LC preprocessed using EEMD is shown in Figure 3.11.

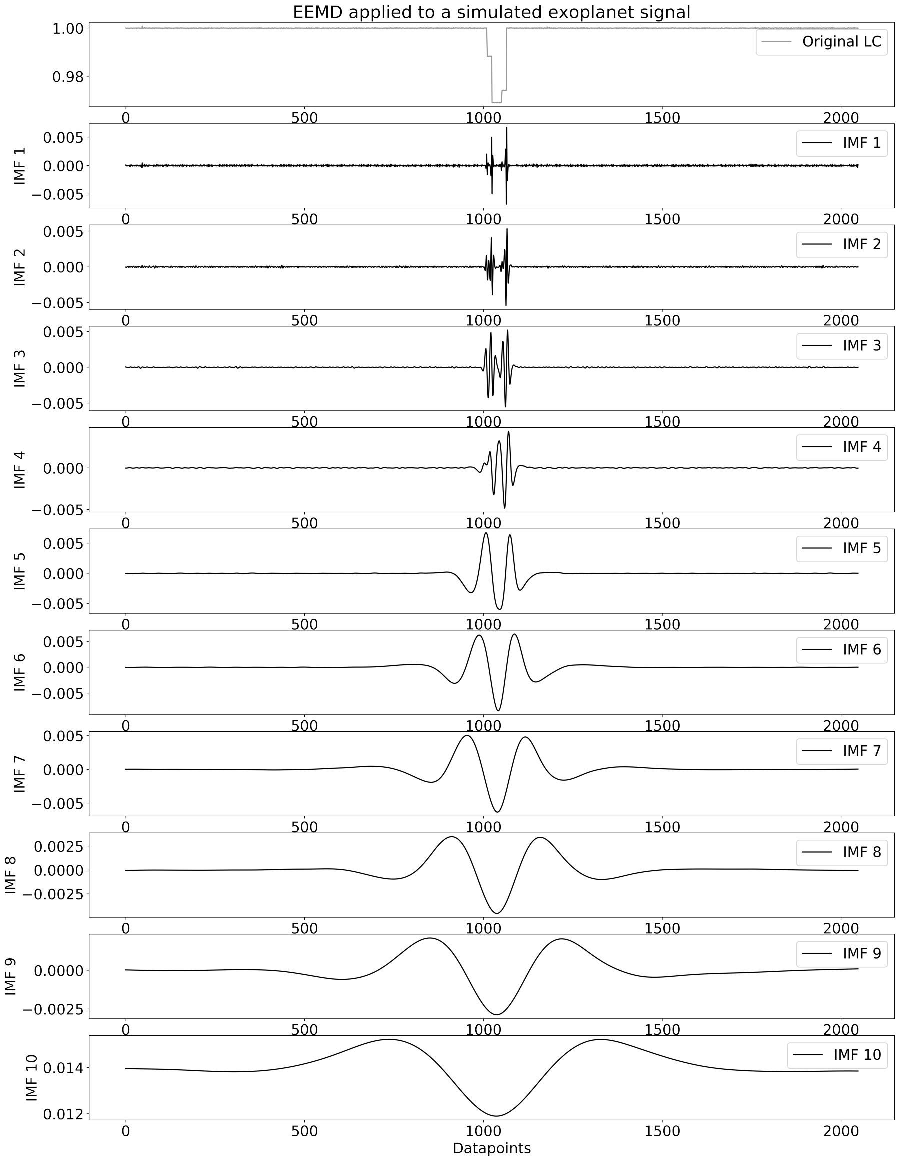


Figure 3.11: Intrinsic Mode Function (IMF) of the first 10 modes of the Ensemble Empirical Mode Decomposition (EEMD) technique applied to a simulated exoplanet transit.

Finally, the pipeline followed to perform the sets of experiments described in this section is presented in Figure 3.12. As it can be observed, the LCs were subjected to one of the different options of preprocessing or none of them. Then, the resulting time series were used as inputs for the ML models which output an identification answer. Such outputs were evaluated and compared to determine if the MRA technique was useful and which setting provides the best results.

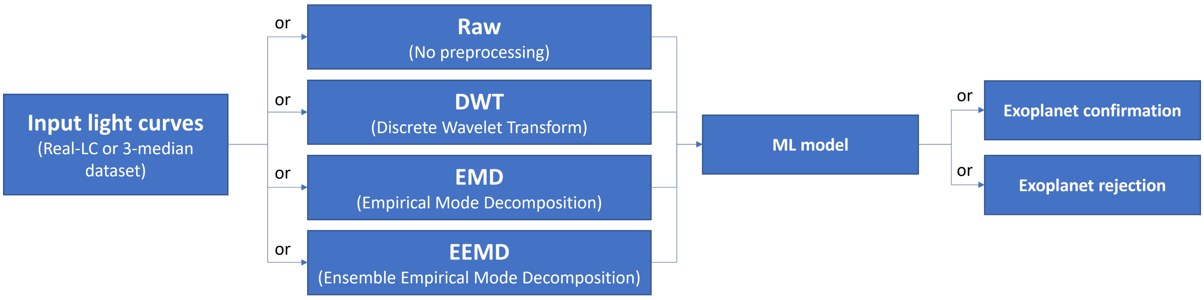

Input light curves
(Real-LC or 3-median dataset)
or
Raw
(No preprocessing)
or
DWT
(Discrete Wavelet Transform)
or
EMD
(Empirical Mode Decomposition)
or
EEMD
(Ensemble Empirical Mode Decomposition)
ML model
or
Exoplanet confirmation
or
Exoplanet rejection

Figure 3.12: Pipeline of the experiments performed with the Machine Learning (ML) models and Multiresolution Analysis (MRA) for preprocessing the inputs.

3.3. Experimental Results

In order to measure the performance of each tested model, we compared the models in terms of their accuracy, precision, recall, specificity and execution time. This metrics are based on the number of correctly classified exoplanets (True Positives (TP)), correctly classified non-exoplanets (True Negatives (TN)), misclassified exoplanets (False Positives (FP)), and misclassified non-exoplanets (False Negatives (FN)). The accuracy measures how many times the model was correct, the precision measures how often a positive answer was correct, the recall measures the percentage of real positive answers that were classified as such, and the specificity measures the percentage of real negative answers that were classified as such (see Equations (3.2), (3.3), (3.4), (3.5) from [Japkowicz and Shah, 2011]).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.5}$$

The results of the Real-LC dataset using the DWT to preprocess the LCs from the Real-LC dataset are presented in Table 3.5, while the results from the 3-median dataset are presented in Table 3.6, which was extracted from [Jara-Maldonado et al., 2020b]. Then, the results of using the EMD and EEMD techniques to preprocess the LCs from the Real-LC dataset are shown in Table 3.7, while the results of these two techniques applied to the 3-median dataset are presented in Table 3.8. For every table, the models are ordered from top to bottom according to the accuracy percentages that they obtained. Also, the best values of the tables have been highlighted using bold characters. In the case of the CNN, the model was trained only once and then tested 100 times. This was done so because of the long training times of the model. The rest of the models were trained 100 times and tested after each train and that is the time that is reported in the tables. The results presented in the tables correspond to the settings that obtained the best values for each ML model and the complete lists of results (e.g. all the decomposition levels and coefficients for each wavelet tested using DWT) are presented in Appendices B and C.

A visual comparison of the accuracy obtained by the models with and without using the DWT is presented in Figures 3.13 and 3.14. The blue bars show the results obtained without using the DWT, and the orange one show the results obtained using it. It is noticeable that in most cases the accuracy is increased, or at least it does not decrease. Then, in Figures 3.15 and 3.16, the execution time results are presented. As it can be seen, the execution times are always reduced, and this is due to the down-sampling property of the DWT. In each level of decomposition, the signal length is halved.

The visual comparison of the accuracy results using the EMD technique and its ensemble variant are presented in Figures 3.17 and 3.18. The blue bars show the signal without MRA preprocessing. The orange bars show the results obtained using EMD, and the gray bars show the results obtained using EEMD. Finally, Figures 3.19 and 3.20 show the execution times for these techniques. These figures demonstrate that in most cases, using EMD or EEMD increases the performance of the identification models, both in time and accuracy. The only model where the execution time is significantly affected by these techniques is the CNN. We attribute this to the fact that the data obtained several decimal positions after the sifting processes introduce calculation errors.

Table 3.5: Real-LC dataset experimental results, averaged from 100 executions of the training and testing processes for each ML model. The ML model inputs are based on using the DWT coefficients with several wavelets and decomposition levels, and without applying DWT to the binned light curves from [Jara-Maldonado et al., 2020b].

ML model	**Accuracy (%)**	**Precision (%)**	**Specificity (%)**	**Recall (%)**	**Time (seconds)**
Random Forests.					
Inputs without DWT.	97.91	98.35	98.37	97.45	10.26
Inputs using db1 cAs ($J = 4$).	**98.50**	**98.59**	**98.59**	**98.41**	**1.18**
CNN.					
Inputs without DWT.	91.46	97.55	91.46	85.21	46.74
Inputs using coif5 cAs ($J = 2$).	**94.28**	**98.15**	**94.28**	**90.16**	**31.59**
SVM.					
Inputs without DWT.	88.67	**99.34**	**99.49**	77.81	60.87
Inputs using db1 cDs ($j = 6$).	**93.06**	98.56	98.72	**87.4**	**0.78**
Sigmoid MLP(5, 2).					
Inputs without DWT.	49.65	28.83	52.0	48.39	4.82
Inputs using db5 cDs ($j = 6$).	**92.0**	**94.76**	**94.92**	**89.08**	**0.67**
Relu MLP(1024).					
Inputs without DWT.	81.63	93.19	87.02	76.23	84.44
Inputs using sym5 cDs ($j = 6$).	**89.12**	**97.63**	**97.93**	**80.36**	**30.06**
Sigmoid MLP(1024).					
Inputs without DWT.	88.73	98.85	**98.48**	**79.0**	49.57
Inputs using coif5 cAs ($J = 2$).	**88.79**	**99.18**	98.33	78.27	**24.14**
Relu MLP(5, 2).					
Inputs without DWT.	49.54	21.26	57.0	43.0	5.75
Inputs using db5 cDs ($j = 6$).	**83.87**	**98.84**	**99.16**	**68.55**	**0.31**
Relu MLP(64, 32, 8, 1).					
Inputs without DWT.	76.98	86.71	**88.56**	65.48	15.42
Inputs using db5 cAs ($J = 2$).	**77.88**	**93.54**	87.81	**67.87**	**6.81**
Naive Bayes.					
Inputs without DWT.	53.78	70.28	86.11	21.76	6.17
Inputs using db1 cDs ($j = 6$).	**77.09**	**94.16**	**92.53**	**61.64**	**0.16**
LS.					
Inputs without DWT.	**65.16**	94.26	98.04	**32.23**	8.51
Inputs using bior2.4 cAs ($J = 2$).	62.32	**98.08**	**99.50**	25.21	**1.8**
Sigmoid MLP(64, 32, 8, 1).					
Inputs without DWT.	49.49	**30.2**	41.0	**59.0**	5.05
Inputs using db5 cDs ($j = 2$).	**49.5**	27.7	**44.0**	56.0	**1.41**

Table 3.6: 3-median dataset experimental results, averaged from 100 executions of the training and testing processes for each ML model. The ML model inputs are based on using the DWT coefficients with several wavelets and decomposition levels, and without applying DWT to the binned light curves from [Jara-Maldonado et al., 2020b].

ML model	**Accuracy (%)**	**Precision (%)**	**Specificity (%)**	**Recall (%)**	**Time (seconds)**
CNN.					
Inputs without DWT.	97.68	**99.94**	97.68	95.48	54.17
Inputs using sym5 cDs ($j = 1$).	**99.13**	99.16	**99.13**	**99.09**	**22.93**
Random Forests.					
Inputs without DWT.	97.82	97.25	97.17	98.45	9.42
Inputs using db1 cAs ($J = 4$).	**98.08**	**97.49**	**97.41**	**98.73**	**1.16**
Relu MLP(64, 32, 8, 1).					
Inputs without DWT.	79.92	72.81	59.48	**99.96**	16.7
Inputs using sym5 cDs ($j = 4$).	**97.48**	**96.66**	**96.57**	98.38	**7.16**
Relu MLP(1024).					
Inputs without DWT.	93.31	88.95	87.39	**99.14**	241.85
Inputs using sym5 cDs ($j = 4$).	**97.47**	**96.32**	**96.17**	98.75	**72.93**
Naive Bayes.					
Inputs without DWT.	94.75	90.81	92.42	**99.67**	8.8
Inputs using bior2.4 cDs ($j = 6$).	**95.95**	**93.37**	**92.85**	98.99	**0.35**
SVM.					
Inputs without DWT.	93.76	88.99	87.41	**100**	36.31
Inputs using sym5 cDs ($j = 5$).	**94.96**	**90.93**	**89.84**	99.99	**1.52**
Relu MLP(5, 2).					
Inputs without DWT.	49.98	42.67	15.0	85.0	7.65
Inputs using bior2.4 cDs ($j = 3$).	**95.12**	**92.94**	**92.27**	**97.93**	**2.54**
Sigmoid MLP(5, 2).					
Inputs without DWT.	84.64	77.43	76.31	92.79	12.42
Inputs using coif5 cDs ($j = 4$).	**94.23**	**92.38**	**91.69**	**96.72**	**1.76**
Sigmoid MLP(1024).					
Inputs without DWT.	86.04	79.85	72.87	**98.93**	166.42
Inputs using bior2.4 cAs ($J = 6$).	**92.65**	**89.72**	**88.13**	97.1	**46.94**
Sigmoid MLP(64, 32, 8, 1).					
Inputs without DWT.	50.01	**41.24**	18.06	**82.0**	7.86
Inputs using coif5 cAs ($J = 6$).	**50.68**	39.66	**23.51**	77.98	**0.43**
LS.					
Inputs without DWT.	37.99	**13.65**	72.22	**4.34**	10.57
Inputs using sym5 cDs ($j = 6$).	**49.62**	0	**99.89**	0	**0.35**

Table 3.7: Experimental results from the Real-LC dataset. Each value corresponds to the average of the 100 test iterations for each model from [Jara-Maldonado et al., 2020a].

Model	**Accuracy (%)**	**Precision (%)**	**Specificity (%)**	**Recall (%)**	**Time (secs.)**
KNN ($k = 3$).					
Without MRA.	97.92	**98.92**	**98.94**	96.90	8.29
Using EMD and 7 IMFs.	97.74	98.07	98.09	**97.38**	11.79
Using EEMD and 6 IMFs.	**98.04**	98.81	98.83	97.25	**5.88**
KNN ($k = 5$).					
Without MRA.	97.59	**98.64**	**98.67**	96.51	6.72
Using EMD and 7 IMFs.	97.26	97.45	97.45	97.08	11.96
Using EEMD and 6 IMFs.	**97.87**	98.60	98.63	**97.10**	**5.93**
KNN ($k = 7$).					
Without MRA.	97.29	98.38	98.41	96.18	7.53
Using EMD and 7 IMFs.	97.10	97.33	97.33	96.86	11.87
Using EEMD and 6 IMFs.	**97.82**	**98.48**	**98.50**	**97.14**	**5.91**
KNN ($k = 9$).					
Without MRA.	97.01	98.25	98.29	95.72	10.80
Using EMD and 7 IMFs.	97.00	97.34	97.39	96.60	11.95
Using EEMD and 6 IMFs.	**97.71**	**98.31**	**98.34**	**97.08**	**5.94**
Ridge Classifier.					
Without MRA.	85.19	**99.79**	**99.85**	70.60	6.21
Using EMD and 6 IMFs.	**88.32**	92.07	92.79	83.84	10.18
Using EEMD and 9 IMFs.	88.05	85.86	84.95	**91.14**	**5.13**
CNN.					
Without MRA.	91.46	97.55	91.46	85.21	46.74
Using EMD and 5 IMFs.	91.67	**99.16**	91.67	83.98	135.89
Using EEMD and 6 IMFs.	**97.13**	98.9	**97.15**	**95.32**	**44.32**
RF.					
Without MRA.	97.91	98.35	98.37	97.45	10.26
Using EMD and 7 IMFs.	**98.43**	**98.8**	**98.81**	**98.04**	13.41
Using EEMD and 6 IMFs.	98.17	98.48	98.49	97.84	**6.19**

Note: The best values for each dataset have been highlighted using bold characters.

Table 3.8: Experimental results from the 3-median dataset. Each value corresponds to the average of the 100 test iterations for each model from [Jara-Maldonado et al., 2020a].

Model	**Accuracy**	**Precision**	**Specificity**	**Recall**	**Time**
	(%)	(%)	(%)	(%)	**(secs.)**
KNN ($k = 3$).					
Without MRA.	64.91	58.99	29.15	**100**	11.02
Using EMD and 9 IMFs.	93.74	**90.80**	**89.97**	97.45	11.71
Using EEMD and 7 IMFs.	**93.82**	89.70	88.43	99.12	**10.48**
KNN ($k = 5$).					
Without MRA.	63.74	58.09	27.09	**100**	11.30
Using EMD and 9 IMFs.	**93.75**	**90.27**	**89.27**	98.17	11.96
Using EEMD and 7 IMFs.	93.21	88.72	87.18	99.14	**10.50**
KNN ($k = 7$).					
Without MRA.	63.13	57.72	25.74	**100**	11.43
Using EMD and 9 IMFs.	**93.61**	**89.84**	**88.66**	98.47	11.94
Using EEMD and 7 IMFs.	92.97	88.35	86.72	99.13	**8.87**
KNN ($k = 9$).					
Without MRA.	62.51	57.29	24.62	**100**	9.44
Using EMD and 9 IMFs.	**93.60**	**89.69**	**88.45**	98.65	11.90
Using EEMD and 7 IMFs.	92.63	87.80	85.98	99.16	**8.62**
Ridge Classifier.					
Without MRA.	76.02	67.79	51.81	**99.87**	8.3
Using EMD and 6 IMFs.	76.33	68.34	53.46	98.86	10.47
Using EEMD and 9 IMFs.	**79.26**	**91.81**	**94.12**	64.68	**5.27**
CNN.					
Without MRA.	97.68	**99.94**	97.68	95.48	**54.17**
Using EMD and 1 IMFs.	**99.32**	99.67	**99.32**	**98.99**	100.34
Using EEMD and 5 IMFs.	95.39	94.43	95.39	96.6	161.93
RF.					
Without MRA.	**97.82**	**97.25**	**97.17**	**98.45**	9.42
Using EMD and 1 IMFs.	97.57	96.84	96.73	98.40	13.97
Using EEMD and 6 IMFs.	97.35	96.42	96.28	98.40	**6.11**

Note: The best values for each dataset have been highlighted using bold characters.

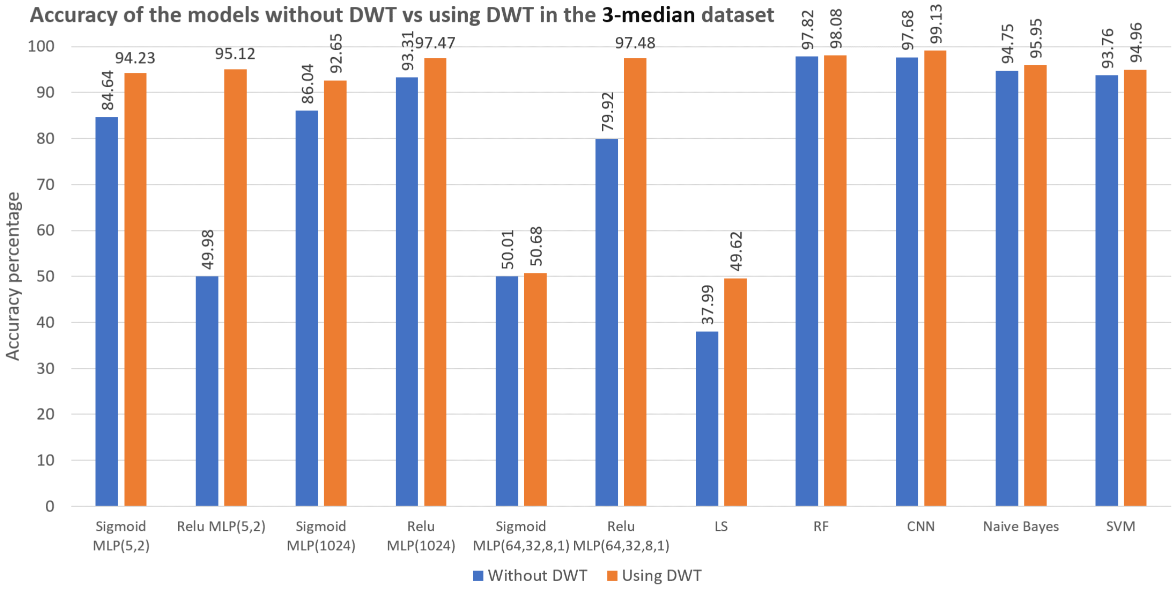


Figure 3.13: Accuracy obtained preprocessing every Light Curve (LC) from the 3-median dataset using the Discrete Wavelet Transform (DWT) technique. Figure published in [Hernndez and Meneses, 2022].

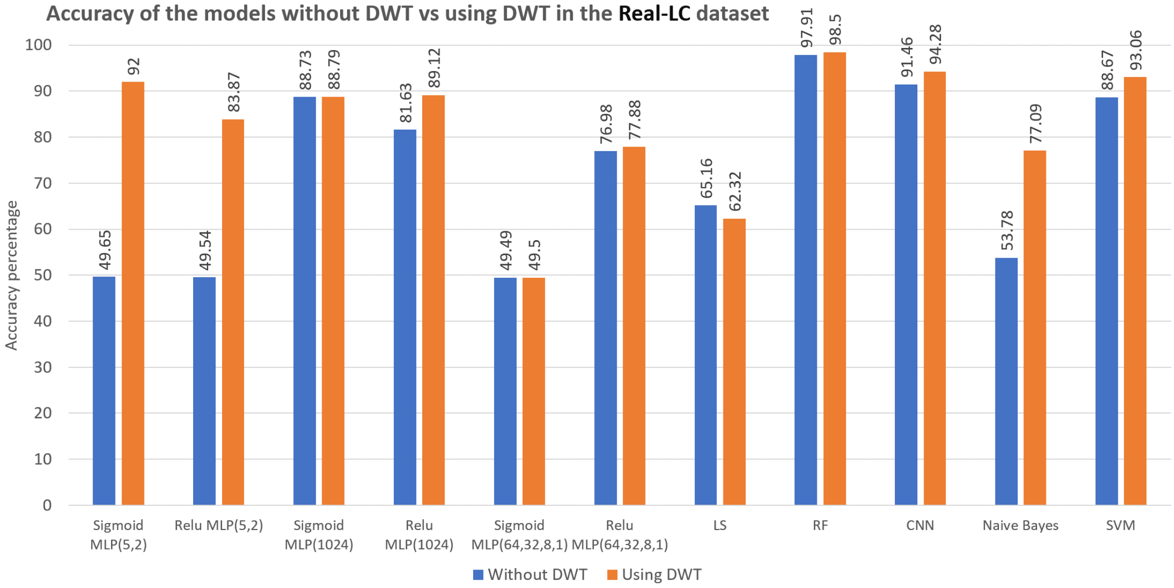


Figure 3.14: Accuracy obtained preprocessing every Light Curve (LC) from the Real-LC dataset using the Discrete Wavelet Transform (DWT) technique. Figure published in [Hernndez and Meneses, 2022].

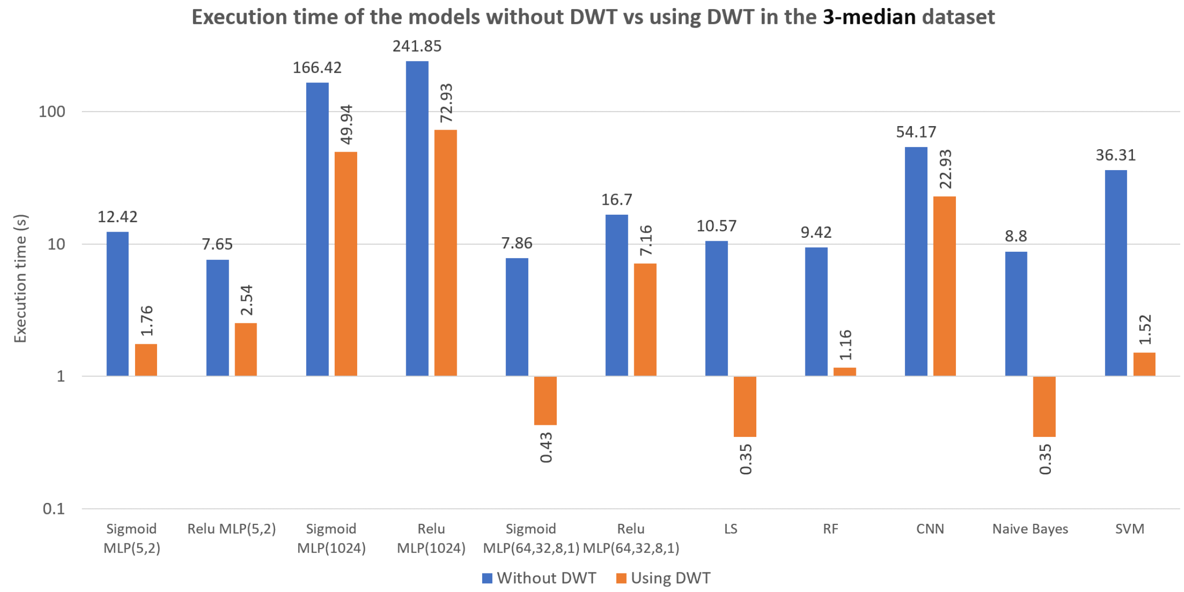


Figure 3.15: Execution time obtained preprocessing every Light Curve (LC) from the 3-median dataset using the Discrete Wavelet Transform (DWT) technique. Figure published in [Hernndez and Meneses, 2022].

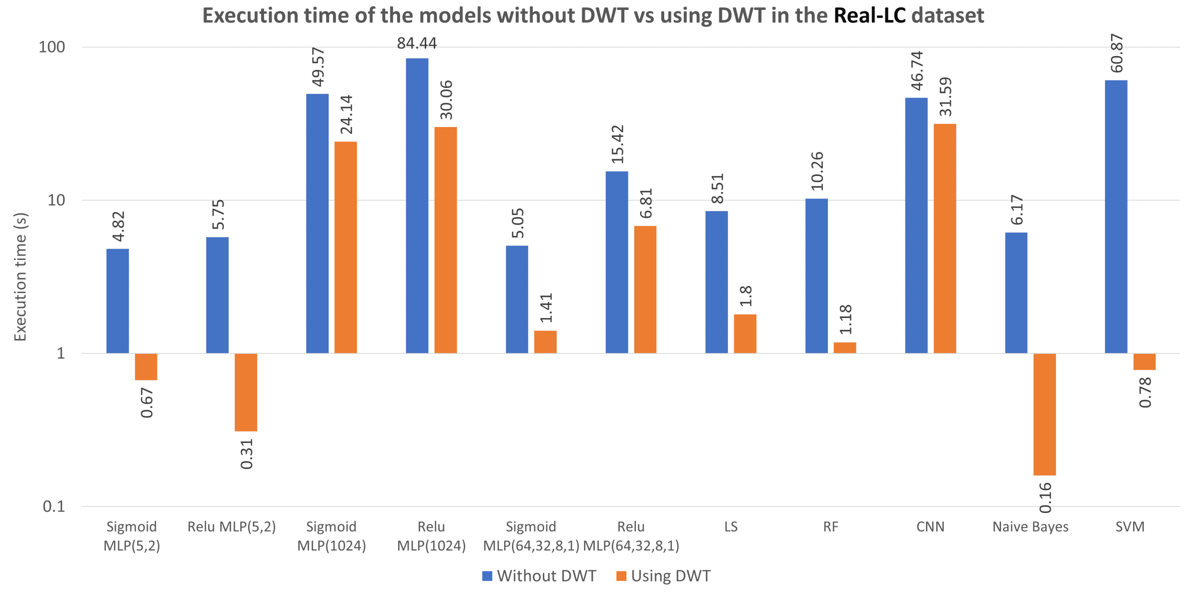


Figure 3.16: Execution time obtained preprocessing every Light Curve (LC) from the Real-LC dataset using the Discrete Wavelet Transform (DWT) technique. Figure published in [Hernndez and Meneses, 2022].

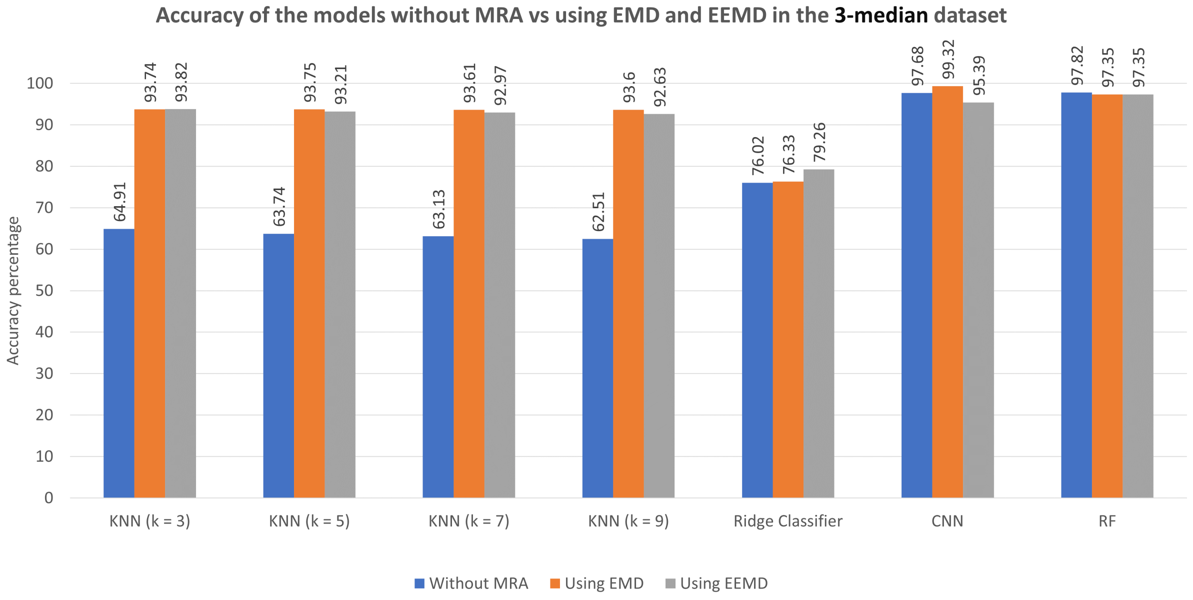


Figure 3.17: Accuracy obtained preprocessing every Light Curve (LC) from the 3-median dataset using the Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) techniques. Figure published in [Hernndez and Meneses, 2022].

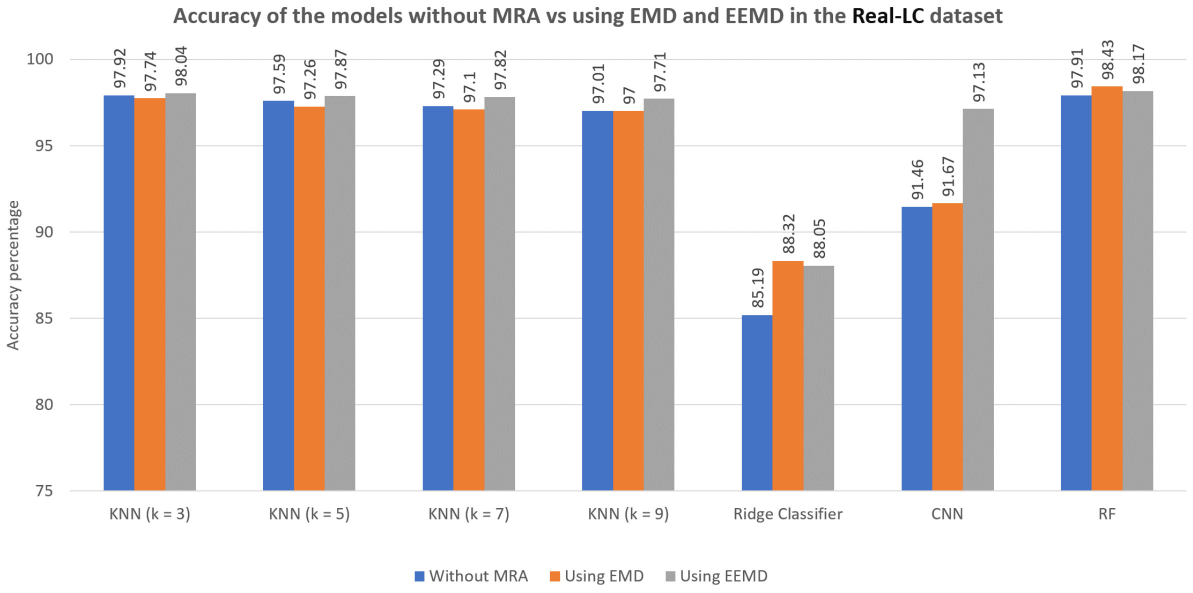


Figure 3.18: Accuracy obtained preprocessing every Light Curve (LC) from the Real-LC dataset using the Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) techniques. Figure published in [Hernndez and Meneses, 2022].

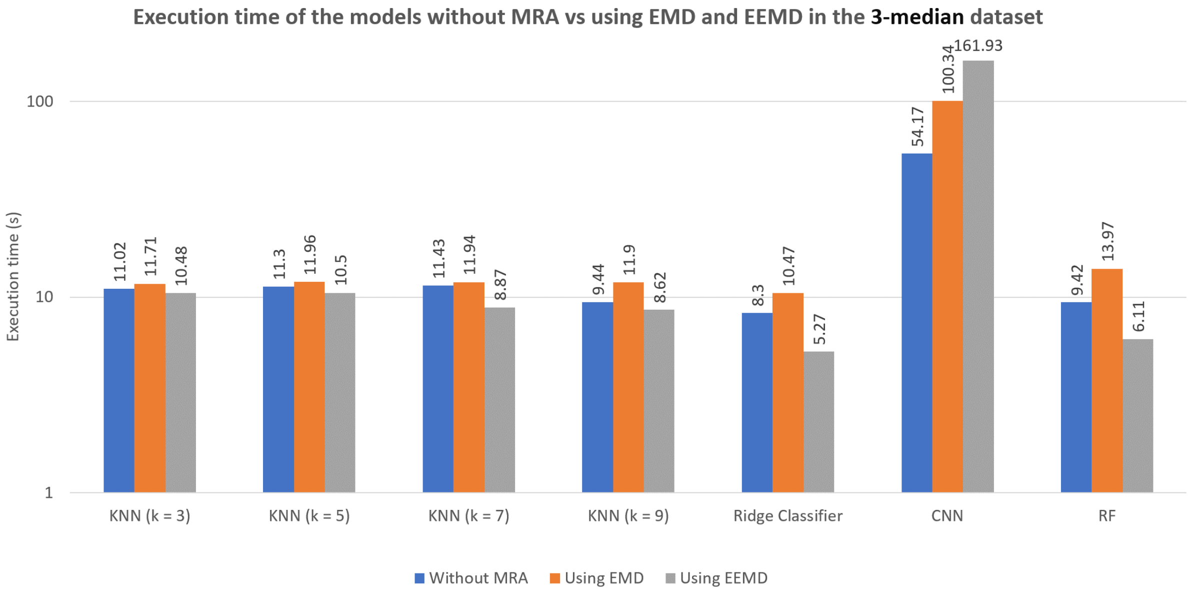


Figure 3.19: Execution time obtained preprocessing every Light Curve (LC) from the 3-median dataset using the Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) techniques. Figure published in [Hernndez and Meneses, 2022].

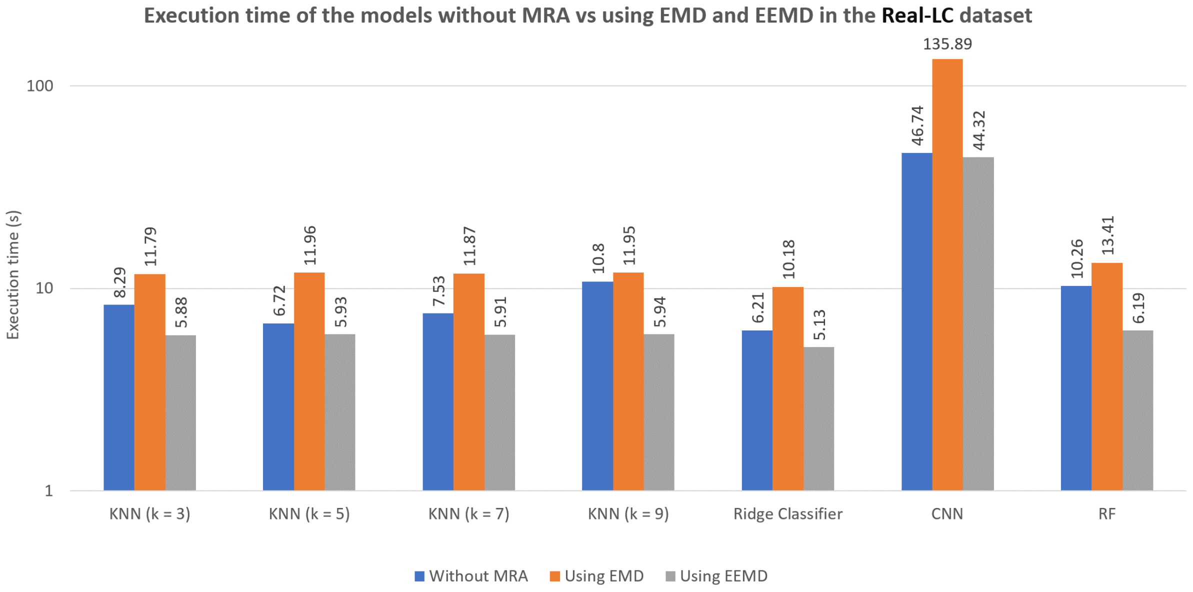


Figure 3.20: Execution time obtained preprocessing every Light Curve (LC) from the Real-LC dataset using the Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) techniques. Figure published in [Hernndez and Meneses, 2022].

3.4. Discussion

Several ML models used for exoplanet identification have been tested in this chapter. The experiments were performed using the 3-median and Real-LC datasets of simulated LCs. The sets of experiments consisted in using the LCs as inputs for the models or preprocessing the LCs before inputting them to the models using an MRA technique, namely the DWT, EMD or EEMD techniques.

In the case of the results of the 3-median dataset, shown in Tables 3.6 and 3.8, the best performing models are two configurations of the CNN model (using MRA preprocessing). In the case of the 3-median dataset, the best accuracy result was obtained with the CNN-EMD model, which is 99.32%. In fact, using EMD helped the CNN to improve in almost every metric. Another setting that considerably improved the results attained by the CNN is the DWT preprocessing technique. Using this configuration, the CNN obtained and accuracy of 99.13%, but the most remarkable improvement is that the execution time was reduced by more than a half. This was expected because of the decimation process. Nevertheless, only one level of decomposition was used, so the architecture is the same as the one presented by [Shallue and Vanderburg, 2018].

It is observed that in most of the cases, the RF classifier is one of the most robust models because it obtained some of the best results with both datasets. For instance, the results in the Real-LC dataset, which are depicted in Tables 3.5 and 3.7, show that the best performing model is the RF classifier. In fact, the top three results of the Real-LC dataset are obtained with the RF classifier in combination with each of the three MRA techniques tested. The execution time of the RF model considerably decreases by using the DWT, making it the fastest model tested in this work. Moreover, in some cases the DWT was capable of reducing the execution time of the models to a 10th of their original values. This is because the DWT down-samples the signal, which means that for each decomposition level that the signal undergoes, the length of the signal is reduced by half. Such reduction does not affect much the accuracy results because the DWT extracts the most relevant features of the signal. In general, most models benefited from the DWT, for example, the sigmoid MLP(5,2) with the Real-LC obtained an accuracy improvement of +42.35%. That is the same reason to which we attribute the good results of the RF-DWT classifier, because the performance of such model depends on the significance of the inputs to represent the signal.

The KNN model obtained very good accuracy results with the Real-LC dataset, specially with the $k = 3$ setting. Having a greater number of neighbors may cause the model to average important data from the transit, decreasing its accuracy. Also, it is noticeable that the execution time of the KNN is not particularly affected by the MRA technique used. Nevertheless, the KNN model obtained poor results without LC preprocessing in the 3-median dataset. For this dataset, the obtained accuracy was lower than 70%. This was caused by the low specificity of the model, meaning that it obtained many FNs. This could be caused by the noise of the signal. In the 3-median dataset, the transit shape could be modified by noise, causing irregularities in its geometry which are then averaged by the KNN model. The use of MRA does improve the performance of this model with the 3-median dataset (it gave the model an improvement of almost +30% in all cases), but the results are not as good as the ones obtained with the Real-LC dataset. This means that MRA improved the performance of the model, making it more robust independently of the dataset that it was used with.

Regarding the experiments with the EMD and EEMD techniques to preprocess the LCs, it can be observed that the execution time of the models is barely improved or worsen. Also, even though the LCs only have to be preprocessed once, adding the EMD or EEMD preprocessing steps involves consuming more execution time. As mentioned earlier, preprocessing the LCs with the EMD technique took 105 minutes with the 3-median dataset and 65 minutes with the Real-LC dataset, and the EEMD technique took 238 minutes with the 3-median dataset and 139 minutes with the Real-LC dataset.

Finally, we performed hypothesis tests to statistically validate that the results obtained using MRA to preprocess the LCs are different from those obtained by the models without preprocessing the LCs. For this purpose, we have used the Welch's t-test to compare the results. This test is used to determine if there is a significative difference between the means of the results. A p-value lower than a threshold which typically is 0.05 (e.g. [Committee, 2011]) is enough to statistically guarantee that the results are meaningfully different. The results of such tests are presented in Table 3.9, where only those cases where both results were equal are presented, along with the metric in which they are equal and the p-value obtained.

Table 3.9: p-values obtained from the Welch's t-tests where the p-value is greater than 0.05.

MRA technique	**Dataset**	**Model**	**Metric**	**p-value**
DWT	3-median	Sigmoid MLP(5, 2).	Recall	0.069
DWT	3-median	Sigmoid MLP(64, 32, 8, 1).	Accuracy	0.0504
DWT	3-median	Sigmoid MLP(64, 32, 8, 1).	Precision	0.58
DWT	3-median	Sigmoid MLP(64, 32, 8, 1).	Specificity	0.33
DWT	3-median	Sigmoid MLP(64, 32, 8, 1).	Recall	0.48
DWT	Real-LC	Sigmoid MLP(1024).	Accuracy	0.9391
DWT	Real-LC	Sigmoid MLP(1024).	Precision	0.4999
DWT	Real-LC	Sigmoid MLP(1024).	Specificity	0.3855
DWT	Real-LC	Sigmoid MLP(1024).	Recall	0.5743
DWT	Real-LC	Sigmoid MLP(64, 32, 8, 1).	Accuracy	0.8794
DWT	Real-LC	Sigmoid MLP(64, 32, 8, 1).	Precision	0.4801
DWT	Real-LC	Sigmoid MLP(64, 32, 8, 1).	Specificity	0.6697
DWT	Real-LC	Sigmoid MLP(64, 32, 8, 1).	Recall	0.6696
DWT	Real-LC	Relu MLP(64, 32, 8, 1).	Accuracy	0.6357
DWT	Real-LC	Relu MLP(64, 32, 8, 1).	Specificity	0.8663
DWT	Real-LC	Relu MLP(64, 32, 8, 1).	Recall	0.4457
DWT	Real-LC	NB.	Recall	0.0975
EMD	3-median	KNN ($k = 9$).	Accuracy	0.78
EMD	Real-LC	RF.	Recall	0.27
EEMD	3-median	RF.	Recall	0.28
EEMD	Real-LC	KNN ($k = 5$).	Precision	0.39
EEMD	Real-LC	KNN ($k = 5$).	Specificity	0.36
EEMD	Real-LC	KNN ($k = 7$).	Specificity	0.7
EEMD	Real-LC	KNN ($k = 9$).	Precision	0.21
EEMD	Real-LC	KNN ($k = 9$).	Specificity	0.34

Note: When p-value > 0.05 there was no difference between preprocessing the LCs and not.

Chapter 4

Proposed WAvelet-Based Broad LEarning System (WABBLES)

The effectiveness of MRA for time series classification has been proven in the previous chapters. In this chapter, we propose an ML model based on the Broad Learning System (BLS) algorithm, which uses MRA as a critical component of the network. First, an introduction to BLS is presented. Next, the proposed model, termed WAvelet-Based Broad LEarning System (WABBLES), is defined. Later, the methodology followed to identify exoplanets using the proposed model is presented, along with the results obtained using simulated data and a comparison of its performance against the one attained by the other models presented in this work. Finally, the performance results obtained by the proposed model are also evaluated using a benchmark dataset of cancer detection.

4.1. Broad Learning System

Deep Learning (DL) architectures suffer from high resource consumption because of the large number of parameters required and the filter operations performed in their intermediate layers. The BLS algorithm offers an alternative to deep structures because it is based on the idea of expanding the nodes in wide instead of depending on hidden layers. BLS has the form of a flat network that uses two different sets of nodes called mapping and enhancement nodes. The original inputs are transformed into mapping features by the mapping nodes, which are then used to create the enhancement nodes.

The BLS model is shown in Fig. 4.1 [Chen and Liu, 2018], where it can be observed that BLS applies the mapping function to the kth sample vector from the input matrix $\boldsymbol{X}_N^M$, where M is the number of samples and N is the number of features. The matrix that contains the mapping nodes is denoted as $\boldsymbol{Z}^n = [\boldsymbol{Z_1}, ..., \boldsymbol{Z_n}]$ and it has all the groups of mapped features. The mapping function is described in Eq. (4.1) from [Chen and Liu, 2018].

$$\boldsymbol{Z_i} = \phi(\boldsymbol{W_{ei}}\boldsymbol{X_k} + \boldsymbol{\beta_{ei}}), \qquad i = 1, ..., n \tag{4.1}$$

where ϕ is the mapping function of the ith mapping node, $\boldsymbol{W_{ei}}$ and $\boldsymbol{\beta_{ei}}$ are the mapping weight and mapping bias of the ith mapping node, and both are randomly generated.

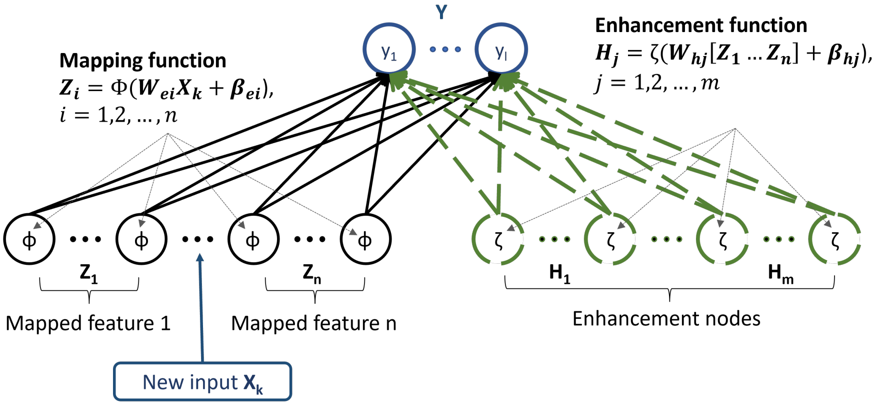


Figure 4.1: The Broad Learning System (BLS) architecture [Chen and Liu, 2018].

A layer of enhancement nodes is created by using the mapping nodes as inputs. The enhancement function is applied to the mapping nodes and the resulting matrix is denoted as $\boldsymbol{H}^m = [\boldsymbol{H_1}, ..., \boldsymbol{H_m}]$. This matrix contains the concatenation of all the enhancement nodes. The enhancement function is described in Eq. (4.2) from [Chen and Liu, 2018].

$$\boldsymbol{H_j} = \xi(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}), \qquad j = 1, ..., m \tag{4.2}$$

where ξ is the enhancement function of the jth enhancement node, $\boldsymbol{W_{hj}}$ and $\boldsymbol{\beta_{hj}}$ are the enhancement weight and enhancement bias of the jth enhancement node, and they are also generated randomly.

Finally, the original BLS architecture reduces the training time by using pseudo-inverse matrices to estimate the optimal parameter values of the network. This parameters are used to calculate the $\boldsymbol{Y}$ matrix that contains the responses of all the samples in $\boldsymbol{X}$ [Chen and Liu, 2018]. Nevertheless, as explained in [Gao et al., 2019], the pseudo-inverse matrices can be replaced by the gradient descent approach.

4.2. WAvelet-Based Broad LEarning System (WABBLES)

As it has been proven in Chapter 3, the use of MRA to preprocess time series data enhances the classification performance of the ML models. For this reason, we propose a new ML model based on the concepts of the BLS architecture and MRA. We termed such algorithm the WAvelet-Based Broad LEarning System (WABBLES) model ([Jara-Maldonado et al., 2022]) and it is presented in Figure 4.2.

The first step of the WABBLES model is to normalize the original inputs by using the *z-score* function. This function prevents having inputs with different normal distributions. The resulting matrix, denoted as $\boldsymbol{X}_N^M$ with M samples and N features, contains the normalized inputs with a mean 0 and a standard deviation 1. Then, an operator called Multidimensional Radial Wavelon (MRW) ([Zhang and Benveniste, 1992] and [Zhang, 1992]) is used to process the kth sample of $\boldsymbol{X}$. The MRW that we based our model on is presented in [Juárez-Guerra et al.,

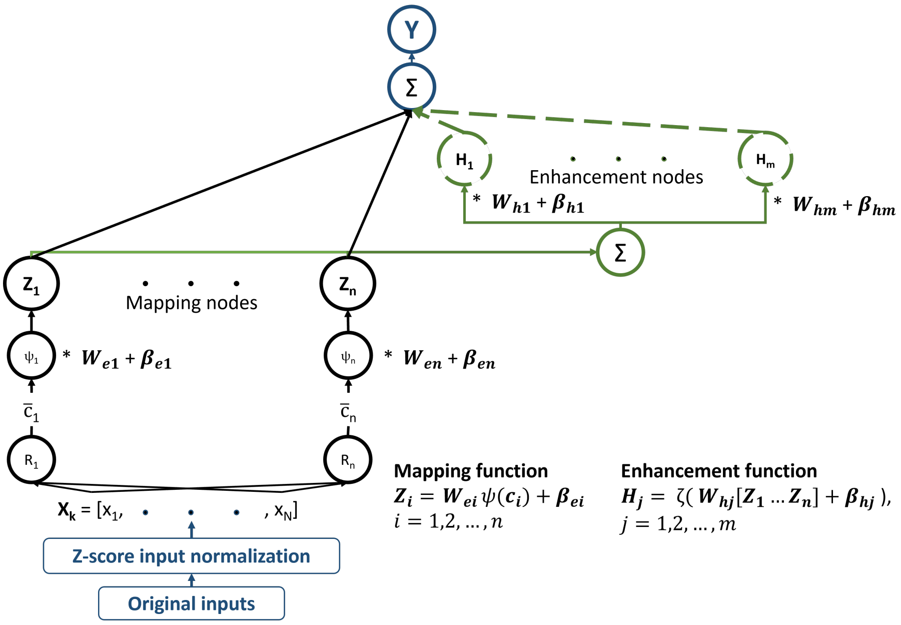


Figure 4.2: Proposed WAvelet-Based Broad LEarning System (WABBLES) architecture [Jara-Maldonado et al., 2022].

2020] and it is shown in Figure 4.3. The MRW transforms the inputs into a one dimensional value denoted by $\boldsymbol{c_i}, i = 1, ..., n$ (where n is the number of mapping nodes). This transformation is performed by the radial function R shown in Eq. (4.3). The value obtained by R is used by the one dimensional wavelet function ψ and the resulting output of the MRW is given by $\psi(\boldsymbol{c_i})$.

$$\boldsymbol{c_i} = ||\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})|| \tag{4.3}$$

where $\boldsymbol{c_i}$ is the output of the R function for the ith mapping node and $\boldsymbol{X_k}$ is the kth sample from the input matrix. Finally, $\boldsymbol{d_i}$ and $\boldsymbol{t}_i$ are the dilation and translation parameters of the wavelet used by the ith mapping node.

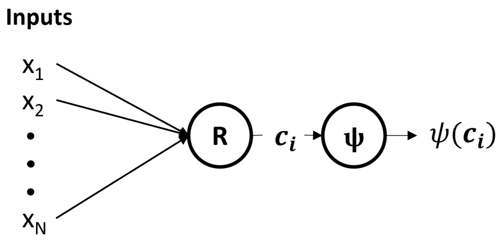


Figure 4.3: The Multidimensional Radial Wavelon (MRW) [Juárez-Guerra et al., 2020].

The output of $\psi(\boldsymbol{c_i})$ is used to calculate each mapping node and the network learns the values of the translation and dilation parameters of the wavelet function by itself. The proposed mapping function is shown in Eq. (4.4).

$$\boldsymbol{Z_i} = \boldsymbol{W_{ei}}\psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}, \qquad i = 1, ..., n \tag{4.4}$$

The enhancement nodes are calculated by obtaining the linear combination of all the mapping nodes processed by the enhancement function ξ. We have used the logistic sigmoid and the hyperbolic tangent functions to test the model with different enhancement functions, but any nonlinear function may be used as well. These functions are presented in Eqs. (4.5) and (4.6).

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{4.5}$$

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.6}$$

The enhancement function of the WABBLES model is presented in Eq. (4.7).

$$\boldsymbol{H_j} = \xi\big(\boldsymbol{W_{hj}}(\Sigma_{i=1}^{n}\boldsymbol{Z_i}) + \boldsymbol{\beta_{hj}}\big), \qquad j = 1, ..., m \tag{4.7}$$

The output of the network is denoted by $\boldsymbol{Y^M} = [y_1, ..., y_M]$ and it contains the expected labels. Finally, the response of the kth input is given by the linear combination of the mapping and enhancement nodes, as shown in Eq. (4.8). Additionally, it is possible to adjust the priority that the model gives to a certain class by adjusting the classification threshold value. Finally, the complete pseudocode of the WABBLES algorithm is presented in Algorithm 1 from

[Jara-Maldonado et al., 2022], which breaks down the process explained earlier.

$$y_k = \Sigma_{i=1}^{n} \boldsymbol{Z_i} + \Sigma_{j=1}^{m} \boldsymbol{H_j} \tag{4.8}$$

Algorithm 1 Wavelet-Based Broad LEarning System (WABBLES)

```
Input: Training samples $\boldsymbol{X}$, training true labels $y$, learning rate $\alpha$;
Output: $y_e$ response;
  $\boldsymbol{X} \leftarrow zscore(\boldsymbol{X})$
  for $i \leftarrow 0, n$ do
      Random $\boldsymbol{W_{ei}}$;
      $\boldsymbol{\beta_{ei}} \leftarrow 0$;
      Initialize the translation and dilation parameters ($\boldsymbol{t}$ and $\boldsymbol{d}$);
  end for
  for $j \leftarrow 0, m$ do
      Random $\boldsymbol{W_{hj}}$;
      $\boldsymbol{\beta_{hj}} \leftarrow 0$;
  end for
  while The error threshold is not satisfied within the limit of epochs do
      for each $\boldsymbol{x} \in \boldsymbol{X}$ do
          $sumZ, sumH \leftarrow 0$;
          for $i \leftarrow 0, n$ do
              $\boldsymbol{c_i} \leftarrow ||\boldsymbol{d_i}(\boldsymbol{x} - \boldsymbol{t_i})||$;
              Calculate $\psi(\boldsymbol{c_i})$;
              $\boldsymbol{Z_i} \leftarrow \boldsymbol{W_{ei}} * \psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}$;
              $sumZ \leftarrow sumZ + \boldsymbol{Z_i}$;
          end for
          for $j \leftarrow 0, m$ do
              Calculate $\boldsymbol{H_j} \leftarrow \xi(\boldsymbol{W_{hj}} * sumZ + \boldsymbol{\beta_{hj}})$;
              $sumH \leftarrow sumH + \boldsymbol{H_j}$;
          end for
          $y_e \leftarrow sumZ + sumH$;
          $error \leftarrow y_e - y$;
          Save parameters $(\boldsymbol{W_e}, \boldsymbol{\beta_e}, \boldsymbol{t}, \boldsymbol{d}, \boldsymbol{W_h}, \boldsymbol{\beta_h})$;
          Calculate the parameter derivatives ($\boldsymbol{W_e}'$, $\boldsymbol{\beta_e'}$, $\boldsymbol{t'}$, $\boldsymbol{d'}$, $\boldsymbol{W_h'}$, $\boldsymbol{\beta_h'}$);
          Update parameters ($\boldsymbol{W_e}$, $\boldsymbol{\beta_e}$, $\boldsymbol{t}$, $\boldsymbol{d}$, $\boldsymbol{W_h}$, $\boldsymbol{\beta_h}$);
      end for
  end while
```

4.2.1. Learning Algorithm of the WABBLES Model

The WABBLES model is trained using the backpropagation algorithm in contrast to the pseudo-inverse matrices used by the original BLS model. The backpropagation was required to train the mapping nodes presented in this work because it is necessary to train the dilation and translation parameters of the wavelet functions. The backpropagation algorithm uses the derivatives of the parameters to reduce the gradient of an error function $E(w)$ by adjusting the parameters. For this reason, the activation functions, $\psi()$ and $\xi()$, must be differentiable and nonlinear functions. We chose to use the quadratic cost function, shown in Eq. (4.9), to calculate the error of the network.

$$E(t) = \frac{1}{2}[y_e(t) - y(t)]^2 \tag{4.9}$$

where $y_e(t)$ is the estimated output, and $y(t)$ is the expected output at time t.

The cost function is used to update the parameters $W = [\boldsymbol{W_e}, \boldsymbol{\beta_e}, \boldsymbol{W_h}, \boldsymbol{\beta_h}, \boldsymbol{t}, \boldsymbol{d}]$ as it is done in [Juárez-Guerra et al., 2020]. Eq. (4.10) depicts the operations necessary to update the parameters.

$$W(t+1) = W(t) + \Delta W(t) = W(t) + \alpha\left(- \frac{\delta E}{\delta W(t)}\right) \tag{4.10}$$

where α is the learning rate.

The partial derivative of the cost function w.r.t the parameters is calculated by using Eq. (4.11).

$$\frac{\delta E}{\delta W(t)} = \Sigma_{k=0} e_k \frac{\delta y(t)}{\delta W(t)} \tag{4.11}$$

where k denotes the current epoch of the gradient descent algorithm, and e_k is the calculated error on that epoch.

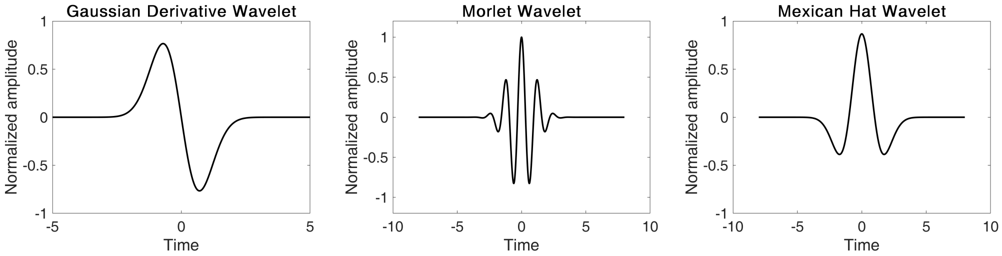
The value $\frac{\delta y(t)}{\delta W(t)}$ is obtained by calculating the partial derivatives of each parameter in the vector $W = [\boldsymbol{W_e}, \boldsymbol{\beta_e}, \boldsymbol{W_h}, \boldsymbol{\beta_h}, \boldsymbol{t}, \boldsymbol{d}]$. The full description of the calculation of the derivatives is presented in Appendix D, and the resulting partial derivatives are presented in Table 4.1.

Table 4.1: Partial Derivatives of the Parameters of the Proposed WABBLES Model.

Parameter	**Partial derivative of the output δy w.r.t. the parameter**
$\boldsymbol{\beta_{ei}}$	$\frac{\delta y}{\delta \boldsymbol{\beta_{ei}}} = 1 + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}}$
$\boldsymbol{\beta_{hj}}$	$\frac{\delta y}{\delta \boldsymbol{\beta_{hj}}} = \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)$
$\boldsymbol{W_{ei}}$	$\frac{\delta y}{\delta \boldsymbol{W_{ei}}} = \psi(\boldsymbol{c_i})\big(1 + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}}\big)$
$\boldsymbol{W_{hj}}$	$\frac{\delta y}{\delta \boldsymbol{W_{hj}}} = \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{Z}$
$\boldsymbol{t_i}$	$\frac{\delta y}{\delta \boldsymbol{t_i}} = \Big(\frac{\delta \psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}} \cdot \boldsymbol{W_{ei}}\Big) \cdot \Big(1 + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}}\Big)$
$\boldsymbol{d_i}$	$\frac{\delta y}{\delta \boldsymbol{d_i}} = \Big(\frac{\delta \psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} \cdot \boldsymbol{W_{ei}}\Big) \cdot \Big(1 + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}}\Big)$
ξ' option 1: Sigmoid function deriv.	$\xi' = \frac{e^{-x}}{(1+e^{-x})^2}$
ξ' option 2: Tanh function deriv.	$\xi' = 1 - tanh(x)^2$

4.2.2. Mapping Functions

The WABBLES model uses a wavelet function ψ to map the original inputs of the network to create the mapping nodes. To select the best wavelet function for ours tests, we experimented using three different mother wavelet functions. The wavelet chosen is the one that obtained the best results for the problems at hand, which are exoplanet identification and cancer detection. It is possible that each classification problem will obtain better results with different wavelets because the results will depend on the resemblance between the wavelet and the input signal. The functions that we tested are the Gaussian Derivative function defined in Eq. (4.12), the Morlet function defined in Eq. (4.13), and Mexican Hat function defined in Eq. (4.14). Other functions could be used as the mapping function with the only condition that they must be



(a) (b) (c)

Figure 4.4: Mother wavelet functions used in the present work. a) Gaussian Derivative wavelet, b) Morlet wavelet, and c) Mexican Hat wavelet.

continuous functions, to calculate their derivatives so they can be used by the training algorithm. This is why the wavelets used in this work are continuous functions and their graphs are shown in Fig. 4.4.

$$\text{Gaussian Derivative:} \quad \psi(\boldsymbol{c_i}) = -\boldsymbol{c_i} e^{-\frac{1}{2}\boldsymbol{c_i}^2} \tag{4.12}$$

$$\text{Morlet:} \quad \psi(\boldsymbol{c_i}) = e^{-\frac{1}{2}\boldsymbol{c_i}^2} cos(5\boldsymbol{c_i}) \tag{4.13}$$

$$\text{Mexican Hat:} \quad \psi(\boldsymbol{c_i}) = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}(1-\boldsymbol{c_i}^2)e^{-\frac{1}{2}\boldsymbol{c_i}^2} \tag{4.14}$$

where e is the exponential function, $\boldsymbol{c_i}$ is the output of the radial function R, and $\psi()$ is the wavelet function.

The initialization of the dilation and translation parameters of the wavelet function requires special attention (see [Abdala, 2008]), thus it is unadvised to use a random initialization strategy. Instead, [Zhang and Benveniste, 1992] propose an iterative procedure that is explained next and it is the one used for the initialization of the dilation and translation parameters of the proposed model. The translation ($\boldsymbol{t}$) and dilation ($\boldsymbol{d}$) parameters are initialized by selecting a p point between a and b, such that $a < p < b$. Then, p is calculated subsequently by dividing all the sub-intervals $[a, b]$ obtained using a recursive function. The next values of the parameters are set as $\boldsymbol{t_1} = p$, and $\boldsymbol{d_1} = c(b-a)$, where c is a constant equal to 0.5. The interval $[a, b]$ is recursively subdivided into two parts by using the new values of the point p, and the process is repeated until all the wavelet parameters have been initialized. There are some cases in which not all the wavelons were initialized by the recursive function. This occurs when the number of wavelons used is not a power of 2. For those cases, the parameters are initialized randomly. The aforementioned process is described in Algorithm 2 from [Jara-Maldonado et al., 2022].

Algorithm 2 Wavelet parameters initialization

```
Input: Number of mapping nodes $n$, number of features $N$;
Output: Wavelet translation and dilation parameter matrices in list representation $\boldsymbol{t}$, $\boldsymbol{d}$;
  $a \leftarrow 0$;
  $b \leftarrow 1$;
  $levels \leftarrow$floor(log($n$)/log(2));
  $\boldsymbol{t}, \boldsymbol{d} \leftarrow$InitializeTD($\boldsymbol{t}, \boldsymbol{d}, a, b, levels, N$);
  while The size of $\boldsymbol{t}$ is smaller than the number of mapping nodes $n$ do
      $\boldsymbol{t}$.add($a + (b-a) * random$);
      $\boldsymbol{d}$.add($0.5 + (b-a) * -levels^2$);
  end while
  procedure InitializeTD($\boldsymbol{t}, \boldsymbol{d}, a, b, levels, N$)
      $p = 0.5 * (a+b)$;
      for $j \leftarrow 0, N$ do
          $\boldsymbol{ti_j} = p$;
      end for
      $di = 0.5 * (b-a)$;
      $\boldsymbol{t}$.add($\boldsymbol{ti}$);
      $\boldsymbol{d}$.add($di$);
      if $level \leq 1$ then
          return $\boldsymbol{t}, \boldsymbol{d}$;
      else
          InitializeTD($\boldsymbol{t}, \boldsymbol{d}, a, p, levels-1, N$);
          InitializeTD($\boldsymbol{t}, \boldsymbol{d}, p, b, levels-1, N$);
      end if
  end procedure
```

Finally, the derivative of the $\psi(\boldsymbol{c_i})$ function w.r.t the $\boldsymbol{t_i}$ and $\boldsymbol{d_i}$ parameters is required for the training algorithm. We calculated the derivatives of the parameters of each wavelet function used in this work and the partial derivatives are presented in in Table 4.2. This table presents each of the three wavelet functions along with the derivatives used to update the translation and dilation parameters of the wavelet functions. Such derivatives are a complement for Table 4.1, where the derivatives of the other parameters were shown. The calculations of the dilation and translation derivatives are presented in Appendix E.

Table 4.2: Partial Derivatives of the Wavelet Parameters for the Proposed WABBLES Model.

Wavelet	**Param.**	**Partial derivatives of the output δy w.r.t. the wavelet parameters**
Gaussian Derivative	$\boldsymbol{t_i}$	$\frac{\delta\psi(\boldsymbol{c_i})}{d\boldsymbol{t_i}} = \boldsymbol{W_{ei}}\Big(e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot \boldsymbol{d_i}^2[\boldsymbol{x} - \boldsymbol{t_i}] \cdot (\frac{1}{\boldsymbol{c_i}} + \boldsymbol{c_i})\Big) \cdot \Big(1 + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}}\Big)$
	$\boldsymbol{d_i}$	$\frac{\delta\psi(\boldsymbol{c_i})}{d\boldsymbol{d_i}} = -\boldsymbol{W_{ei}}\Big((\boldsymbol{d_i}[\boldsymbol{x} - \boldsymbol{t_i}]^2) \cdot \big(\frac{1}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \psi(\boldsymbol{c_i})\big)\Big) \cdot \big(1 + \xi'(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}) \cdot \boldsymbol{W_{hj}}\big)$
Morlet	$\boldsymbol{t_i}$	$\frac{\delta\psi(\boldsymbol{c_i})}{d\boldsymbol{t_i}} = \boldsymbol{W_{ei}}\Big(\boldsymbol{d_i}^2[\boldsymbol{x} - \boldsymbol{t_i}] \cdot \big(\psi(\boldsymbol{c_i}) + \frac{5}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot sen(5\boldsymbol{c_i})\big)\Big) \cdot \big(1 + \xi'(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}) \cdot \boldsymbol{W_{hj}}\big)$
	$\boldsymbol{d_i}$	$\frac{\delta\psi(\boldsymbol{c_i})}{d\boldsymbol{d_i}} = -\boldsymbol{W_{ei}}\Big(\boldsymbol{d_i}(\boldsymbol{x} - \boldsymbol{t_i})^2 \cdot [\psi(\boldsymbol{c_i}) + \frac{5}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot sen(5\boldsymbol{c_i})]\Big) \cdot \big(1 + \xi'(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}) \cdot \boldsymbol{W_{hj}}\big)$
Mexican Hat	$\boldsymbol{t_i}$	$\frac{\delta\psi(\boldsymbol{c_i})}{d\boldsymbol{t_i}} = \boldsymbol{W_{ei}}\big(\boldsymbol{d_i}^2[\boldsymbol{x} - \boldsymbol{t_i}]\big)\big(\frac{4}{\pi^{\frac{1}{4}}\sqrt{3}} \cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \psi(\boldsymbol{c_i})\big) \cdot \big(1 + \xi'(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}) \cdot \boldsymbol{W_{hj}}\big)$
	$\boldsymbol{d_i}$	$\frac{\delta\psi(\boldsymbol{c_i})}{d\boldsymbol{d_i}} = -\boldsymbol{W_{ei}}\left(\boldsymbol{d_i}(\boldsymbol{x} - \boldsymbol{t_i})^2\Big(\frac{4}{\pi^{\frac{1}{4}}\sqrt{3}}(e^{-\frac{1}{2}\boldsymbol{c_i}^2}) + \psi(\boldsymbol{c_i})\Big)\right) \cdot \big(1 + \xi'(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}) \cdot \boldsymbol{W_{hj}}\big)$

4.2.3. Discussion

We have proposed a new ML model that is based on the BLS architecture. The fact that there are no hidden layers makes it possible to reduce the number of parameters to be trained, which results in less computational complexity. Moreover, the proposed model takes advantage of the mapping capacities of the wavelet functions to process the LC data. We have proven in Chapter 3 that MRA helps to enhance the transit signals, thus adding a wavelet function for the mapping nodes was expected to improve the performance of the proposed network. The main advantage of using wavelets is that they can adapt to the signal by carefully adjusting the dilation and translation parameters. Due to the self-learning nature of neural networks, the WABBLES model is capable of automatically adjusting the wavelet dilation and translation parameters, resulting on a perfect match between MRA and ML.

4.3. WABBLES Experiments

4.3.1. Experiment Settings for Exoplanet Identification

We have used the 3-median dataset reported in Chapter 3 to test the performance of the RF, CNN, NB, SVM, MLP (specifically the Relu MLP (1024)), Ridge classifier, KNN, and LS models described in Chapter 3. We have also added the Wavelet MLP, BLS and WABBLES models to increase the reach of the experiments. We based on the Wavelet MLP presented in [Pearson et al., 2017], which is an MLP that uses the DWT to preprocess the inputs. As mentioned before, the Wavelet MLP model uses a concatenation of the cAs and cDs obtained from the first level of decomposition of the second order Daubechies wavelet as inputs for the network. The configuration of the MLP network used by the Wavelet MLP is the same as the Sigmoid and Relu MLP(64, 32, 8, 1) presented in Chapter 3 but it uses the preprocessed inputs.

The setting of the WABBLES model was defined by testing and comparing the results obtained by the model with the different options of hyperparameters, which are presented in Table 4.3. This table shows the different numbers of mapping and enhancement nodes, wavelets and activation functions tested. Additionally, we tested using amplitude inversion, where the LCs were multiplied by -1 to give them a similar geometry to some of the wavelets used. The learning rate of the network was set to 0.001 after testing with the values 0.0005, 0.001, and 0.005. We chose these values because we noticed that the performance of the model dramatically dropped out of this range. Each test was executed 10 times to obtain a statistical validation of the results. In total, we tested 218 settings of the WABBLES. The best setting obtained from these tests was trained with a limit of 500 training epochs, from which it required 473 epochs to reach its optimal performance. Such setting consists of the Mexican Hat wavelet, 20 mapping and enhancement nodes, the tanh activation function and amplitude inverted LCs. Finally, the WABBLES model was implemented in Python[1].

Similar to the WABBLES model, the configuration of the BLS model was determined using different settings from the hyperparameter grid presented in Table 4.4. The parameters that

[1] WABBLES model code: `https://github.com/MiguelJM/WABBLES.git`

Table 4.3: Hyperparameter grid for the selection of the WABBLES configuration [Jara-Maldonado et al., 2022].

Wavelet	Gaussian Deriv.	Morlet	**Mexican Hat**
Mapping nodes (n)	10	**20**	50
Enhancement nodes (m)	10	**20**	50
Activation function (ϕ)	**Tanh**	Sigmoid	
LC display	**Inverted**	Normal	

Note: The best combination of parameters for exoplanets is highlighted with bold characters.

were varied are the shrinkage parameter, the number of mapping nodes, mapping groups, and enhancement nodes. The sparse regularization parameter (C) was set to $2.00E^{-25}$ in all cases because we found that there is not much difference when setting a value in the range between $2.00E^{-10}$ and $2.00E^{-100}$, so we used the default value. Each instance of the model was tested 10 times to statistically validate the results. In total, we tested 250 different settings, where the best performing setting consisted of the shrinkage parameter, s set to 0.8, the number of mapping nodes N_1 to 10, and the number of groups of mapping nodes N_2, as well as the number of enhancement nodes N_3 were set to 100. We used the Python implementation of the BLS from [Li et al., 2019][2].

Table 4.4: Hyperparameter grid for the selection of the BLS configuration [Jara-Maldonado et al., 2022].

Shrinkage (s)	0.1	**0.8**			
Mapping nodes (N_1)	**10**	20	30	70	100
Groups of mapping nodes (N_2)	10	20	30	70	**100**
Enhancement nodes (N_3)	10	20	30	70	**100**

Note: The best combination of parameters for exoplanets is highlighted with bold characters.

4.3.2. Experiment Settings for Breast Cancer Detection

We performed tests with a benchmark dataset called Wisconsin Breast Cancer Dataset (Original) [Wolberg and Mangasarian, 1990]. This was done to test the performance of the proposed model with a real world application. This dataset has been retrieved from the UCI

[2] BLS Python code: `http://www.sfu.ca/~ljilja/cnl/projects/BLS_intrusion_detection/index.html`

machine learning repository[3]. It contains breast cancer clinical cases that have been labeled as malignant or benign. There is a total of 699 samples and each contain 10 attributes. The attributes are sample code number, clump thickness, uniformity of cell size and shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. Finally, the dataset is unbalanced, meaning that there are more samples corresponding to one class than the other. In this case the 65.5% of the samples belong to the benign class and the other 34.5% cases belong to the malignant class. Some of the samples have missing values. In those cases we replaced the values with 0's. Also, we changed the classification labels to 0 for malignant (which appear as 4's in the original dataset), and 1 for the benign cases (originally 2's). Finally, we split the dataset into 60% to be used as training set, and the resting 40% as testing set.

The ML classifiers tested with this dataset are the proposed WABBLES model, BLS, RF, and two MLP (both with one hidden layer and 5 neural units in accordance to what is proposed in [Alarcon-Aquino and Barria, 2006]). The Sidmoid MLP(5) uses the sigmoid activation function and the Relu MLP(5) uses the Relu activation function. In the case of the WABBLES model, we used the same hyperparameter grid shown in Table 4.3 with the exception that we did not test amplitude inversion. This resulted in 54 different tests with 5 training epochs. The best configuration found is composed by the Gaussian derivative wavelet, 20 mapping and enhancement nodes, and the sigmoid enhancement function. This model was trained with a maximum of 500 training epochs, from which it only used 332 to attain its best value. Then, we varied the threshold value from 0 to 1 with advancements of 0.1. These other 10 tests allowed us to configure the best threshold value, which was found to be 0.5. We performed the same threshold tests with the rest of the models and determined that the best values were obtained with a threshold value of 0.6 for the RF, 0.9 with the two MLPs (Relu MLP(5) and Sigmoid MLP(5)), and 0.5 for the BLS model. The threshold tests were performed 100 times for statistical validation. The configuration of the BLS model was defined by using the hyperparameter grid presented in Table 4.4. From the 250 BLS configurations tested, we found that the best performing one was composed of a shrinkage parameter set to 0.8, 100 mapping

[3] Breast Cancer Wisconsin Dataset (Original): `https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28original%29`

nodes and groups of mapping nodes (N_1 and N_2), and 20 enhancement nodes (N_3).

4.3.3. Exoplanet Identification Experimental Results

First, the ML models were implemented and tested using the 3-median dataset presented in Chapter 3. The metrics used to assess the performance of the models compared are accuracy, precision, specificity and recall, which are described in Chapter 3. In addition, we also included the F-score metric, which is calculated as $(2 * (\text{precision} * \text{recall})/(\text{precision} + \text{recall}))$ and is useful for unbalanced datasets, such as the breast cancer dataset that was also used. All the experiments were performed using a computer with an Intel Core i7-7700 HQ CPU, 16.0 GB of RAM, Windows 10 operative system of 64 bits, and a NVIDIA GeForce GTX 1060 graphics card. The results of the exoplanet experiments are presented in Table 4.5. The only two models that were trained once were the CNN and WABBLES models. This was done in this way because of the long training times of both models. In contrast, the rest of the models were trained 100 times and then tested after each train. We present the average of all the training and testing epochs on Table 4.5. As it can be observed in this table, the model that obtained the best accuracy and specificity percentages is the proposed WABBLES model, while it obtained descent values in the other metrics, although lower than the other models presented. In the case of the RF and CNN models, which obtained the best results in Chapter 3, the proposed model has a better accuracy and F-score by more than one percent, which is a descent result for a new ML model. Also, Table 4.6 shows the testing and training times taken by the best performing models from Table 4.5. As it can be seen, the WABBLES model requires more training time than the rest of the models, but once it has been trained it classifies almost three times faster than the BLS model.

We also conducted an experiment to evaluate how the number of training epochs affects the performance of the WABBLES model. The best results were obtained using a total of 473 epochs from a limit of 500. The other training epochs tested were 2, 5, and 30 epochs. In that order, the accuracy values obtained were 93.36%, 95.51%, and 95.91%; while the execution times were 324.29, 604.88, and $1,905.68$ seconds. Thus, we found that using more training

Table 4.5: Experimental results with the 3-median exoplanet dataset. Each value corresponds to the average of the 100 executions of each model with the testing set [Jara-Maldonado et al., 2022].

Model	**Accuracy (%)**	**F-Score (%)**	**Precision (%)**	**Specificity (%)**	**Recall (%)**
WABBLES.	**99.01**	**99.01**	99.24	**99.23**	98.79
BLS.	98.57	98.56	97.28	97.25	99.89
RF.	97.82	97.84	97.25	97.17	98.45
CNN.	97.68	97.65	**99.94**	97.68	95.48
NB.	94.75	95.03	90.81	92.42	99.67
SVM.	93.76	94.17	88.99	87.41	**100**
Relu MLP (1024).	93.31	93.77	88.95	87.39	99.14
Ridge Classifier.	76.02	80.76	67.79	51.81	99.87
KNN.	64.91	74.20	58.99	29.15	**100**
LS.	37.99	6.58	13.65	72.22	4.34

Note: The best values are highlighted with bold characters.

Table 4.6: Execution time against training time of the best performing models with the 3-median exoplanet dataset [Jara-Maldonado et al., 2022].

Model	**Testing time (secs.)**	**Training time (secs.)**
RF.	**0.04**	8.64
CNN.	16.34	3,727.62
WABBLES.	47.67	73,210.30
BLS.	136.78	**1.28**

Note: The best values are highlighted with bold characters.

epochs results in a better accuracy with the disadvantage of consuming more time to train the model. Furthermore, a comparison of the performance of the WABBLES model with the wavelet functions used is shown in Table 4.7. The best results were obtained using the Mexican Hat wavelet transform. We attribute this to the similarity between the exoplanet transit and the inverted Mexican Hat wavelet.

Furthermore, we performed tests to evaluate the variance of the results using different numbers of runs. As it is mentioned before, we performed 100 runs, from which we extracted the mean values. Those are the results reported in this section. Nevertheless, Figures 4.5 and 4.6 show the mean and standard deviation of the accuracy and F-score results obtained by the four

Table 4.7: WABBLES results using different wavelets as mapping function with the 3-median exoplanet dataset [Jara-Maldonado et al., 2022].

Wavelet	Test Accuracy (%)	Train Accuracy (%)	No. of training epochs
Mexican Hat.	**99.01**	**98.8**	473
Morlet.	97.95	96.88	460
Gaussian Deriv.	97.52	97.24	**40**

Note: The best values are highlighted with bold characters.

best performing models with 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 runs. In this figures, the lines represent the mean of the results in the corresponding number of runs, while the shaded areas show the standard deviation of the results. As it can be seen, the model that obtains broader ranges of result values is the BLS model, which probably means that it is not being optimized, but rather it is stuck in a local minimum.

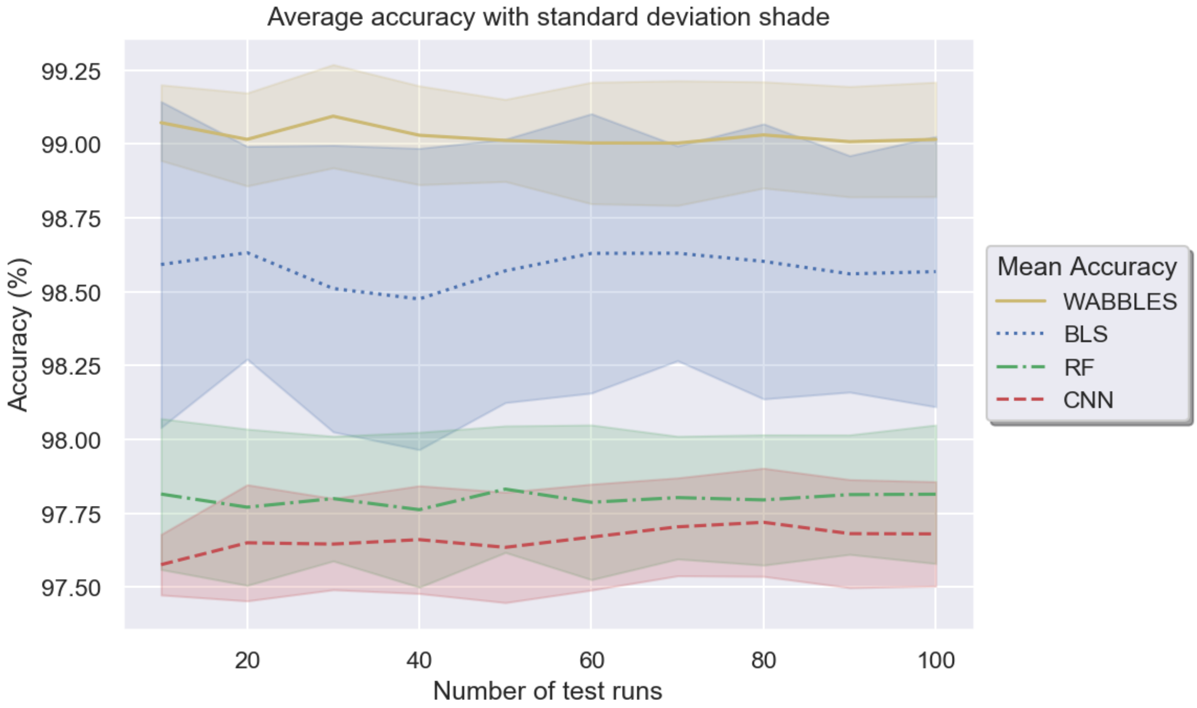


Figure 4.5: Mean and standard deviation of the accuracy from the best four models used with the 3-median dataset using different numbers of test runs.

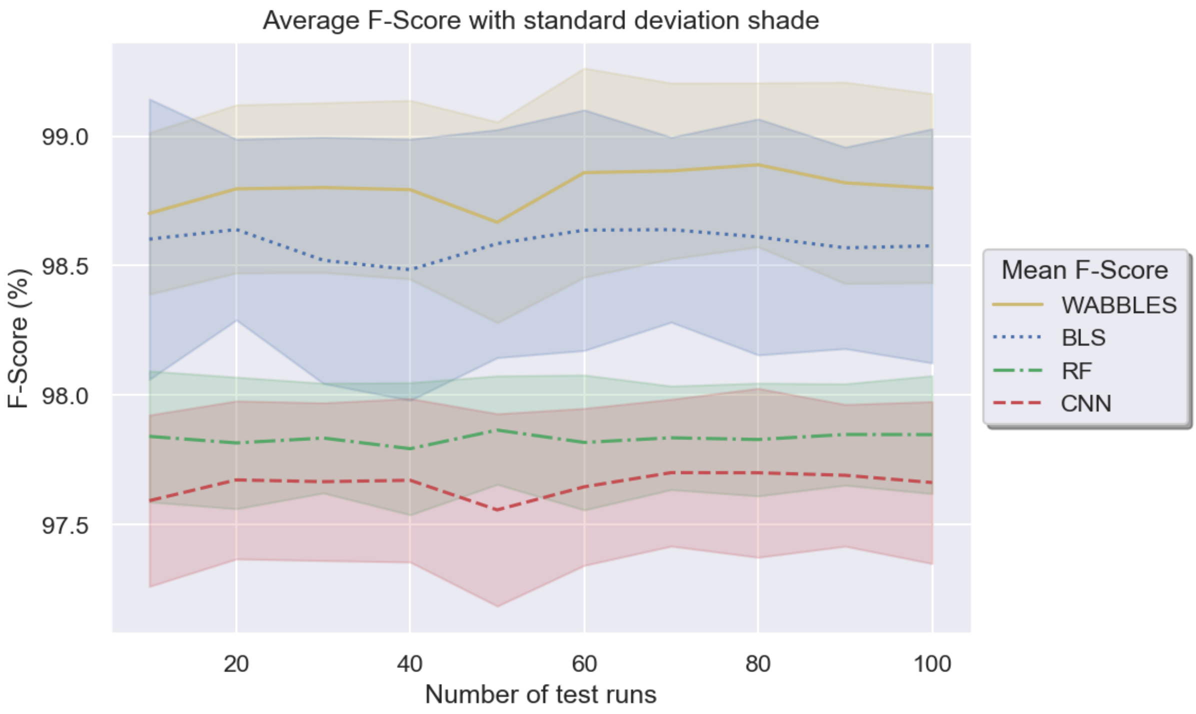


Figure 4.6: Mean and standard deviation of the F-score from the best four models used with the 3-median dataset using different numbers of test runs.

4.3.4. Cancer Detection Experimental Results

The results of the experiments performed with the cancer detection dataset are presented in Tables 4.8 and 4.9. The former table presents the performance metrics. As it can be seen, the WABBLES model is almost more than 1% better than any of the other models presented in every metric. Furthermore, the difference in F-score between the proposed model and the former BLS model is greater than 6%. Recall that the F-score is one of the most useful metrics for unbalanced datasets because it establishes a relation between the precision and recalls, which means that both classes are taken into account. In the case of time, it is noticeable from Table 4.9 that WABBLES is still one of the most time consuming models, although it needed less testing time than BLS. Nevertheless, the training time of WABBLES is the main caveat of this model. For this reason, we will perform experiments with the aim of parallelizing the model in the future.

More over, as in the case of the 3-median dataset tests, we evaluated the variance of the results using different numbers of runs. Figures 4.7 and 4.8 show the mean and standard

Table 4.8: Experimental results with the Wisconsin Breast Cancer Dataset (Original). Each value corresponds to the average of the 100 executions of each model with the testing set [Jara-Maldonado et al., 2022].

Model	Accuracy (%)	F-Score (%)	Precision (%)	Specificity (%)	Recall (%)
WABBLES.	**97.36**	**96.23**	**95.17**	**97.41**	**97.4**
RF.	96.37	94.76	93.17	96.35	96.42
Sigmoid MLP(5).	95.87	94.09	92.46	95.94	95.77
Relu MLP(5).	95.43	93.38	92.41	95.99	94.37
BLS.	87.83	90.41	92.84	88.35	87.02

Note: The best values are highlighted with bold characters.

Table 4.9: Testing time against training time of the models with the Wisconsin Breast Cancer Dataset (Original) dataset [Jara-Maldonado et al., 2022].

Model	Testing time (secs.)	Training time (secs.)
Sigmoid MLP(5).	**Close to 0.**	**0.0006**
Relu MLP(5).	**Close to 0.**	0.001
RF.	0.0002	0.0011
WABBLES.	0.83	2,509.59
BLS.	1.2	679.48

Note: The best values are highlighted with bold characters.

deviation of the accuracy and F-score results obtained by the three best performing models and the BLS model with 10 to 100 runs. As before, the lines represent the mean of the results in the corresponding number of runs, while the shaded areas show the standard deviation of the results. Once again, the model with the greater standard deviation is the BLS, while WABBLES still obtains the best results of all, even considering the standard deviations.

4.3.5. Discussion

We proposed a new ML model, which we called WABBLES. Then, we compared its performance with the results obtained by other exoplanet identification models that were tested in this work. Such tests were performed using the 3-median dataset, which consists of simulated LCs that were injected with simulated exoplanet transits and the breast cancer detection dataset. In the case of the exoplanet experiments presented in this chapter, we compared the

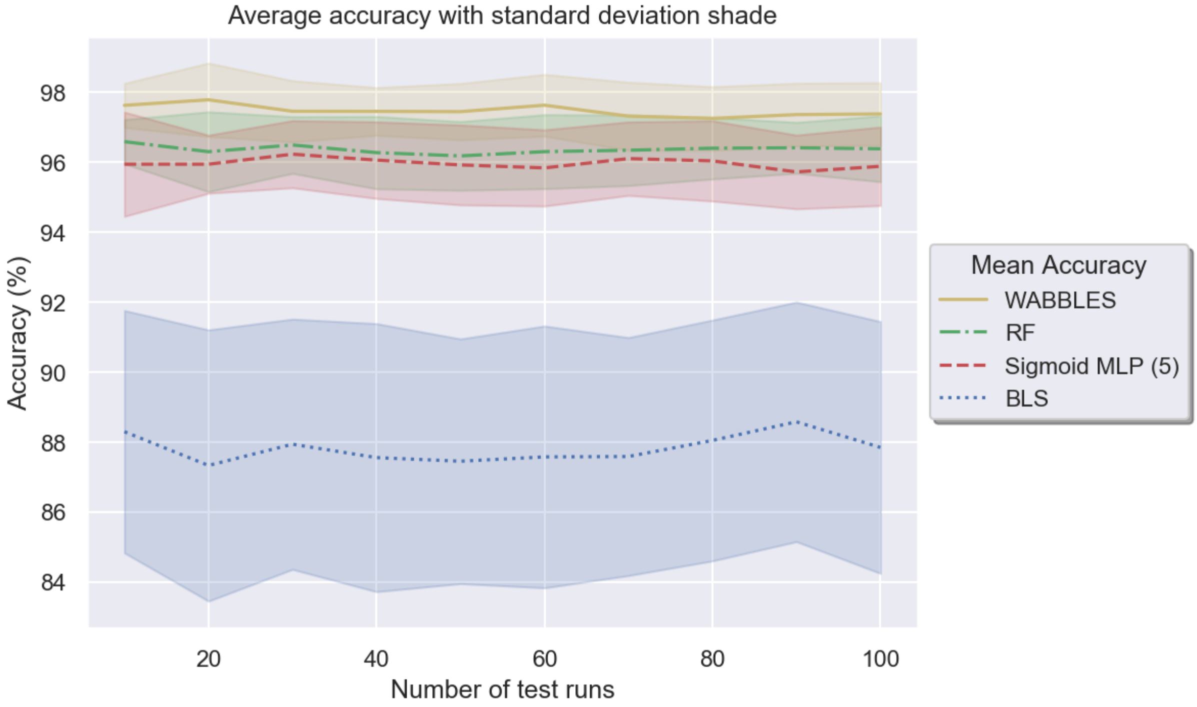


Figure 4.7: Mean and standard deviation of the accuracy from the best three models and the BLS model used with the breast cancer dataset; using different numbers of test runs.

results of the models with inputs that were not preprocessed using MRA techniques (contrarily to what was done in Chapter 3). In this regard, the best performing models are the proposed WABBLES model and the BLS model that it is based on. Nevertheless, for comparison purposes Table 4.10 shows the top five results obtained with the 3-median dataset. Those results include every experiment performed in this work. It can be observed that the WABBLES model obtained very similar results to those attained by the best settings of the CNN.

Table 4.10: Top 5 results from all the models tested with the 3-median exoplanet dataset. Each value corresponds to the average of the 100 executions of each model.

Model	Accuracy (%)	Precision (%)	Specificity (%)	Recall (%)
CNN+EMD (1 IMF).	**99.32**	**99.67**	**99.32**	98.99
CNN+DWT (sym5, cDs, $j = 1$).	99.13	99.16	99.13	99.09
Proposed WABBLES.	99.01	99.24	99.23	98.79
BLS.	98.57	97.28	97.25	**99.89**
RF+DWT (db1, cAs, $J = 4$).	98.08	97.49	97.41	98.73

Note: The best values are highlighted with bold characters.

It is worth highlighting that the accuracy percentage of the WABBLES model is 99.01%

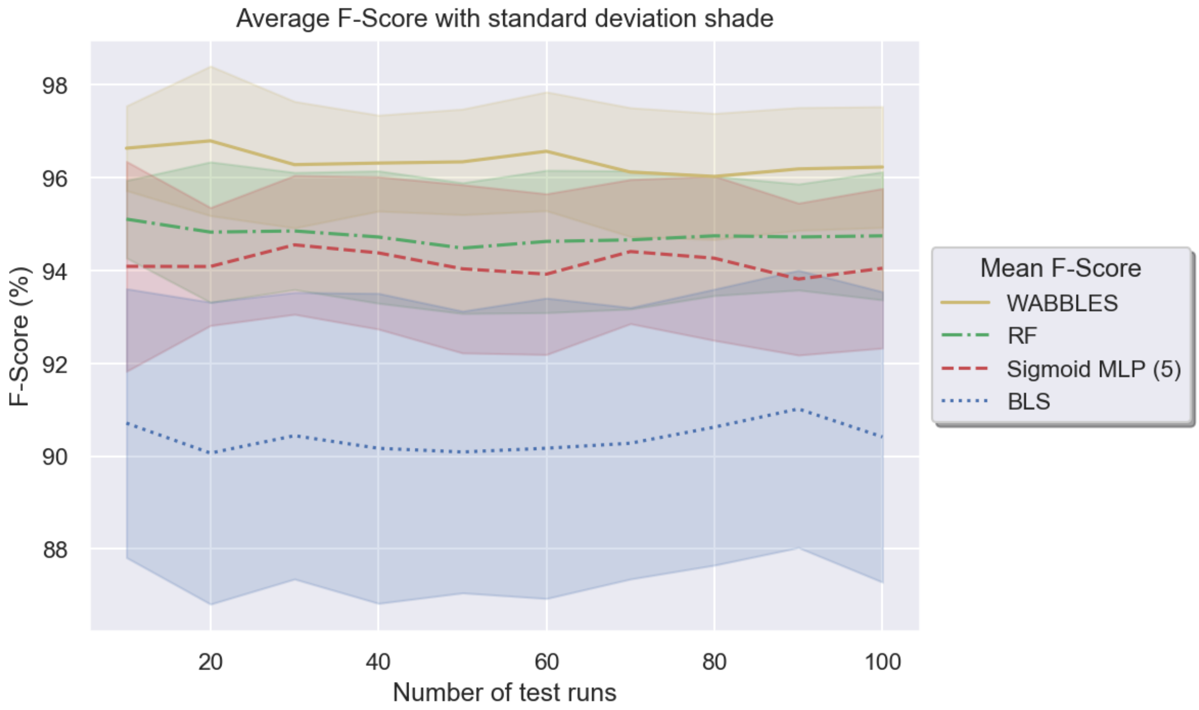


Figure 4.8: Mean and standard deviation of the F-score from the best three models and the BLS model used with the breast cancer dataset; using different numbers of test runs.

without the need of applying any MRA technique to preprocess the LCs before using them as inputs. Under these conditions, the CNN model obtained an accuracy of 97.68%, while the RF classifier obtained 97.82%, as it is shown in Table 4.5. These two values are considerably lower than that which was attained by the proposed model. We attribute the positive results of the proposed model to the fact that MRA is part of its core design, and it is used to map the inputs, thus highlighting subtle yet important features of the transit signal within the LCs. The fact that WABBLES is capable of autonomously training the wavelet dilation and translation parameters using the advantages of the neural network architecture and the backpropagation algorithm guarantees a better configuration of the wavelet transform. WABBLES also obtains the best results of the models tested with the Wisconsin Breast Cancer Dataset (Original) dataset.

Finally, the main limitation of the proposed model is its training time. It requires many training epochs to adjust its parameters because of the backpropagation algorithm. We have experimentally demonstrated in this chapter that the more training epochs the better accuracy, but also the more execution time consumed.

Chapter 5

Conclusions and Future Work

5.1. Conclusions

In this thesis, we have investigated how MRA can help to improve the identification accuracy of ML models. The hypothesis plotted is that *A machine learning model based on multiresolution analysis is capable of overmatching the transiting exoplanet identification accuracy performance of the state-of-the-art models for noisy light curves classification.* For this purpose, we have proposed a novel ML algorithm that can be adapted to any identification problem, such as cancer detection, which is another application of the model tested in this work. Since exoplanet research requires the analysis of big amounts of noisy data, the experiments performed with simulated LCs provide a proof of concept of the reach of combining MRA and ML for the identification of non-linear signals.

In the particular case of exoplanet discovery, we have studied how ML can be used for speeding up and reducing the effort needed to preprocess LCs, detect possible exoplanetary signals, and finally analyze the detected signals to evaluate if they actually correspond to an exoplanet transit (see [Jara-Maldonado et al., 2020b]). For the last step, the identification process, we have gone beyond by proposing different automation solutions. First, in Chapter 3, we designed a system that allowed us to create simulated LCs based on real exoplanet and LCs data. Using these time series we performed different experiments to improve the identification performance of the state-of-the-art ML models such as CNNs and RFs. We compared the performance of the ML models tested using the LCs without any preprocessing as inputs against those performances obtained with LCs preprocessed with different MRA techniques; namely the DWT, EMD, and EEMD techniques. From these sets of experiments we concluded that MRA does improve the performance of the models. For example, the use of the DWT reduces the execution time of the models because the signal is down-sampled. The DWT also increases the accuracy performance of the models because, even though some information is lost during the down-sampling process, it recovers the most significant features from the signal. The EMD and EEMD techniques do not down-sample the signal. Even though this increases the execution time of the models, it makes them robust against different noise sources. The main limitation found with the EMD and EEMD models is that they lack theoretical properties, which is not the case of the DWT. This is important because one of the most problematic issues in exoplanet discovery is the noise present in the LCs. An example of this is that looking for weak transit signals is important to find Earth-like exoplanets (see for example [Petigura et al., 2013]). In this sense, a more refined identification has to be performed; which is another reason for choosing MRA as an useful tool for exoplanetary science.

After proving the effectiveness of MRA for exoplanet identification, and basing on the idea that to automate the identification process it is necessary to overcome the noise issues, we proposed and implemented a new ML model called WABBLES in Chapter 4. The WABBLES model is based on the BLS architecture, which is a flattened network that maps the inputs and then enhances the mapped features to obtain the outputs. WABBLES is capable of extracting the most representative features from the signal of interest (in this case we looked for exoplanet transits and breast cancer) by analyzing the input signal at different resolution levels. This is done by using wavelets (i.e. mapping the inputs with wavelets). Furthermore, since neural networks are capable of automatically optimizing their parameters, the wavelet parameters are trained for a better identification performance. The proposed model can be thus adapted to any non-linear signal (not only LCs as it was proven in Chapter 4), with the only requirement that the most suitable wavelet and enhancement function must be used.

Finally, we tested the performance of the WABBLES model and then compared it against other ML models using one of the synthetic datasets that we created and the Wisconsin Breast Cancer Detection Dataset (Original). From these tests we concluded that the identification performance of the WABBLES model is superior in terms of accuracy and F-score percentage to that of the compared models, including the original BLS model.

5.2. Future Work

Even though the WABBLES model attained the best accuracy and F-score metrics, it still suffers from long training times, which could be mitigated by developing an adaptive learning rate strategy. The current model uses a fixed learning rate. Adapting it during the training process would accelerate its training times because the model could adjust the parameters with more freedom at the first training epochs and then, as the gradient approaches to the optimal value, the updating values could be smaller. Another improvement would be parallelizing the model. This would be possible because the model is trained by using the gradient descent algorithm, and because the mapping and enhancement nodes work individually in their respective layers. By parallelizing the model, it would have a significant acceleration of the training and testing processes. Furthermore, we only tested the proposed model with simulated data, for which we will also test the performance of the WABBLES model with real data.

Moreover, it is possible that the proposed model could be adapted to perform incremental learning, which means adding new nodes to an already trained network based on new knowledge. In the case of astronomic data, this can be useful to keep the pace of new space missions that continue to acquire LCs. Even more, we will study the possibility of the model to perform transfer learning, which would allow the user to use a previously trained architecture as starting point to train the model for a different problem. In this way, we expect that it will be possible to transfer the knowledge obtained by WABBLES from one telescope dataset to another.

In addition, in this work we have investigated the benefits of different MRA techniques, but we have not tested them all. One technique that could obtain good results is the SWT. This technique works similarly to the DWT with the difference that it does not decimate the information. For this reason this technique does not suffer from loss of information and it could be used to improve the performance of certain models such as the CNN which was constrained by the length of the input feature vector. Even more, we have not tested using the DWT to preprocess the inputs of the proposed model. This technique could reduce the execution time of the proposed model because the downsampling process limits the length of the inputs.

Furthermore, the proposed WABBLES model could also be used for exoplanet detection (i.e. spotting signals that possibly belong to exoplanet transits in the LCs) after applying the correct modifications. This would require adapting the model to skim over the LCs to detect periodic signals that could belong to exoplanet transits. By doing so, the model could be used for a completely automated pipeline, capable of extracting exoplanet candidates from the LCs for their future study by astronomers.

Finally, we have found that there is probably a relation between the LCs that contain exoplanet transits and the maximum number of IMFs that can be obtained by applying the EMD technique to the LCs. It would be of great interest to delve into this finding because a new exoplanet identification model could be proposed if there is actually a relation between the exoplanets and the IMFs obtained.

Bibliography

[Abdala, 2008] Abdala, M. (2008). Genetically based wavenets for system modeling. *i-manager's Journal for software engineering*, 3:66–72.

[Aigrain and Favata, 2002] Aigrain, S. and Favata, F. (2002). Bayesian detection of planetary transits. A modified version of the Gregory-Loredo method for Bayesian periodic signal detection. Astronomy & Astrophysics, 395:625–636.

[Alarcon-Aquino and Barria, 2009] Alarcon-Aquino, V. and Barria, J. (2009). Change detection in time series using the maximal overlap discrete wavelet transform. *Latin American Applied Research*, 39(2).

[Alarcon-Aquino and Barria, 2006] Alarcon-Aquino, V. and Barria, J. A. (2006). Multiresolution fir neural-network-based learning algorithm applied to network traffic prediction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(2):208–220.

[Ansdell et al., 2018] Ansdell, M. et al. (2018). Scientific domain knowledge improves exoplanet transit classification with deep learning. *The Astrophysical Journal*, 869(1):L7.

[Armstrong et al., 2018] Armstrong, D. J. et al. (2018). Automatic vetting of planet candidates from ground-based surveys: machine learning with NGTS. *Monthly Notices of the Royal Astronomical Society*, 478(3):4225–4237.

[Beck et al., 2017] Beck, T. et al. (2017). The CHEOPS (characterising exoplanet satellite) mission: telescope optical design, development status and main technical and programmatic challenges. volume 10562.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[Bravo et al., 2014] Bravo, J. P. et al. (2014). Wavelets: a powerful tool for studying rotation, activity, and pulsation in Kepler and CoRoT stellar light curves. Astronomy & Astrophysics, 568:A34.

[Burke et al., 2014] Burke, C. J. et al. (2014). Planetary candidates observed by Kepler IV: Planet sample from Q1-Q8 (22 months). *The Astrophysical Journal Supplement Series*, 210(2):19.

[Carpano, S. et al., 2003] Carpano, S., Aigrain, S., and Favata, F. (2003). Detecting planetary transits in the presence of stellar variability - optimal filtering and the use of colour information. *A&A*, 401(2):743–753.

[Carter and Nathan Winn, 2009] Carter, J. and Nathan Winn, J. (2009). Parameter estimation from time-series data with correlated errors: A wavelet-based method and its application to transit light curves. *The Astrophysical Journal*, 704.

[Catanzarite, 2015] Catanzarite, J. H. (2015). Autovetter planet candidate catalog for Q1-Q17 data release 24. *Astronomy & Astrophysics*.

[Chaushev et al., 2019] Chaushev, A., Raynard, L., Goad, M. R., Eigmller, P., Armstrong, D. J., Briegal, J. T., Burleigh, M. R., Casewell, S. L., Gill, S., Jenkins, J. S., Nielsen, L. D., Watson, C. A., West, R. G., Wheatley, P. J., Udry, S., and Vines, J. I. (2019). Classifying exoplanet candidates with convolutional neural networks: application to the Next Generation Transit Survey. *Monthly Notices of the Royal Astronomical Society*, 488(4):5232–5250.

[Chen and Liu, 2018] Chen, C. L. P. and Liu, Z. (2018). Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):10–24.

[Chui, 1997] Chui, C. (1997). *Wavelets: A Mathematical Tool for Signal Analysis*. Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics.

[Committee, 2011] Committee, E. S. (2011). Statistical significance and biological relevance. *EFSA Journal*, 9(9):2372.

[Coughlin et al., 2016] Coughlin et al. (2016). Planetary candidates observed by kepler. VII. the first fully uniform catalog based on the entire 48-month data set (Q1–Q17 DR24). *Astrophysical Journal Supplement Series*, 224(1):12.

[Dattilo et al., 2019] Dattilo, A. et al. (2019). Identifying exoplanets with deep learning. II. two new super-earths uncovered by a neural network in k2 data. *The Astronomical Journal*, 157(5):169.

[Daubechies, 1992] Daubechies, I. (1992). *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics.

[de Mayo, 2015] de Mayo, B. (2015). *The Everyday Physics of Hearing and Vision.* 2053-2571. Morgan & Claypool Publishers.

[Fontugne et al., 2017] Fontugne, R., Borgnat, P., and Flandrin, P. (2017). Online Empirical Mode Decomposition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4306–4310.

[Foster, 1996] Foster, G. (1996). Wavelets for period analysis of unevenly sampled time series. AJ, 112:1709.

[Gabor, 1947] Gabor, D. (1947). Theory of communication. *Journal of the Institution of Electrical Engineers - Part I: General*, 94(73):58–.

[Gao et al., 2019] Gao, S., Guo, G., and Chen, C. L. P. (2019). Event-based incremental broad learning system for object classification. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2989–2998.

[Graps, 1995] Graps, A. (1995). An introduction to wavelets. *IEEE Comput. Sci. Eng.*, 2(2):50–61.

[Grziwa and Pätzold, 2016] Grziwa, S. and Pätzold, M. (2016). Wavelet-based filter methods to detect small transiting planets in stellar light curves. *arXiv e-prints*, page arXiv:1607.08417.

[Haykin, 2009] Haykin, S. S. (2009). *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition.

[Hernndez and Meneses, 2022] Hernndez, L. R. and Meneses, M. A. S. (2022). Technology, science and culture: A global vision, volume iii. In Hernndez, L. R. and Meneses, M. A. S., editors, *Technology, Science and Culture*, chapter 1. IntechOpen, Rijeka.

[Howell et al., 2014] Howell, S. B. et al. (2014). The K2 Mission: Characterization and early results. *Publications of the Astronomical Society of the Pacific*, 126(938):398–408.

[Huang and Wu, 2008] Huang, N. E. and Wu, Z. (2008). A review on Hilbert-Huang transform: Method and its applications to geophysical studies. *Reviews of Geophysics*, 46(2).

[J. Lissauer, 2000] J. Lissauer, J. (2000). How common are habitable planets? *Nature*, 402:C11–4.

[Japkowicz and Shah, 2011] Japkowicz, N. and Shah, M. (2011). Evaluating learning algorithms: A classification perspective.

[Jara-Maldonado et al., 2022] Jara-Maldonado, M., Alarcon-Aquino, V., and Rosas-Romero, R. (2022). A new machine learning model based on the broad learning system and wavelets. *Engineering Applications of Artificial Intelligence*, 112:104886.

[Jara-Maldonado et al., 2020a] Jara-Maldonado, M., Aquino, V. A., and Rosas-Romero, R. (2020a). A multiresolution machine learning technique to identify exoplanets. In Martínez-Villaseñor, L., Herrera-Alcántara, O., Ponce, H. E., and Castro-Espinoza, F., editors, *Advances in Soft Computing - 19th Mexican International Conference on Artificial Intelligence, MICAI 2020, Mexico City, Mexico, October 12-17, 2020, Proceedings, Part I*, volume 12468 of *Lecture Notes in Computer Science*, pages 50–64. Springer.

[Jara-Maldonado et al., 2020b] Jara-Maldonado, M. et al. (2020b). Transiting exoplanet discovery using machine learning techniques: A survey. *Earth Science Informatics*, 13(3):573–600.

[Juárez-Guerra et al., 2020] Juárez-Guerra, E., Alarcon-Aquino, V., Gomez-Gil, P., Ramirez-Cortes, J., and Garca-Trevio, E. (2020). A new wavelet-based neural network for classification of epileptic-related states using eeg. *Journal of Signal Processing Systems*, 92.

[Koch et al., 2010] Koch, D. G. et al. (2010). Kepler Mission Design, Realized Photometric Performance, and Early Science. ApJ, 713:L79–L86.

[Kovacs et al., 2002] Kovacs, G., Zucker, S., and Mazeh, T. (2002). A box-fitting algorithm in the search for periodic transits. *Astronomy and Astrophysics*, 391.

[Kreidberg, 2015] Kreidberg, L. (2015). batman: BAsic transit model cAlculatioN in python. *Publications of the Astronomical Society of the Pacific*, 127(957):1161–1165.

[Li et al., 2019] Li, Z., Rios, A. L. G., Xu, G., and Trajkovi, L. (2019). Machine learning techniques for classifying network anomalies and intrusions. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.

[Mallat, 2008] Mallat, S. (2008). *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* Academic Press, Inc., USA, 3rd edition.

[Mallat, 1989] Mallat, S. G. (1989). Multiresolution approximations and wavelet orthonormal bases of l2(r). *Transactions of the American Mathematical Society*, 315(1):69–87.

[Mandic et al., 2013] Mandic, D. P., u. Rehman, N., Wu, Z., and Huang, N. E. (2013). Empirical Mode Decomposition-based time-frequency analysis of multivariate signals: The power of adaptive data analysis. *IEEE Signal Processing Magazine*, 30(6):74–86.

[Masciadri and Raga, 2004] Masciadri, E. and Raga, A. (2004). Exoplanet recognition using a wavelet analysis technique. *The Astrophysical Journal*, 611.

[Mayor and Queloz, 1995] Mayor, M. and Queloz, D. (1995). A Jupiter-mass companion to a solar-type star. Nature, 378:355–359.

[McCauliff et al., 2015] McCauliff, S. D. et al. (2015). Automatic classification of Kepler planetary transit candidates. *The Astrophysical Journal*, 806(1):6.

[Mislis et al., 2018] Mislis, D., Pyrzas, S., and Alsubai, K. A. (2018). TSARDI: a Machine Learning data rejection algorithm for transiting exoplanet light curves. *Monthly Notices of the Royal Astronomical Society*, 481(2):1624–1630.

[Nason and Silverman, 1995] Nason, G. P. and Silverman, B. W. (1995). *The Stationary Wavelet Transform and some Statistical Applications*, pages 281–299. Springer New York, New York, NY.

[Newton and Henry, 2000] Newton, I. and Henry, R. C. (2000). Circular motion. *American Journal of Physics*, 68(7):637–639.

[Osborn, H. P. et al., 2020] Osborn, H. P. et al. (2020). Rapid classification of tess planet candidates with convolutional neural networks. *A&A*, 633:A53.

[Pearson et al., 2017] Pearson, K., Palafox, L., and Griffith, C. (2017). Searching for exoplanets using artificial intelligence. *Monthly Notices of the Royal Astronomical Society*, 474.

[Pearson, 2019] Pearson, K. A. (2019). A search for multiplanet systems with TESS using a bayesian n-body retrieval and machine learning. *The Astronomical Journal*, 158(6):243.

[Petigura et al., 2013] Petigura, E. A., Marcy, G. W., and Howard, A. W. (2013). A plateau in the planet population below twice the size of earth. *The Astrophysical Journal*, 770(1):69.

[Priyadarshini and Puri, 2021] Priyadarshini, I. and Puri, V. (2021). A convolutional neural network (cnn) based ensemble model for exoplanet detection. *Earth Science Informatics*, 14:1–13.

[Ricker et al., 2014] Ricker, G. R. et al. (2014). Transiting exoplanet survey satellite. *Journal of Astronomical Telescopes, Instruments, and Systems.*

[Schanche et al., 2018] Schanche, N. et al. (2018). Machine-learning approaches to exoplanet transit detection and candidate validation in wide-field ground-based surveys. *Monthly Notices of the Royal Astronomical Society*, 483(4):5534–5547.

[Seager and Bains, 2015] Seager, S. and Bains, W. (2015). The search for signs of life on exoplanets at the interface of chemistry and planetary science. *Science Advances*, 1(2):e1500047.

[Shallue and Vanderburg, 2018] Shallue, C. J. and Vanderburg, A. (2018). Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *The Astronomical Journal*, 155(2):94.

[Statistics and Breiman, 2001] Statistics, L. B. and Breiman, L. (2001). Random forests. In *Machine Learning*, pages 5–32.

[Sturrock et al., 2019] Sturrock, G. C. et al. (2019). Machine learning pipeline for exoplanet classification. SMU Data Science Review.

[Theodoridis and Koutroumbas, 2008] Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition, Fourth Edition.* Academic Press, Inc., USA, 4th edition.

[Thompson et al., 2015] Thompson, S. E. et al. (2015). A machine learning technique to identify transit shaped signals. *The Astrophysical Journal*, 812:46.

[Torres et al., 2011] Torres, M. E., Colominas, M. A., Schlotthauer, G., and Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4144–4147.

[Treu et al., 2012] Treu, T., Marshall, P. J., and Clowe, D. (2012). Resource Letter: Gravitational Lensing. *Am. J. Phys.*, 80:753.

[Vázquez, 1981] Vázquez, M. (1981). *Derecho y política en el espacio cósmico.* Instituto de Investigaciones Juridicas: Serie H, Estudios de derecho internacional público. Universidad Nacional Autónoma de México, Instituto de Investigaciones Jurídicas.

[Verma et al., 2014] Verma, A., singh, T. N., and Maheshwar, S. (2014). Comparative study of intelligent prediction models for pressure wave velocity. *Journal of Geosciences and Geomatics*, 2(3):130–138.

[Wolberg and Mangasarian, 1990] Wolberg, W. H. and Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Pro-*

ceedings of the National Academy of Sciences of the United States of America, 87:9193 – 9196.

[Wolszczan and Frail, 1992] Wolszczan, A. and Frail, D. A. (1992). A planetary system around the millisecond pulsar PSR1257 + 12. Nature, 355:145–147.

[Wright, 2017] Wright, J. T. (2017). Radial velocities as an exoplanet discovery method. *Handbook of Exoplanets*, pages 1–13.

[Yaqoob, 2011] Yaqoob, T. (2011). *Exoplanets and Alien Solar Systems.* New Earth Labs.

[Yu et al., 2021] Yu, C. et al. (2021). A survey on machine learning based light curve analysis for variable astronomical sources. *WIREs Data Mining and Knowledge Discovery*, 11(5):e1425.

[Yu et al., 2019] Yu, L., Vanderburg, A., Huang, C., Shallue, C. J., Crossfield, I. J. M., Gaudi, B. S., Daylan, T., Dattilo, A., Armstrong, D. J., Ricker, G. R., Vanderspek, R. K., Latham, D. W., Seager, S., Dittmann, J., Doty, J. P., Glidden, A., and Quinn, S. N. (2019). Identifying exoplanets with deep learning. III. automated triage and vetting of TESS candidates. *The Astronomical Journal*, 158(1):25.

[Zeiler et al., 2010] Zeiler, A. et al. (2010). Empirical Mode Decomposition - an introduction. *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8.

[Zhang, 1992] Zhang, Q. (1992). *Wavelet Networks: The Radial Structure and an Efficient Initialization Procedure.* LiTH-ISY-I.: Institutionen för Systemteknik. Linköping Univ.

[Zhang and Benveniste, 1992] Zhang, Q. and Benveniste, A. (1992). Wavelet networks. *Trans. Neur. Netw.*, 3(6):889898.

Appendix A

Publications

The different publications that were produced during the elaboration of this work are presented in this appendix.

A.1. Journal Citation Reports Articles

- An article that describes the design and implementation of the WABBLES model:
  Miguel Jara-Maldonado, Vicente Alarcon-Aquino, Roberto Rosas-Romero, A new machine learning model based on the broad learning system and wavelets, *Engineering Applications of Artificial Intelligence*, Volume 112, 2022, 104886, ISSN 0952-1976. `https://doi.org/10.1016/j.engappai.2022.104886`

- An article that gathers three contributions: the synthesis of the related work of ML applied to exoplanet discovery; a pipeline for synthetic LC dataset creation; and a comparative framework of the different ML models that can be used for exoplanet identification, along with the use of the DWT to enhance their performance:
  Jara-Maldonado, M., Alarcon-Aquino, V., Rosas-Romero, R., Starostenko, O., & Ramirez-Cortes, J. M. (2020). Transiting Exoplanet Discovery Using Machine Learning Techniques: A Survey. *Earth Science Informatics*. `https://doi.org/10.1007/s12145-020-00464-7`

A.2. Conference Articles and Posters

- An article that proves the usefulness of the EMD and EEMD techniques to improve the robustness of the ML models of exoplanet identification:

  Jara-Maldonado M., Alarcon-Aquino V., Rosas-Romero R. (2020) A Multiresolution Machine Learning Technique to Identify Exoplanets. In: Martnez-Villaseor L., Herrera-Alcntara O., Ponce H., Castro-Espinoza F.A. (eds) *Advances in Soft Computing. MICAI 2020.* Lecture Notes in Computer Science, vol 12468. Springer, Cham. `https://doi.org/10.1007/978-3-030-60884-2_4`

- A summary of the work done in [Jara-Maldonado et al., 2020b]:

  Jara-Maldonado, M., Alarcon-Aquino, V., & Rosas-Romero, R. (2020). A Multiresolution Analysis Technique to Improve Exoplanet Identification, poster presented at: Exoplanets III; Heidelberg (Online).

Appendix B

Discrete Wavelet Transform Results

This appendix presents the plots that correspond to all the accuracy results obtained using the DWT technique to preprocess the LCs. The resulting coefficients were then used as inputs for the ML models. The figures include both the cDs and cAs of the DWT, along with the different decomposition levels tested. The wavelets used were the bior2.4, coif5, db1, db5, and sym5 wavelets. These figures allow one to observe that there are several models that have a better performance with cDs, as is the case of the CNN (see Fig. B.1); while others, such as the RF classifier with cAs (see Fig. B.13). Even more, this is sometimes switched depending on the dataset that was used, see the case of the CNN in Figs. B.1 and B.12. We attribute this differences to the nature of the different coefficients used. For instance, the cAs are always positive, while the cDs can be both positive and negative. Also, since the cDs represent the high frequencies of the signal, this could mean that the classifiers that obtained better results using the cDs are basing on the transit location; while those that obtain better results with the cAs are basing on the frequency that better represents the transit, because cAs have a good frequency representation. These figures also allow the observer to locate the best decomposition level for each different ML approach. Notice how the CNN works better with lower decomposition levels. Recall that the architecture of the CNN presented depends on the number of decomposition levels because it needs a minimum length of inputs to still work. Finally, the figures presented on this section can be used as a reference to future experiments with different wavelets because it demonstrates that there are several decomposition levels and coefficients with which each model works better.

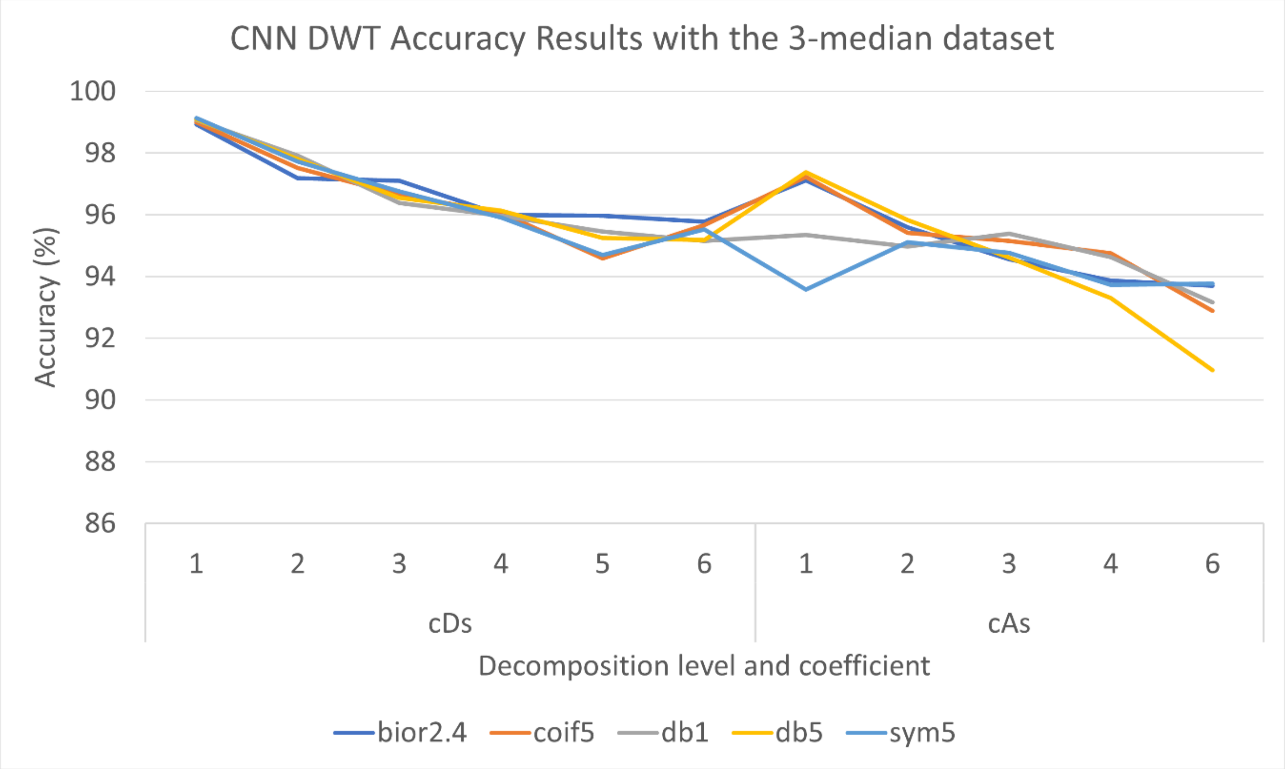


Figure B.1: Accuracy results obtained by the Convolutional Neural Network (CNN) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

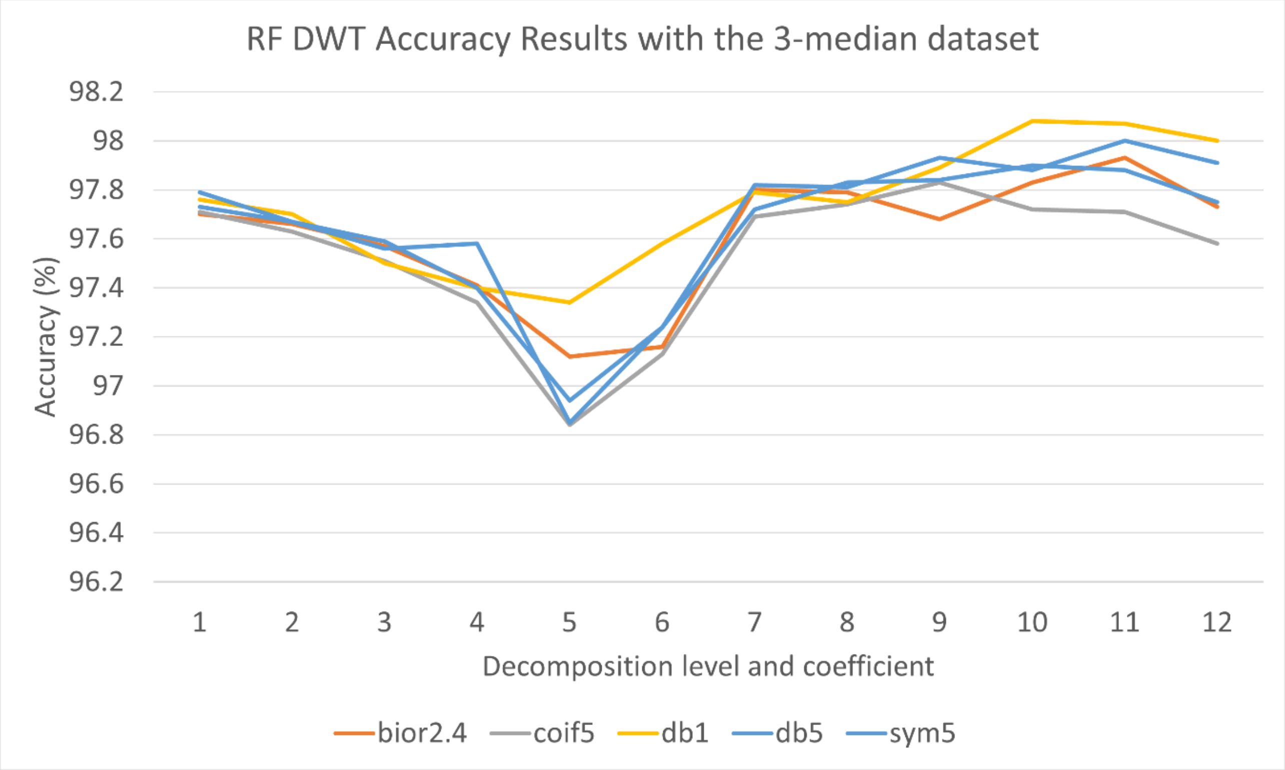


Figure B.2: Accuracy results obtained by the Random Forests (RF) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

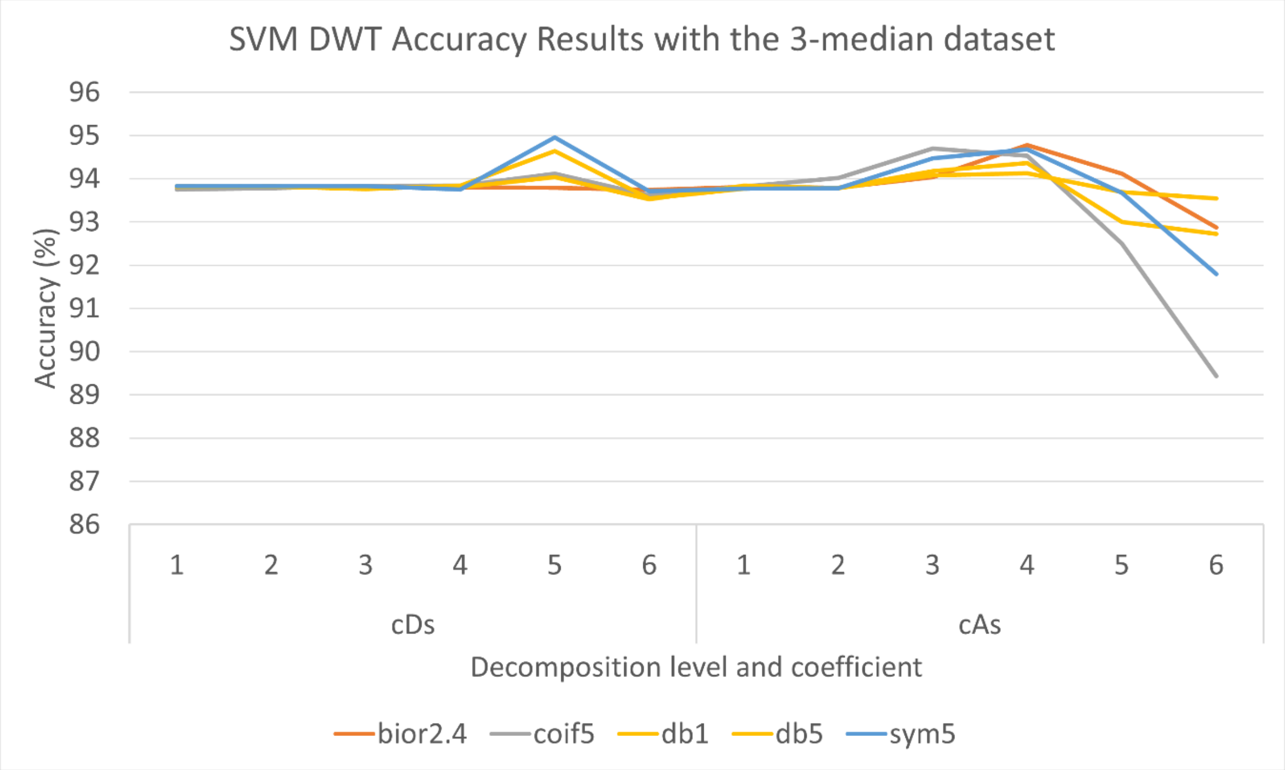


Figure B.3: Accuracy results obtained by the Support Vector Machine (SVM) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

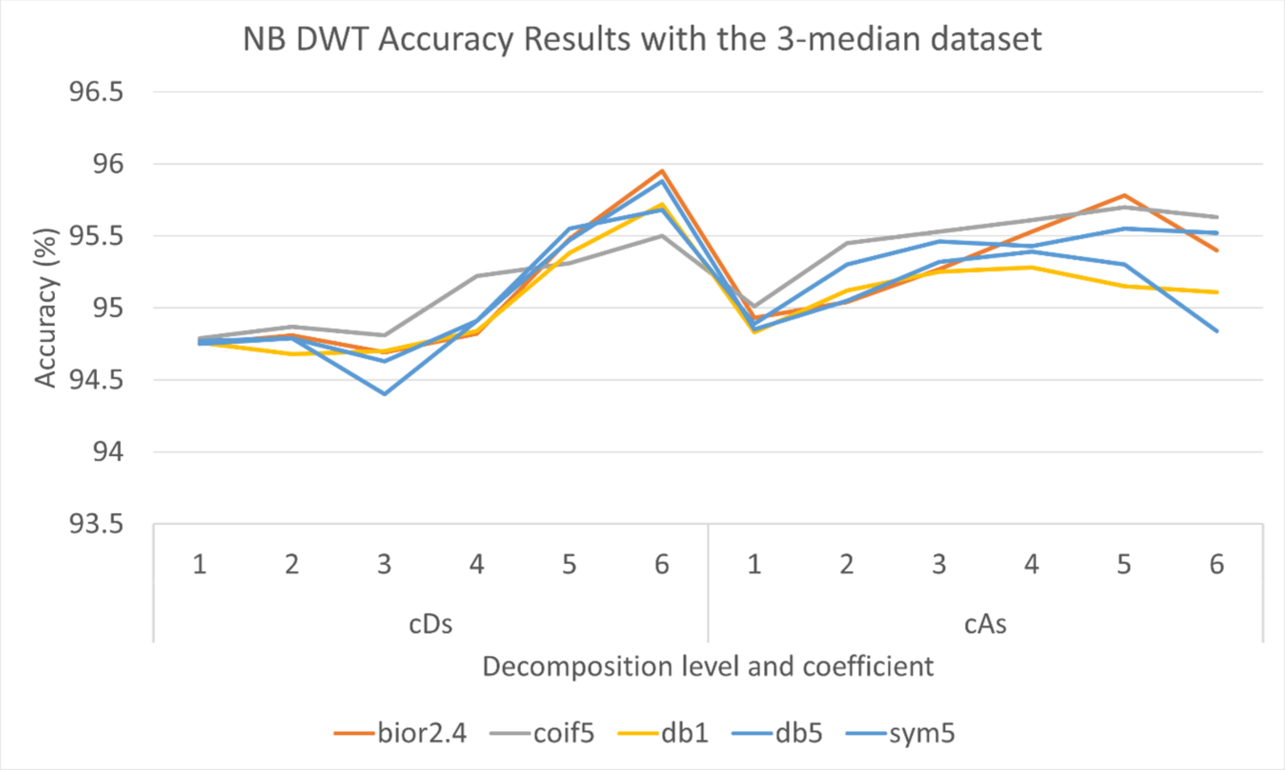


Figure B.4: Accuracy results obtained by the Naïve Bayes (NB) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

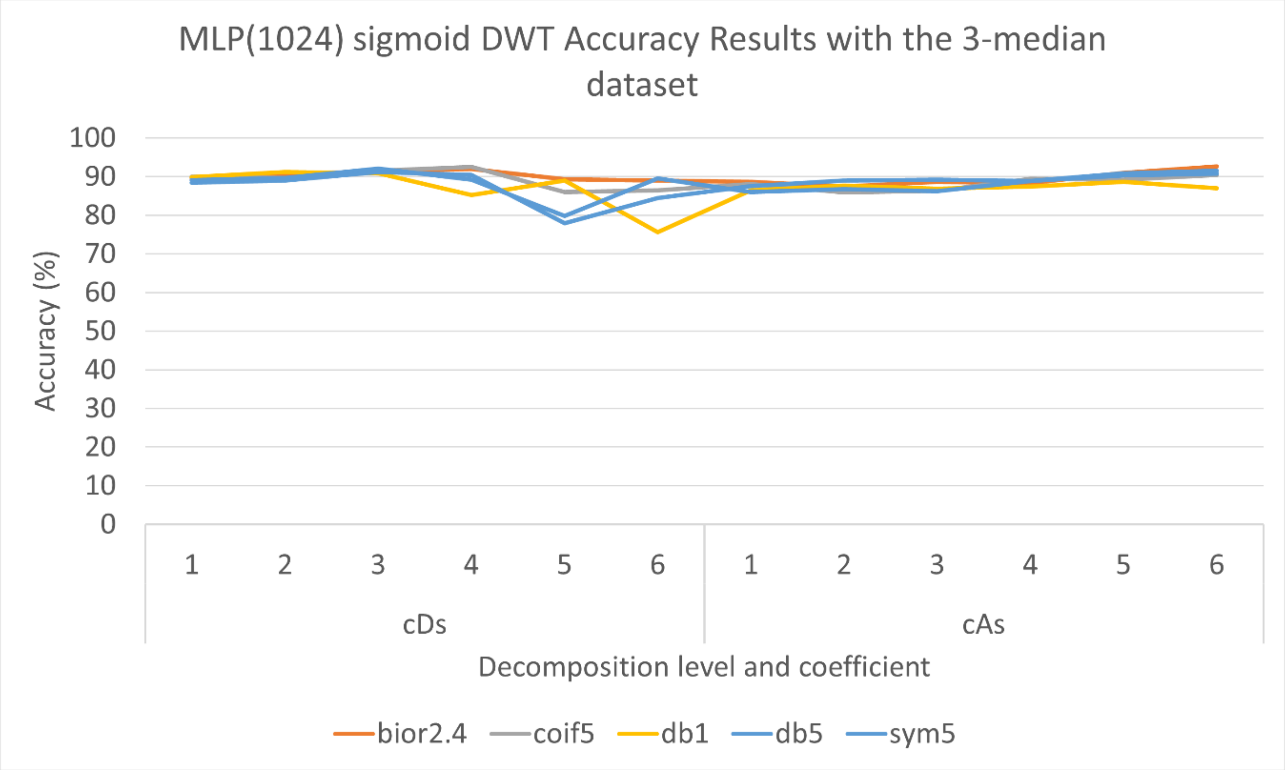


Figure B.5: Accuracy results obtained by the Sigmoid MLP(1024) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

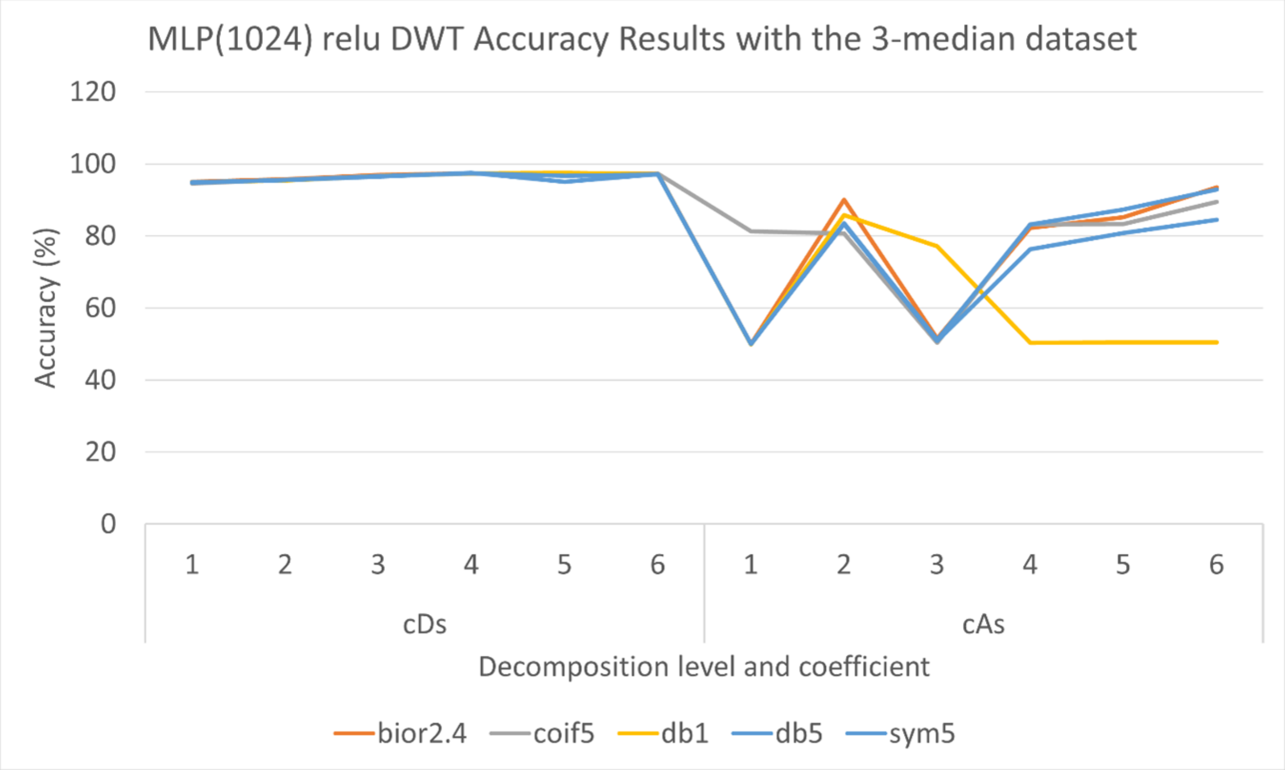


Figure B.6: Accuracy results obtained by the Relu MLP(1024) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

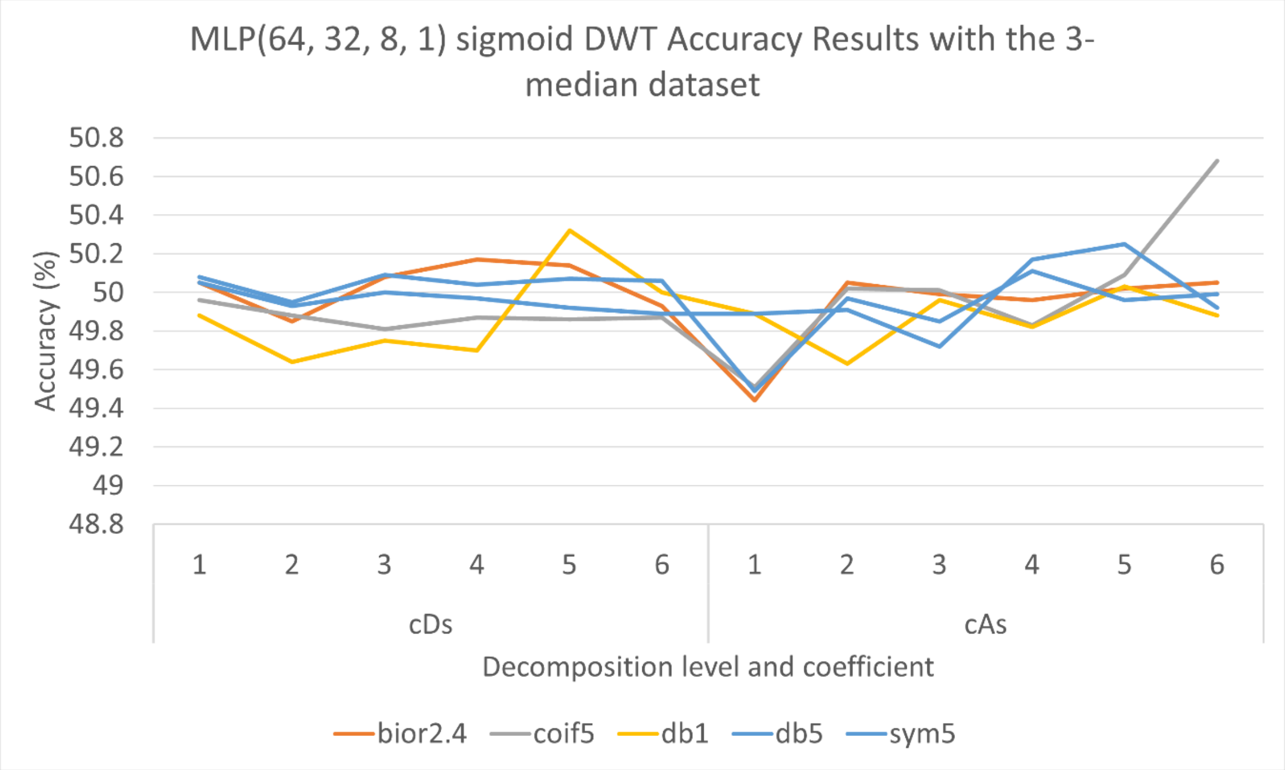


Figure B.7: Accuracy results obtained by the Sigmoid MLP(64, 32, 8, 1) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

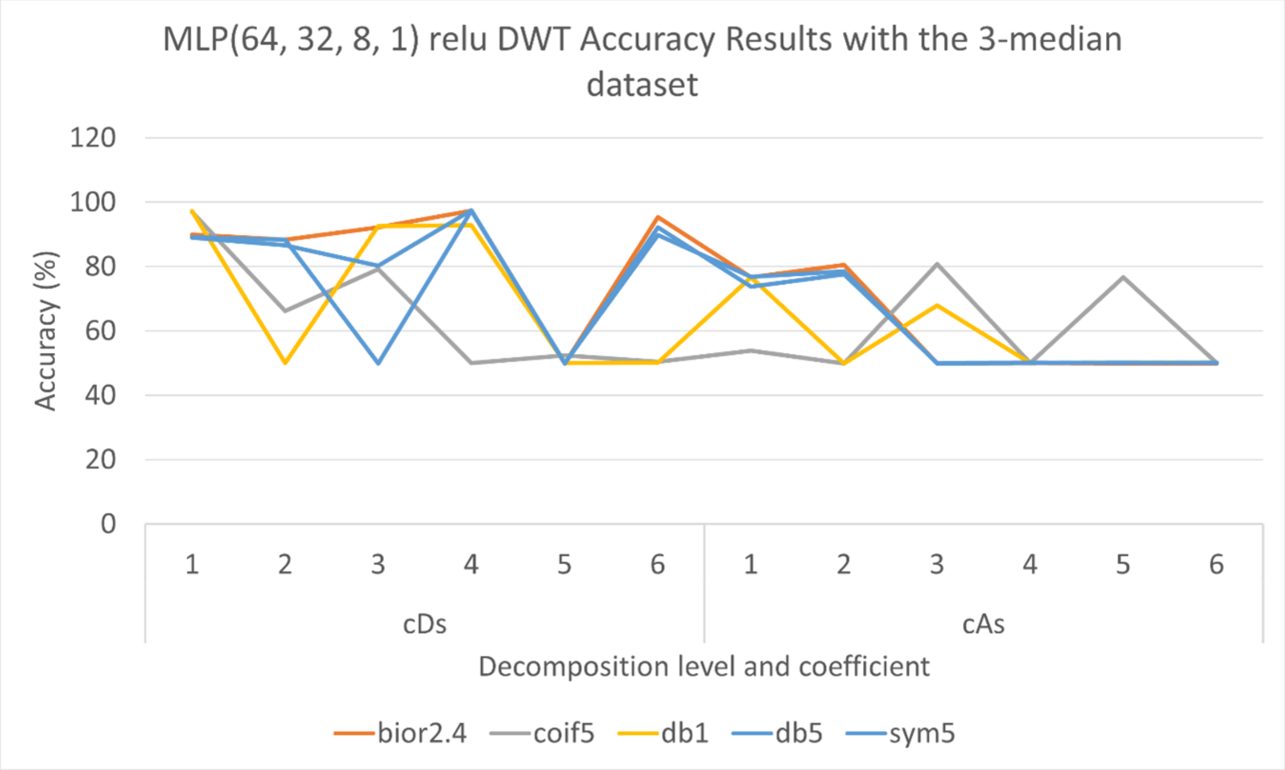


Figure B.8: Accuracy results obtained by the Relu MLP(64, 32, 8, 1) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

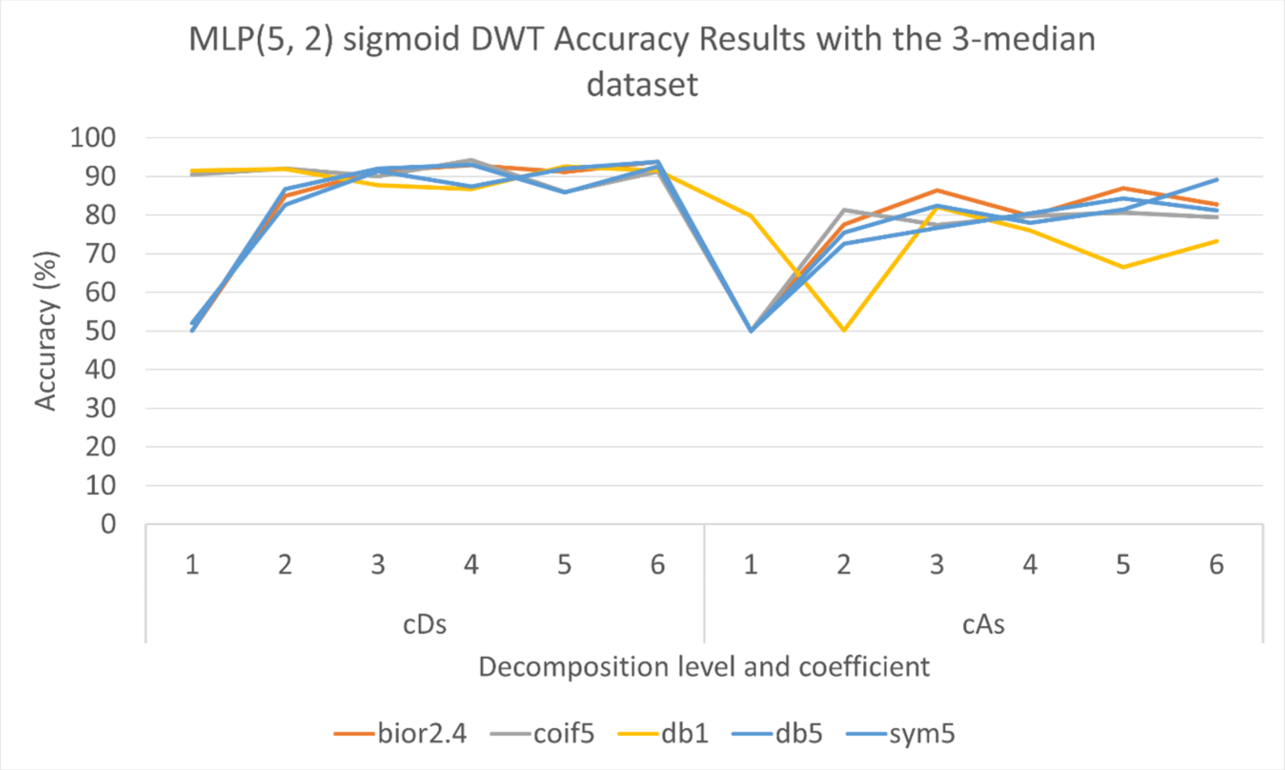


Figure B.9: Accuracy results obtained by the Sigmoid MLP(5, 2) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

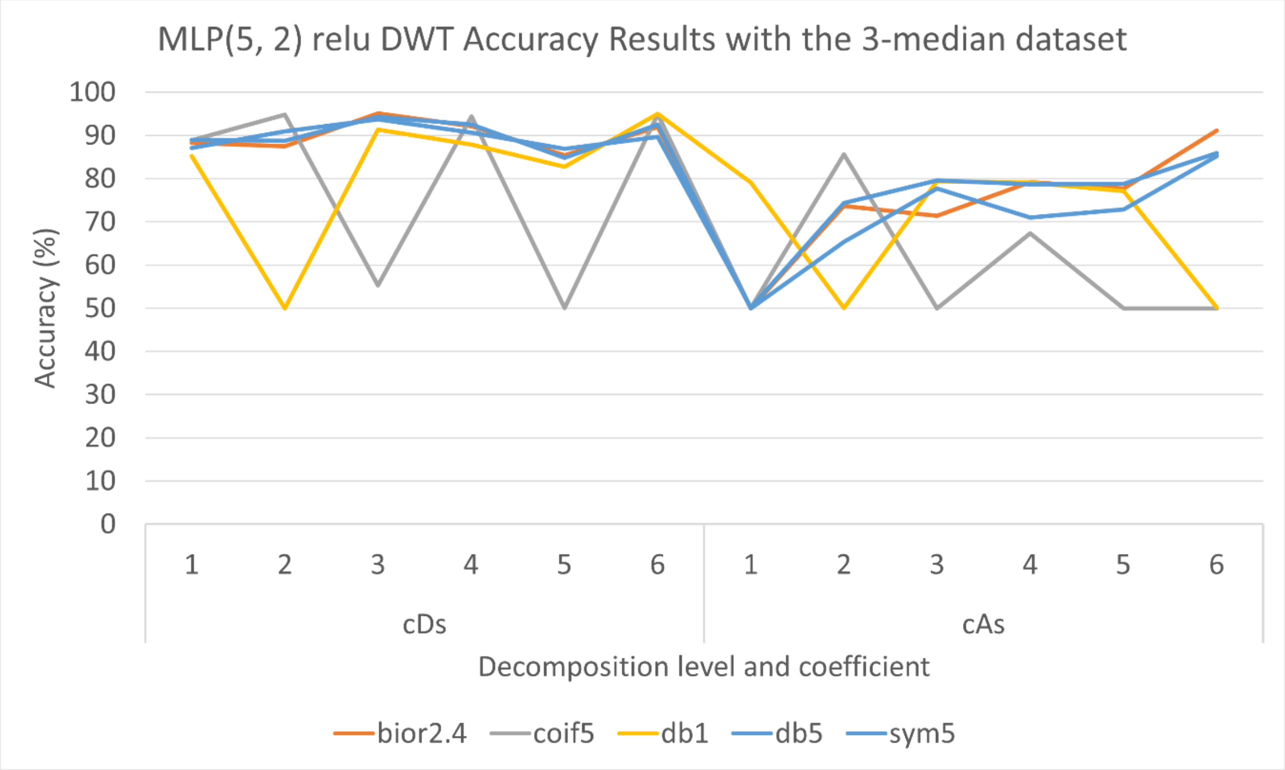


Figure B.10: Accuracy results obtained by the Relu MLP(5, 2) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

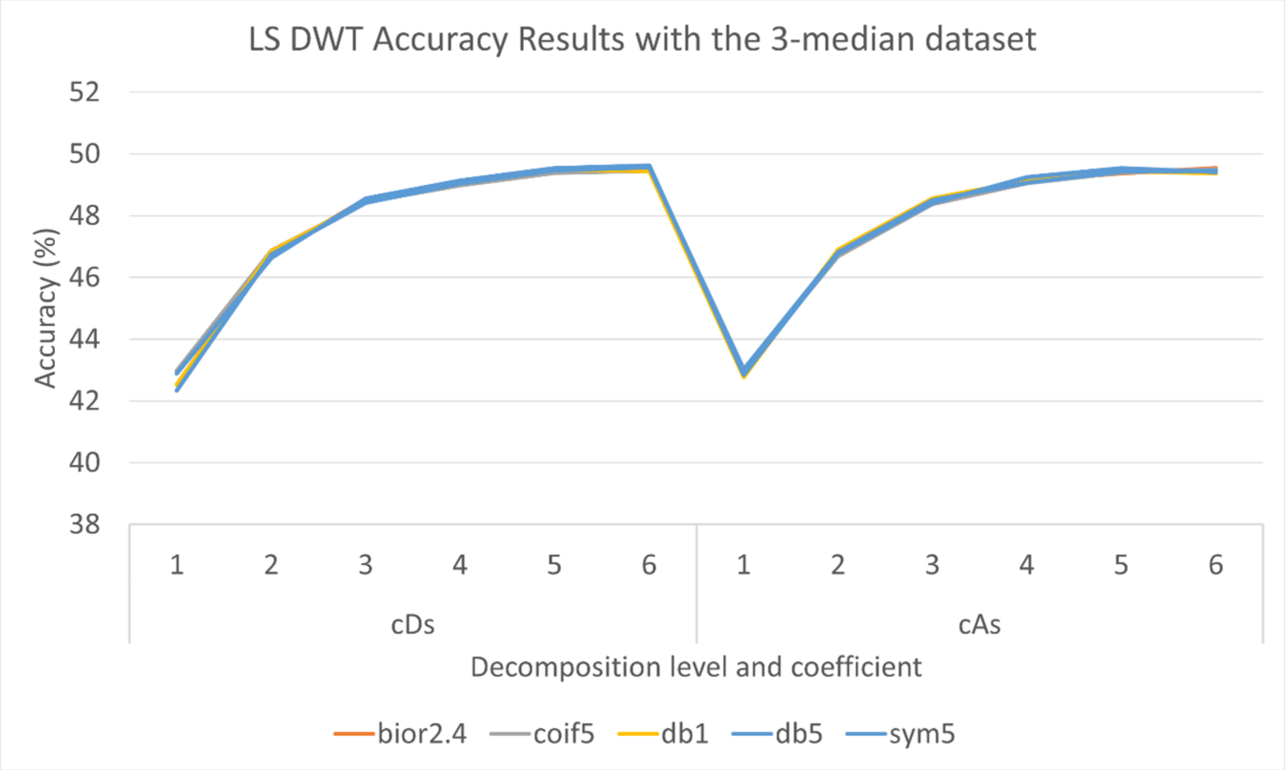


Figure B.11: Accuracy results obtained by the Least Squares (LS) model with the 3-median dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

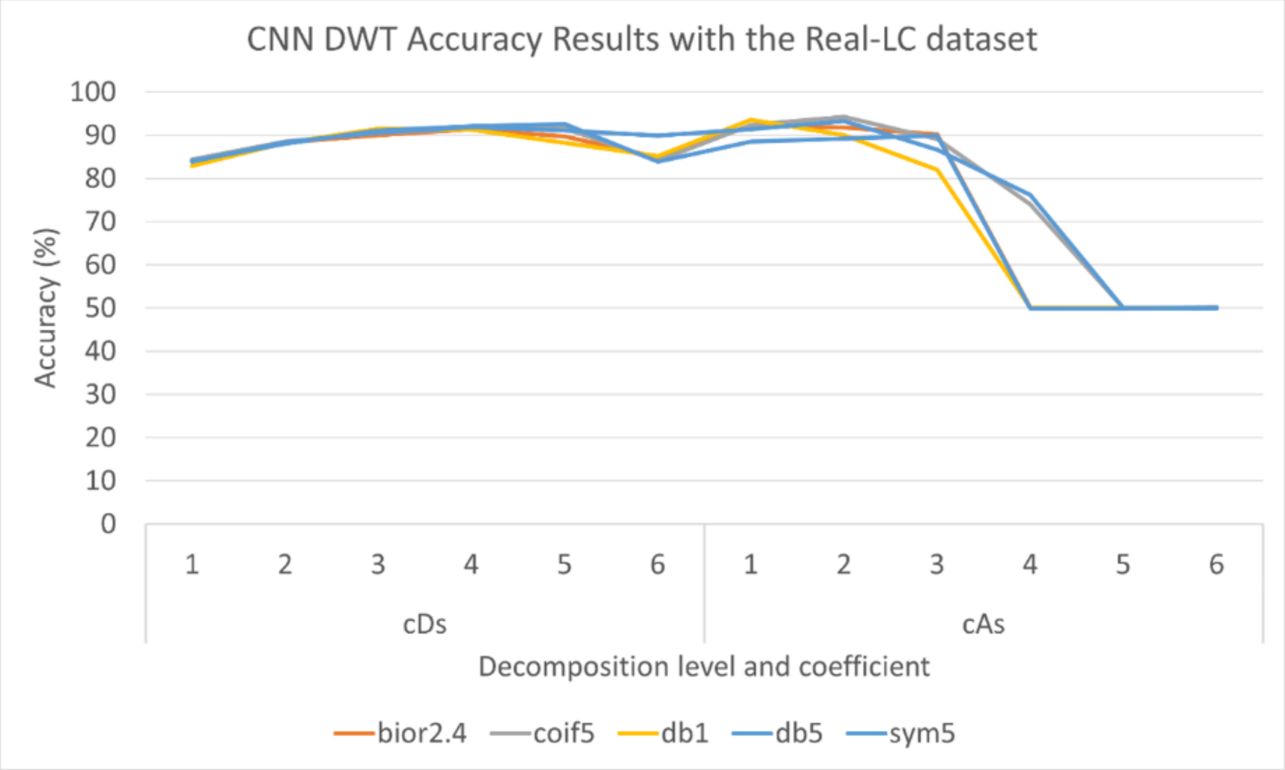


Figure B.12: Accuracy results obtained by the Convolutional Neural Network (CNN) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

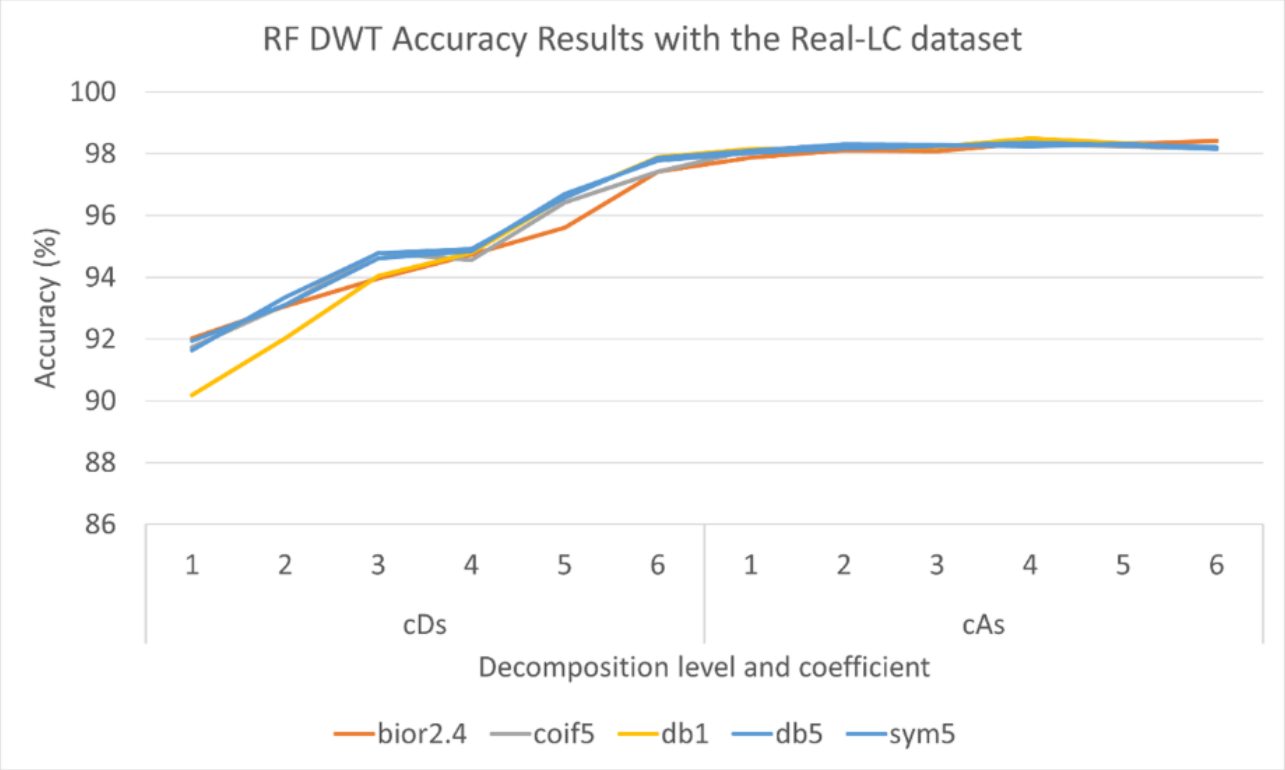


Figure B.13: Accuracy results obtained by the Random Forests (RF) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

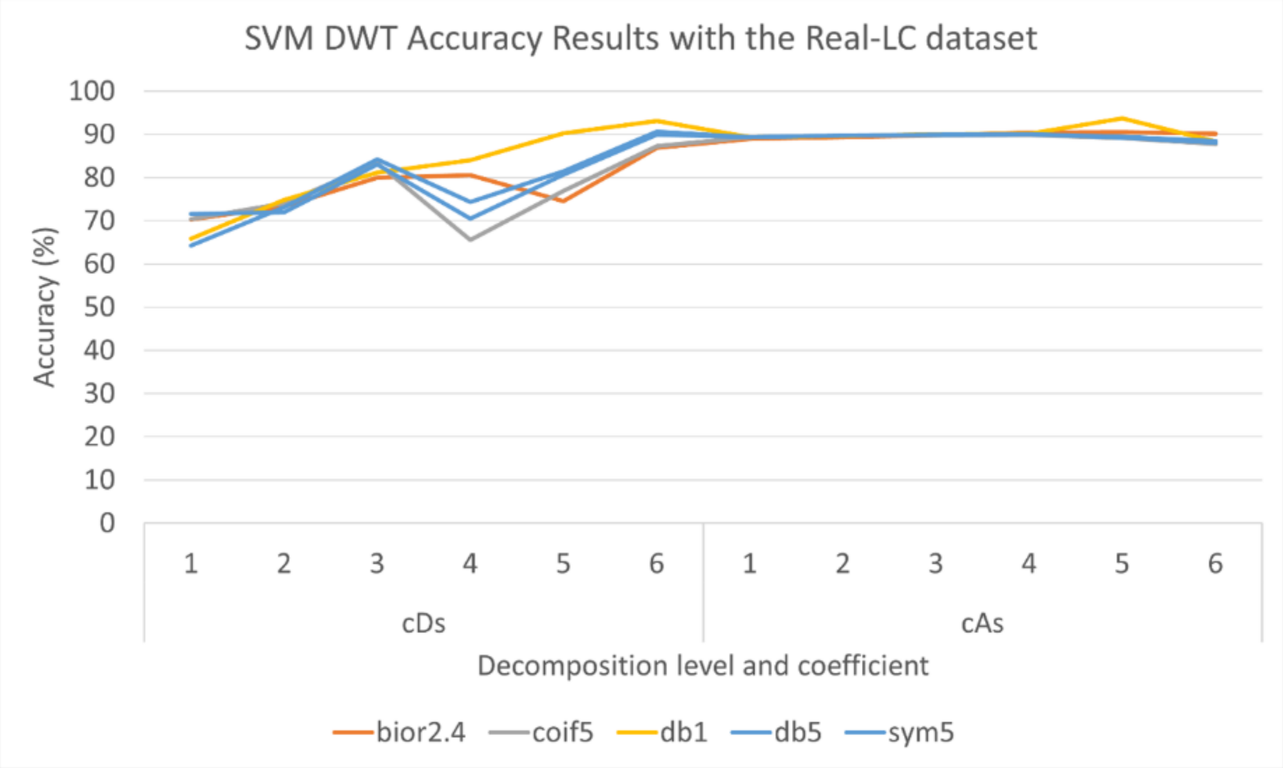


Figure B.14: Accuracy results obtained by the Support Vector Machine (SVM) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

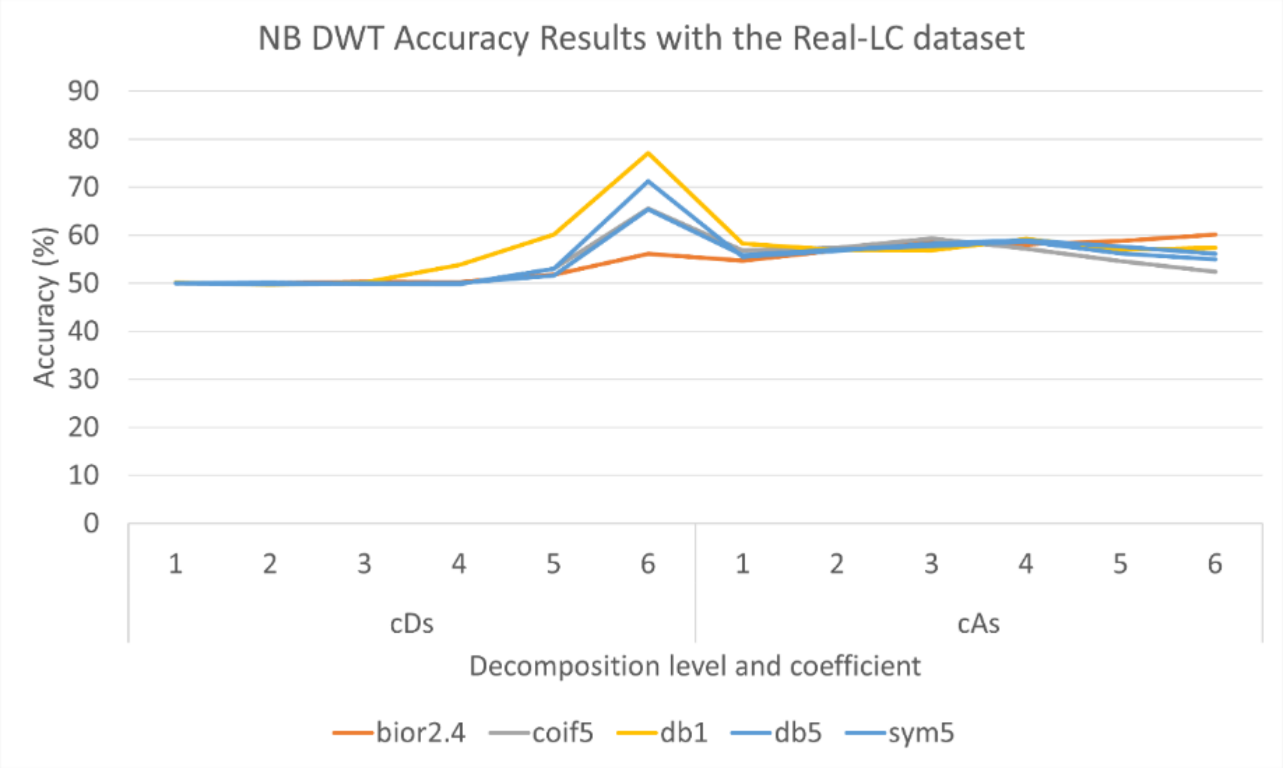


Figure B.15: Accuracy results obtained by the Naïve Bayes (NB) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

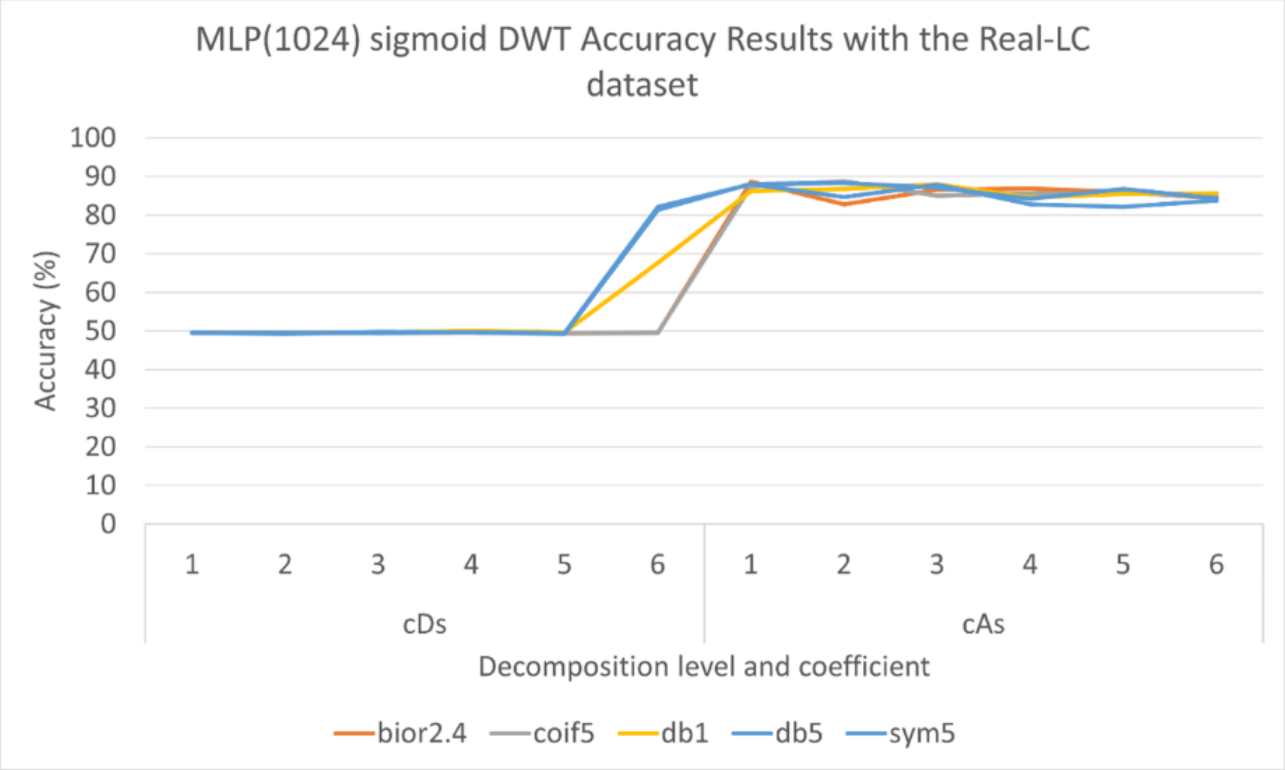


Figure B.16: Accuracy results obtained by the Sigmoid MLP(1024) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

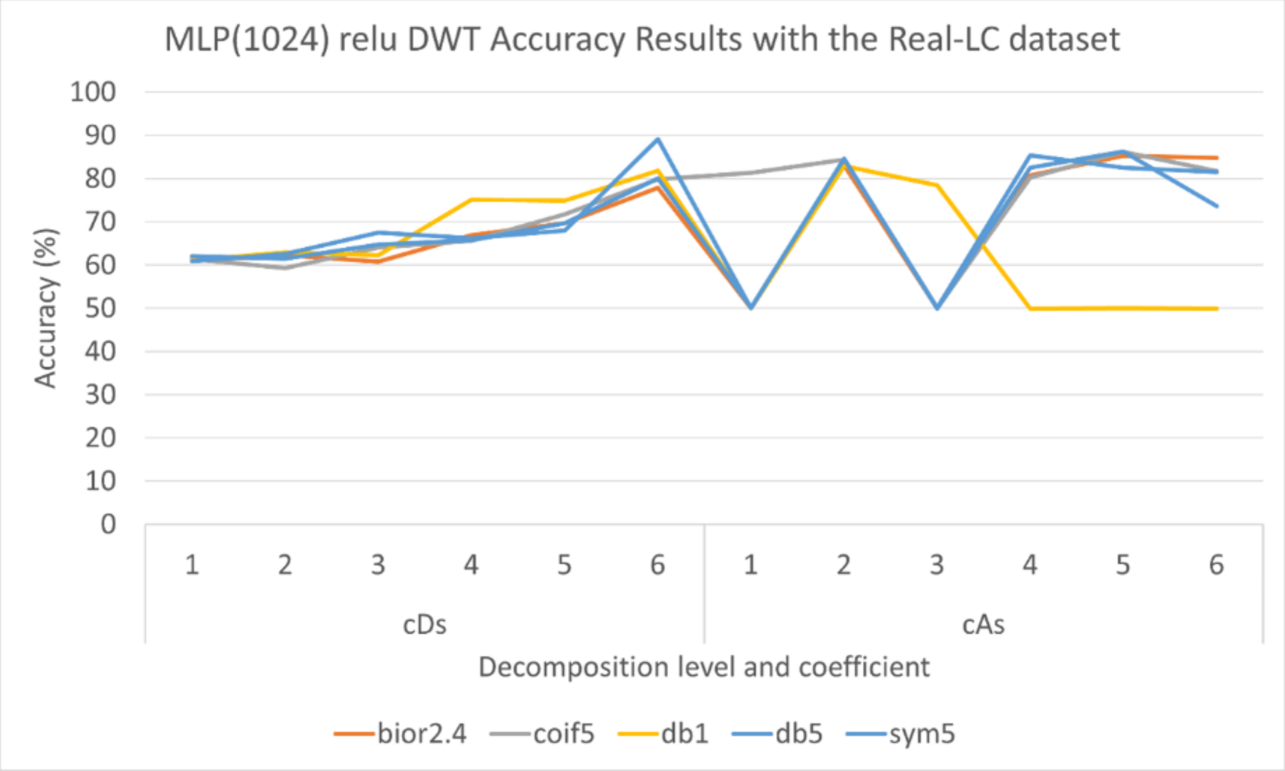


Figure B.17: Accuracy results obtained by the Relu MLP(1024) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

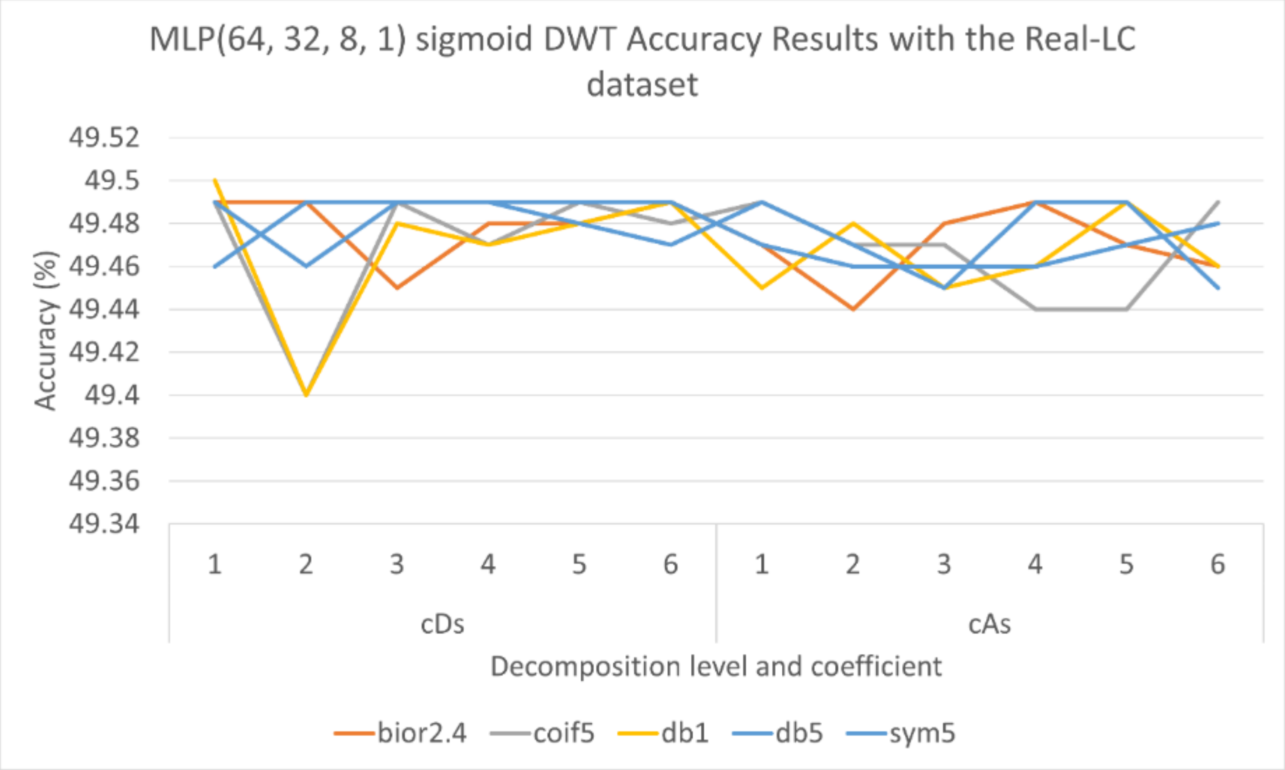


Figure B.18: Accuracy results obtained by the Sigmoid MLP(64, 32, 8, 1) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

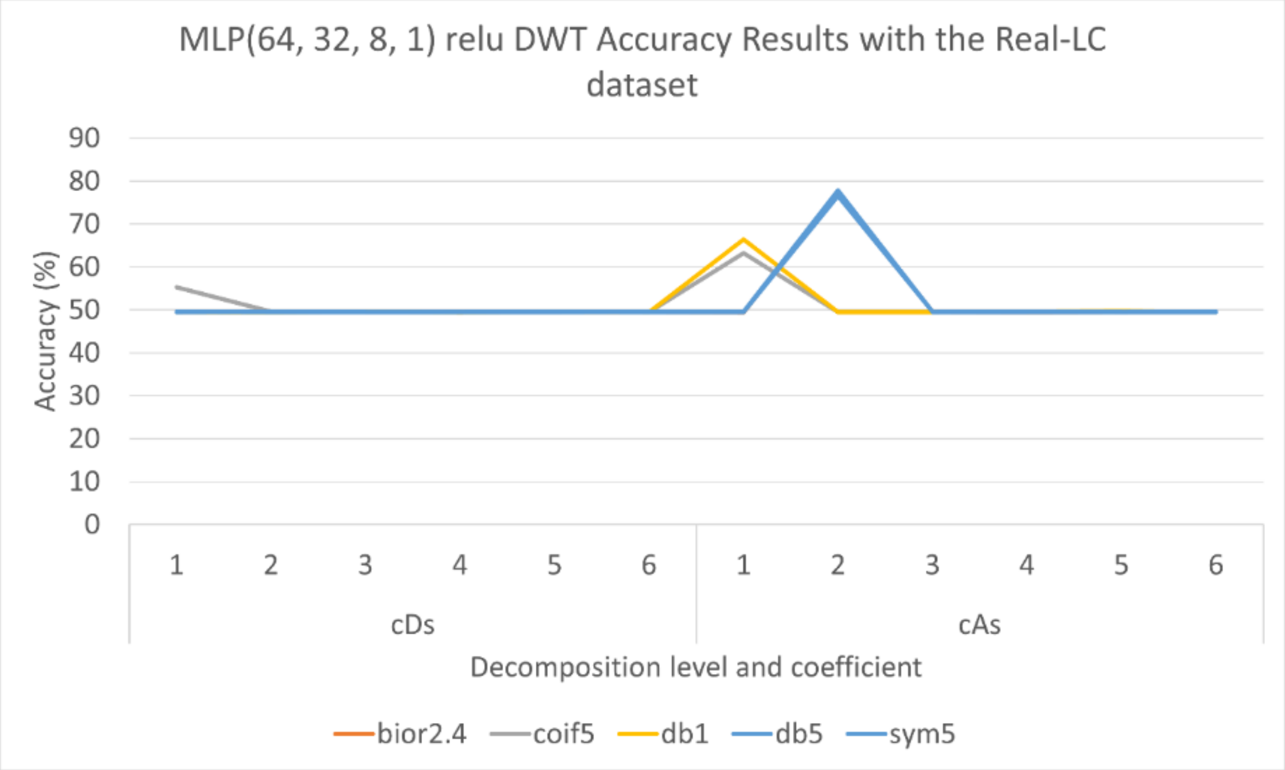


Figure B.19: Accuracy results obtained by the Relu MLP(64, 32, 8, 1) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

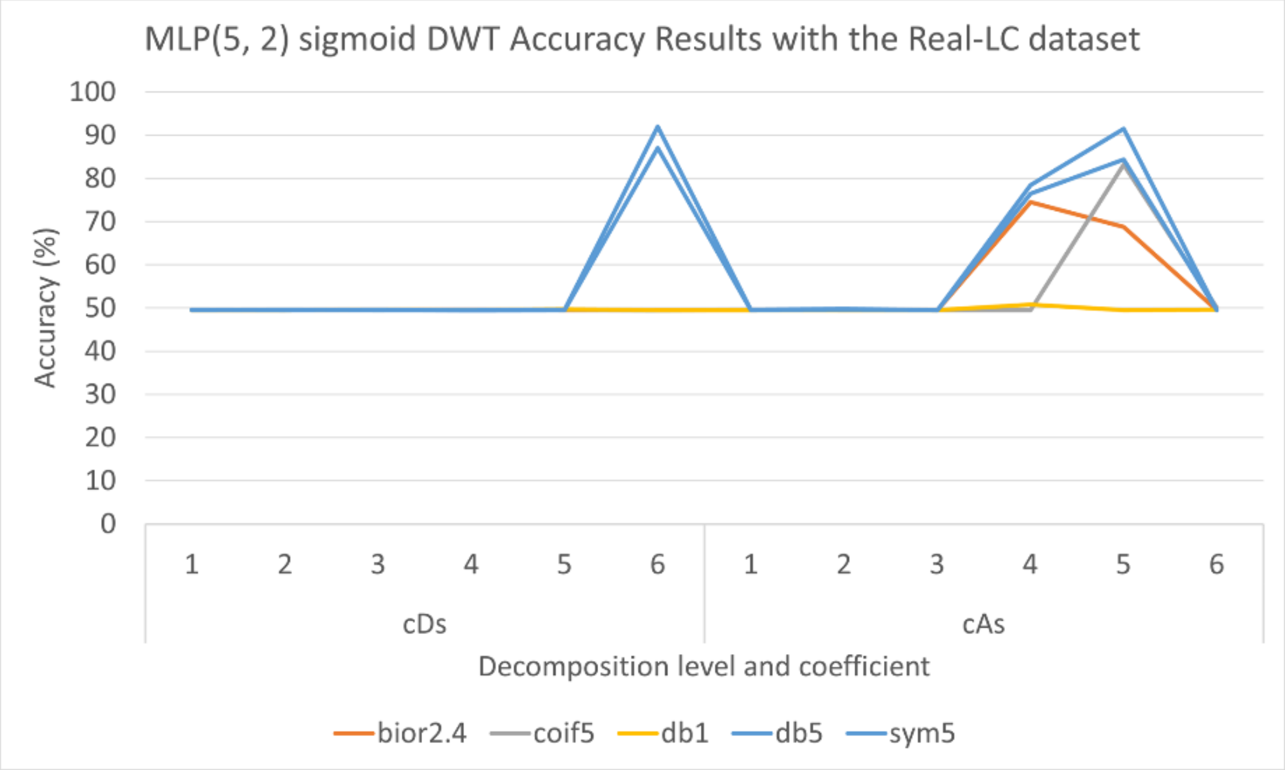


Figure B.20: Accuracy results obtained by the Sigmoid MLP(5, 2) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

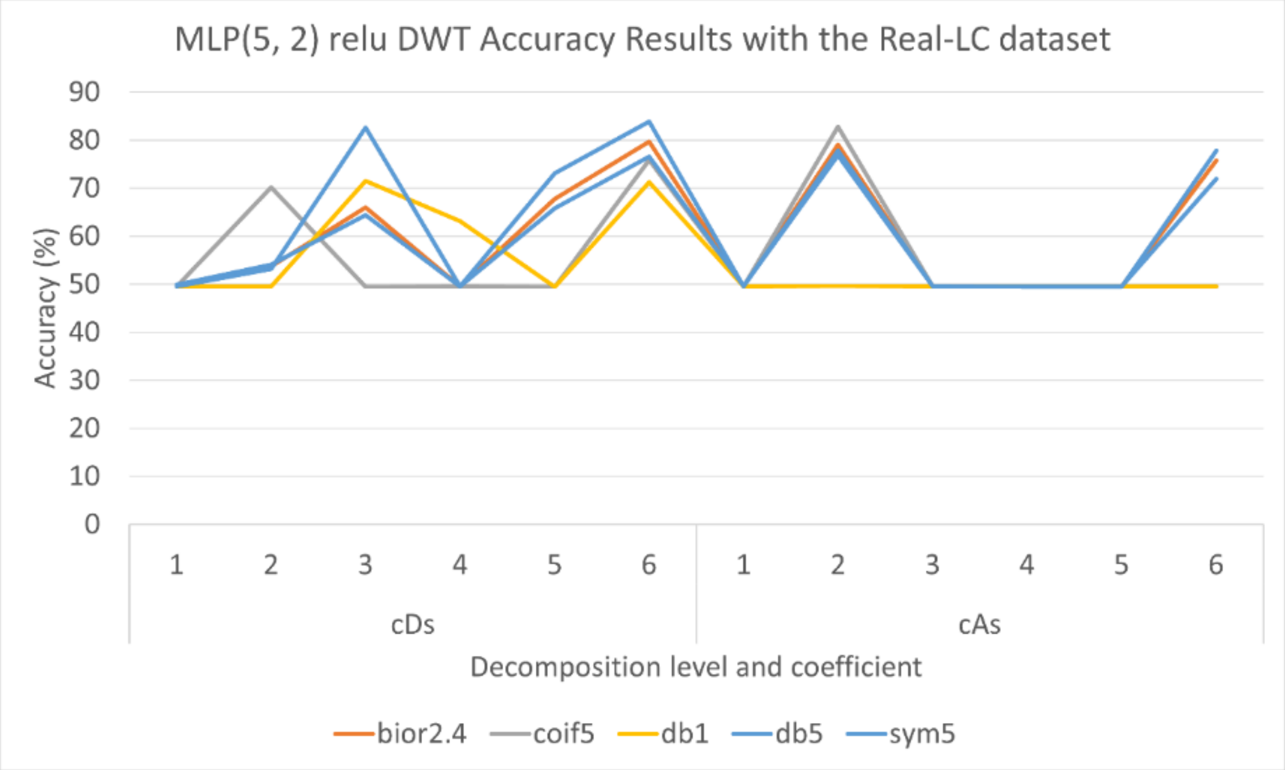


Figure B.21: Accuracy results obtained by the Relu MLP(5, 2) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

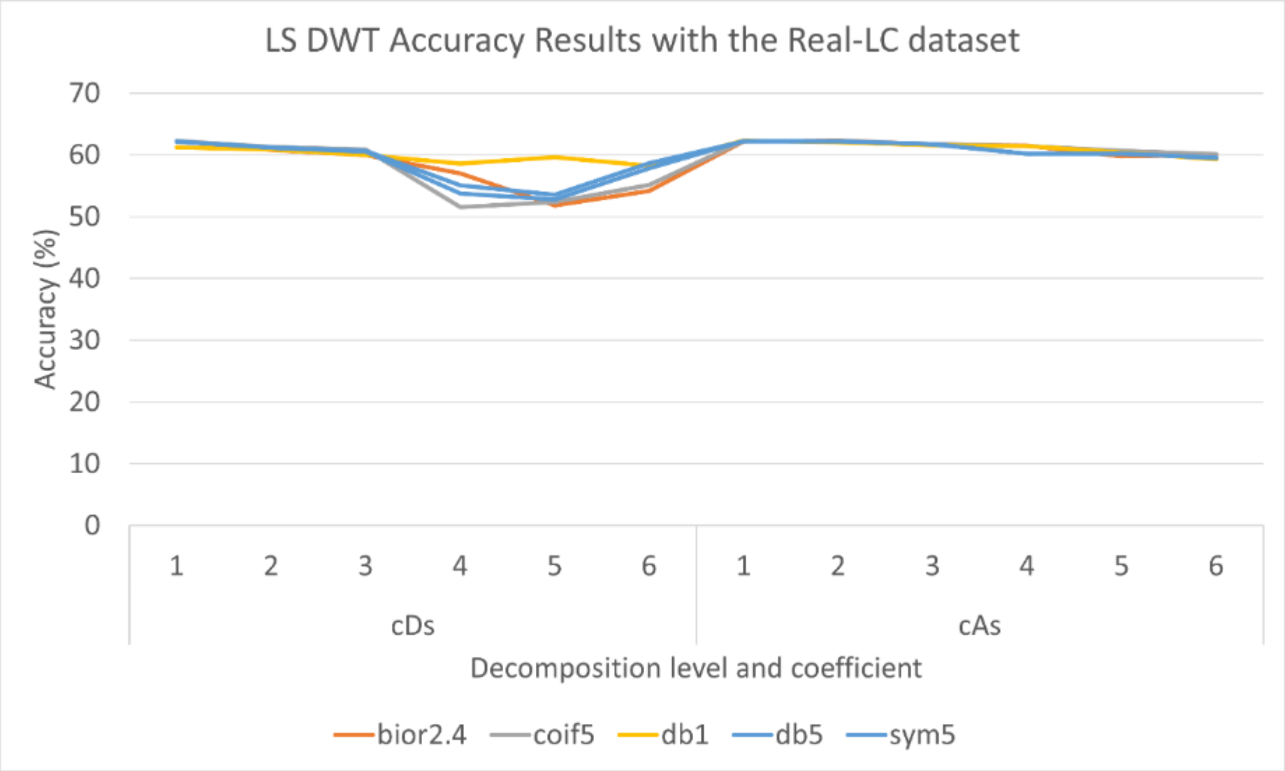


Figure B.22: Accuracy results obtained by the Least Squares (LS) model with the Real-LC dataset preprocessed using the Discrete Wavelet Transform (DWT) technique.

Appendix C

Empirical Mode Decomposition and Ensemble Empirical Mode Decomposition Results

This appendix presents the plots that correspond to all the results obtained using the EMD and EEMD techniques to preprocess the LCs. The IMFs obtained were then used as inputs for the ML models, where up to 10 modes were tested. As it can be seen in Figs. C.2, C.3, and C.4, in most cases the best results are obtained between the mode 5 and 7. This is an important finding because for future experiments the other IMFs could be ignored, saving thus execution time. This also means that the exoplanet transits are better represented within these modes of the signal.

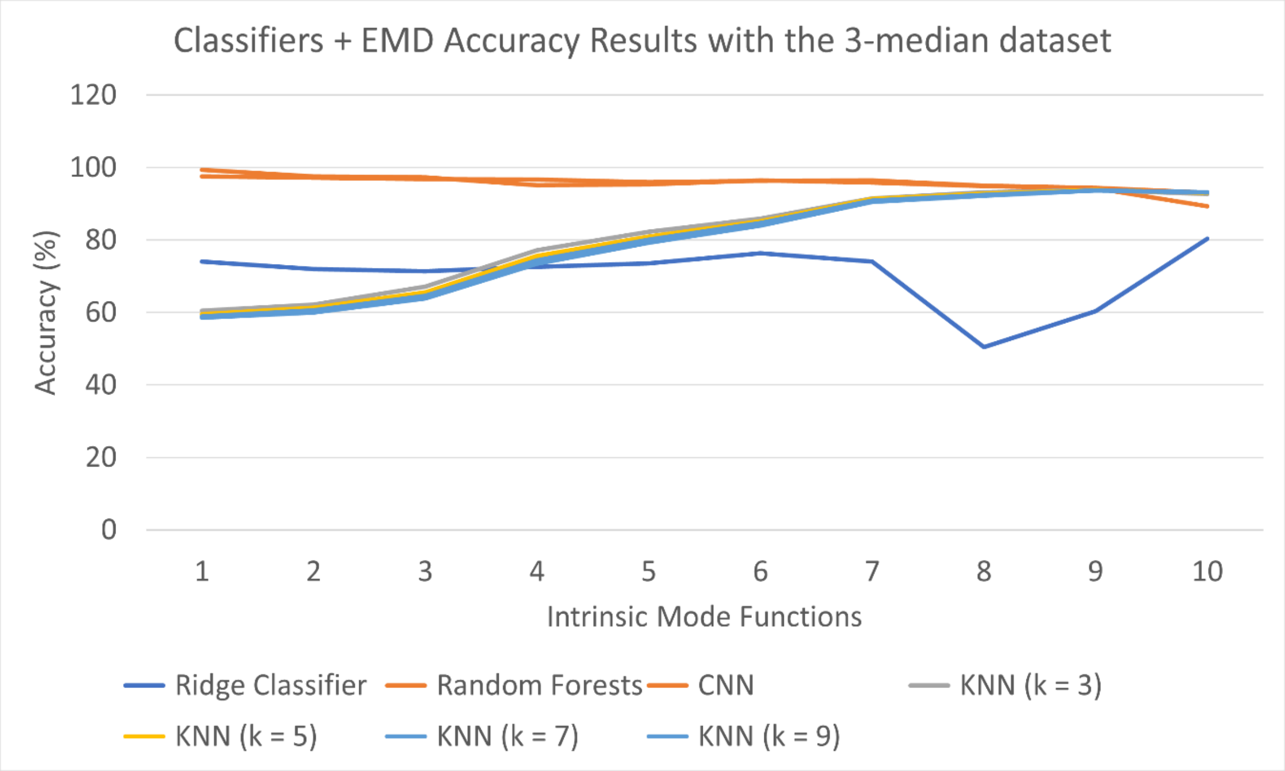


Figure C.1: Accuracy results obtained using light curves from the 3-median dataset which were preprocessed with the Empirical Mode Decomposition (EMD) technique.

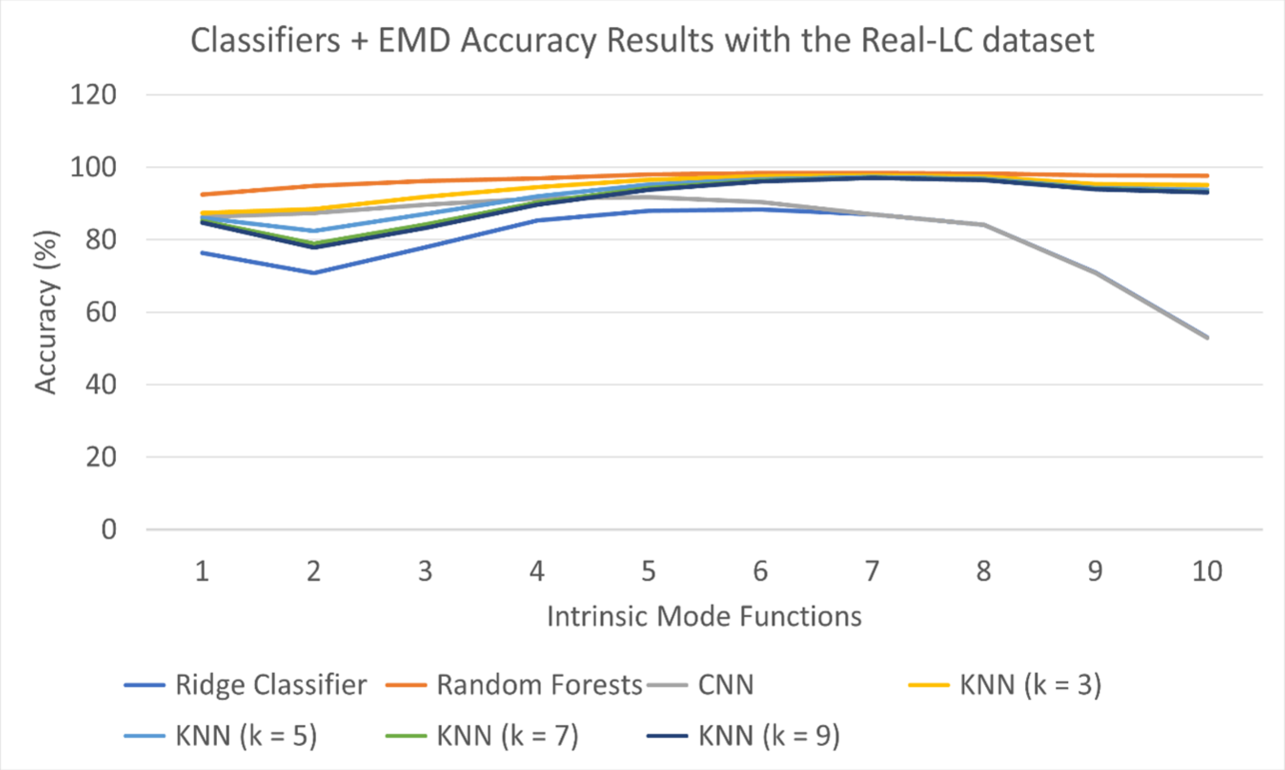


Figure C.2: Accuracy results obtained using light curves from the Real-LC dataset which were preprocessed with the Empirical Mode Decomposition (EMD) technique.

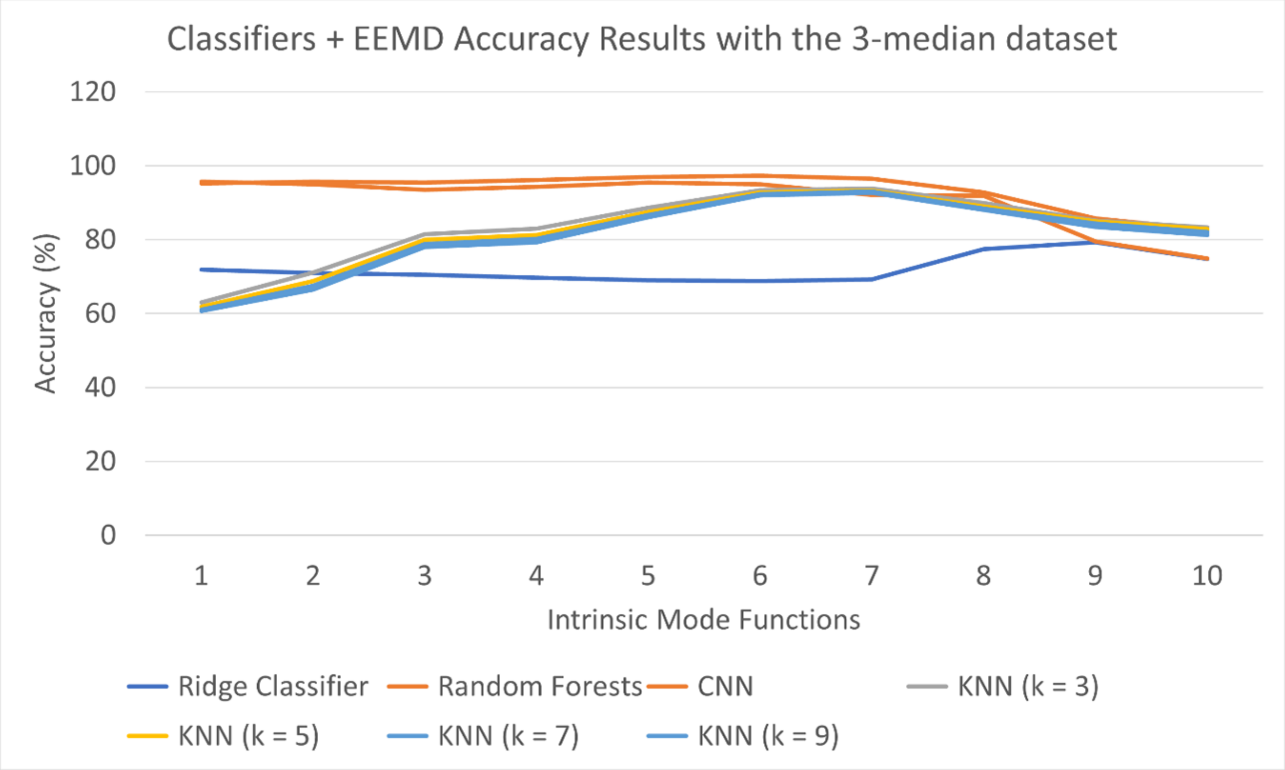


Figure C.3: Accuracy results obtained using light curves from the 3-median dataset which were preprocessed with the Ensemble Empirical Mode Decomposition (EEMD) technique.

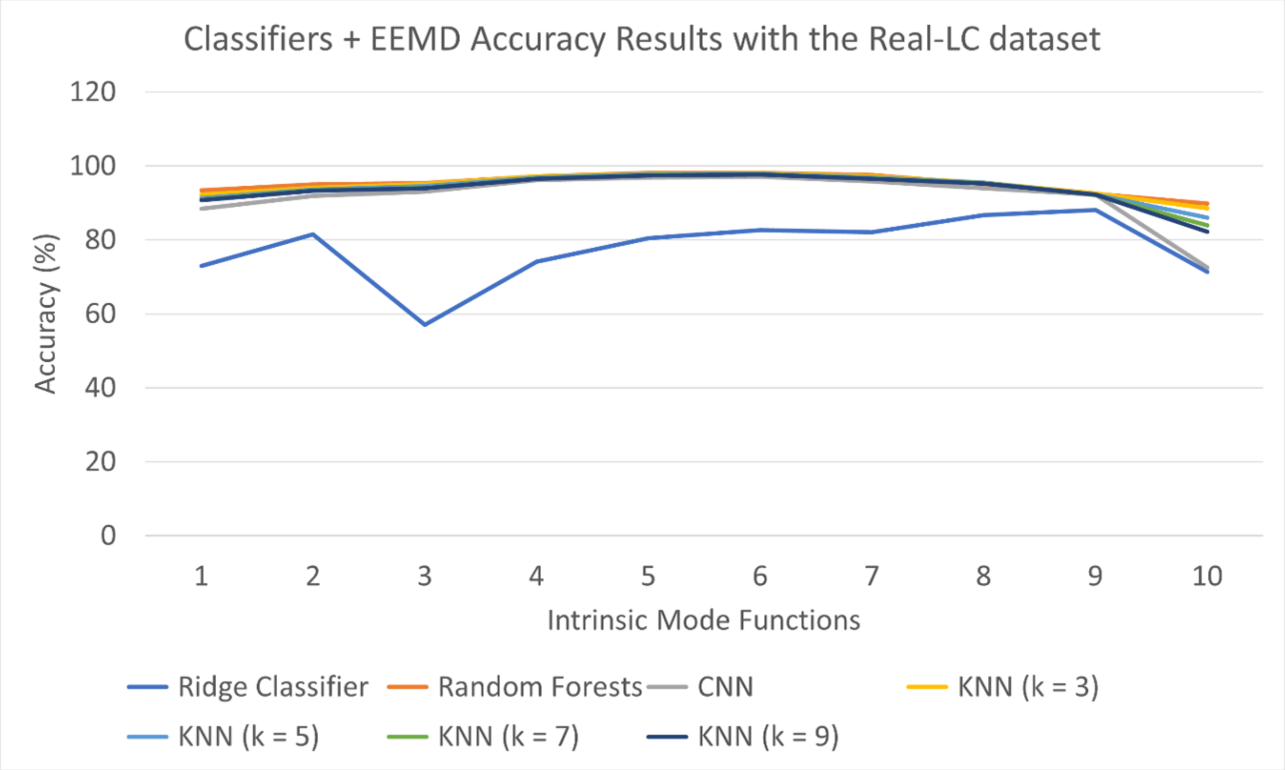


Figure C.4: Accuracy results obtained using light curves from the Real-LC dataset which were preprocessed with the Ensemble Empirical Mode Decomposition (EEMD) technique.

Appendix D

Partial Derivatives Used by the Proposed WABBLES Model

In this appendix, we present the calculation of the derivatives of the WABBLES parameters. Such derivatives are used by the model for backpropagation during the training process. These parameters are the mapping and enhancement biases ($\boldsymbol{\beta_{ei}}$ and $\boldsymbol{\beta_{hj}}$), the mapping and enhancement weights ($\boldsymbol{W_{ei}}$ and $\boldsymbol{W_{hj}}$), the wavelet translation parameter ($\boldsymbol{t_i}$), and the wavelet dilation parameter ($\boldsymbol{d_i}$). The derivatives are calculated from Eq. (4.8), which can be broken down to Eq. (D.1).

$$\sum_{i=1}^{n} \big(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}\big) + \sum_{j=1}^{m} \xi\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \tag{D.1}$$

D.1. Mapping Bias Parameter Derivative

The derivative of the mapping bias parameter $\frac{\delta y}{\delta \boldsymbol{\beta_{ei}}}$ (i.e. $\boldsymbol{\beta_{ei}}'$) is defined by:

$$\frac{\delta y}{\delta \boldsymbol{\beta_{ei}}} = \frac{\delta \sum_{i=1}^{n} \big(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}\big) + \sum_{j=1}^{m} \xi\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{\beta_{ei}}} \tag{D.2}$$

The term $\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i})$ does not depend on $\boldsymbol{\beta_{ei}}$, so it can be deleted.

$$\begin{aligned}\frac{\delta y}{\delta \boldsymbol{\beta_{ei}}} &= 1 + \frac{\delta \xi\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{\beta_{ei}}} \\ &= 1 + \xi'\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big) \cdot \frac{\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{\beta_{ei}}}\end{aligned} \tag{D.3}$$

where ξ' is the derivative of the enhancement function. Next, applying the chain rule on the last term, and removing $\boldsymbol{\beta_{hj}}$, which does not depend on $\boldsymbol{\beta_{ei}}$ gives:

$$\frac{\delta y}{\delta \boldsymbol{\beta_{ei}}} = 1 + \xi'\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}} \tag{D.4}$$

Equation (D.4) corresponds to the partial derivative of the output w.r.t. $\boldsymbol{\beta_{ei}}$.

D.2. Enhancement Bias Parameter Derivative

The derivative of the enhancement bias parameter $\frac{\delta y}{\delta \boldsymbol{\beta_{hj}}}$ (i.e. $\boldsymbol{\beta_{hj}}'$) is defined by:

$$\frac{\delta y}{\delta \boldsymbol{\beta_{hj}}} = \frac{\delta \sum_{i=1}^{n}\big(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}\big) + \sum_{j=1}^{m}\xi\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{\beta_{hj}}} \tag{D.5}$$

The first term does not depend on $\boldsymbol{\beta_{hj}}$, so it can be eliminated as follows,

$$\begin{aligned}\frac{\delta y}{\delta \boldsymbol{\beta_{hj}}} &= \frac{\delta \xi\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{\beta_{hj}}} \\ &= \xi'\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big) \cdot \frac{\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{\beta_{hj}}}\end{aligned} \tag{D.6}$$

Since $\boldsymbol{W_{hj}Z}$ does not depend on $\boldsymbol{\beta_{hj}}$, this can be converted into:

$$\frac{\delta y}{\delta \boldsymbol{\beta_{hj}}} = \xi'\big(\boldsymbol{W_{hj}Z} + \boldsymbol{\beta_{hj}}\big) \tag{D.7}$$

Equation (D.7) corresponds to the partial derivative of the output w.r.t. $\boldsymbol{\beta_{hj}}$.

D.3. Mapping Weight Parameter Derivative

The derivative of the mapping weight parameter $\frac{\delta y}{\delta \boldsymbol{W_{ei}}}$ (i.e. $\boldsymbol{W_{ei}}'$) is defined by:

$$\frac{\delta y}{\delta \boldsymbol{W_{ei}}} = \frac{\delta \sum_{i=1}^{n} \big(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}\big) + \sum_{j=1}^{m} \xi\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{W_{ei}}} \tag{D.8}$$

Since $\boldsymbol{\beta_{ei}}$ does not depend on $\boldsymbol{W_{ei}}$, it can be eliminated giving the next expression,

$$\frac{\delta y}{\delta \boldsymbol{W_{ei}}} = \frac{\delta\big(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i})\big) + \sum_{j=1}^{m} \xi\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{W_{ei}}} \tag{D.9}$$

Then, the following is obtained after applying the chain rule on the first term.

$$\begin{aligned}\frac{\delta y}{\delta \boldsymbol{W_{ei}}} &= \psi(\boldsymbol{c_i}) + \frac{\delta\xi\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{W_{ei}}} \\ &= \psi(\boldsymbol{c_i}) + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \frac{\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{W_{ei}}} \\ &= \psi(\boldsymbol{c_i}) + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \frac{\big(\boldsymbol{W_{hj}}(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}) + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{W_{ei}}}\end{aligned} \tag{D.10}$$

After applying the chain rule on the last term, and removing those terms that do not depend on $\boldsymbol{W_{ei}}$, the following is obtained:

$$\frac{\delta y}{\delta \boldsymbol{W_{ei}}} = \psi(\boldsymbol{c_i})\Big(1 + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}}\Big) \tag{D.11}$$

Equation (D.11) corresponds to the partial derivative of the output w.r.t. $\boldsymbol{W_{ei}}$.

D.4. Enhancement Weight Parameter Derivative

The derivative of the enhancement weight parameter $\frac{\delta y}{\delta \boldsymbol{W_{hj}}}$ (i.e. $\boldsymbol{W_{hj}}'$) is defined by:

$$\frac{\delta y}{\delta \boldsymbol{W_{hj}}} = \frac{\delta \sum_{i=1}^{n} \big(\boldsymbol{W_{ei}} \psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}\big) + \sum_{j=1}^{m} \xi\big(\boldsymbol{W_{hj}} \boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{W_{hj}}} \tag{D.12}$$

Since the first term does not depend on $\boldsymbol{W_{hj}}$ it can be eliminated as follows,

$$\begin{aligned} \frac{\delta y}{\delta \boldsymbol{W_{hj}}} &= \frac{\delta \xi\big(\boldsymbol{W_{hj}} \boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{W_{hj}}} \\ &= \xi'\big(\boldsymbol{W_{hj}} \boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \frac{\big(\boldsymbol{W_{hj}} \boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{W_{hj}}} \end{aligned} \tag{D.13}$$

The following is obtained after applying the chain rule on the last term, and removing $\boldsymbol{\beta_{hj}}$, which does not depend on $\boldsymbol{W_{hj}}$.

$$\frac{\delta y}{\delta \boldsymbol{W_{hj}}} = \xi'\big(\boldsymbol{W_{hj}} \boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{Z} \tag{D.14}$$

Equation (D.14) corresponds to the partial derivative of the output w.r.t. $\boldsymbol{W_{hj}}$.

D.5. Wavelet Translation Parameter Derivative

The derivative of the wavelet translation parameter $\frac{\delta y}{\delta \boldsymbol{t_i}}$ (i.e. $\boldsymbol{t_i}'$) is defined by:

$$\frac{\delta y}{\delta \boldsymbol{t_i}} = \frac{\delta \sum_{i=1}^{n} \big(\boldsymbol{W_{ei}} \psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}\big) + \sum_{j=1}^{m} \xi\big(\boldsymbol{W_{hj}} \boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{t_i}} \tag{D.15}$$

where the $\boldsymbol{t_i}$ parameter is used to calculate $\psi(\boldsymbol{c_i})$ (see Eq. (4.3)). The $\boldsymbol{\beta_{ei}}$ and $\boldsymbol{\beta_{hj}}$ parameters do not depend on $\boldsymbol{t_i}$, so they can be removed, leaving the following:

$$\begin{aligned}\frac{\delta y}{\delta \boldsymbol{t_i}} &= \frac{\delta\big(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i})\big) + \xi\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{t_i}} \\ \frac{\delta y}{\delta \boldsymbol{t_i}} &= \frac{\delta\big(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i})\big)}{\delta \boldsymbol{t_i}} + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \frac{\big(\boldsymbol{W_{hj}}\boldsymbol{Z}\big)}{\delta \boldsymbol{t_i}}\end{aligned} \tag{D.16}$$

The following is obtained after applying the chain rule to the first and last terms.

$$\frac{\delta y}{\delta \boldsymbol{t_i}} = \Big(\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}} \cdot \boldsymbol{W_{ei}}\Big) \cdot \Big(1 + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}}\Big) \tag{D.17}$$

Equation (D.17) corresponds to the partial derivative of the output w.r.t. $\boldsymbol{t_i}$.

D.6. Wavelet Dilation Parameter Derivative

The derivative of the $\boldsymbol{d_i}$ parameter is the same than the derivative of the $\boldsymbol{t_i}$ parameter. They differ when calculating the term $\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}}$, which depends on the wavelet that has been chosen. $\frac{\delta y}{\delta \boldsymbol{d_i}}$ (i.e. $\boldsymbol{d_i}'$) is defined by:

$$\frac{\delta y}{\delta \boldsymbol{d_i}} = \frac{\delta \sum_{i=1}^{n}\big(\boldsymbol{W_{ei}}\psi(\boldsymbol{c_i}) + \boldsymbol{\beta_{ei}}\big) + \sum_{j=1}^{m}\xi\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big)}{\delta \boldsymbol{d_i}} \tag{D.18}$$

where the $\boldsymbol{d_i}$ parameter is used to calculate $\psi(\boldsymbol{c_i})$ (see Eq. (4.3)). In the same way as with the $\boldsymbol{t_i}$ parameter, the $\boldsymbol{\beta_{ei}}$ and $\boldsymbol{\beta_{hj}}$ parameters can be eliminated because they do not depend on $\boldsymbol{d_i}$. Also, after applying the chain rule to the resulting first and last terms, the following is obtained:

$$\frac{\delta y}{\delta \boldsymbol{d_i}} = \Big(\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} \cdot \boldsymbol{W_{ei}}\Big) \cdot \Big(1 + \xi'\big(\boldsymbol{W_{hj}}\boldsymbol{Z} + \boldsymbol{\beta_{hj}}\big) \cdot \boldsymbol{W_{hj}}\Big) \tag{D.19}$$

Equation (D.19) corresponds to the partial derivative of the output w.r.t. $\boldsymbol{d_i}$.

Appendix E

Partial Derivatives of the Translation and Dilation Parameters Used by the Proposed WABBLES Model

In this appendix, the calculation of the derivatives of the wavelet parameters used by the proposed model (i.e. $\boldsymbol{t_i}'$ and $\boldsymbol{d_i}'$) are presented. The derivatives of the translation and dilation parameters are used to calculate the derivative of the wavelet function $\psi(\boldsymbol{c_i})'$. The calculation of such derivatives is different for each wavelet.

E.1. Derivative of the Radial Function Output

Recall from Eq. (4.3) that the output of the radial function $\boldsymbol{c_i} = ||\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})||$. This function remains the same regardless of the wavelet chosen. For this reason, the derivative of $\boldsymbol{c_i}$ will be the same for all wavelets. Nevertheless, it is necessary to calculate its derivative w.r.t. to the translation and the dilation parameters to use it for the calculation of the derivatives of such parameters for each wavelet function. The derivatives of the radial function output (i.e. $\boldsymbol{c_i}$) are presented next.

E.1.1. Derivative of the Radial Function Output w.r.t. the Wavelet Translation Parameter

The derivative of the radial function output w.r.t. the wavelet translation parameter $\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}}$ (i.e. $\boldsymbol{t_i}'$) is defined by:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}} = \frac{\delta\Big(\big(\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2\big)^{\frac{1}{2}}\Big)}{\delta \boldsymbol{t_i}} \tag{E.1}$$

The following is obtained after applying the chain rule.

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}} = \frac{1}{2}(\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2)^{-\frac{1}{2}} \frac{\delta\big(\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2\big)}{\delta \boldsymbol{t_i}} \tag{E.2}$$

The term $\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2\big)^{\frac{1}{2}}$ can be substituted by $\boldsymbol{c_i}$ as follows:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}} = \frac{1}{2}\boldsymbol{c_i}^{-1} \frac{\delta\big(\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2\big)}{\delta \boldsymbol{t_i}} \tag{E.3}$$

Eq. E.4 is obtained by using the chain rule:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}} = \frac{1}{2}\boldsymbol{c_i}^{-1}(\frac{\delta \boldsymbol{d_i}^2}{\delta \boldsymbol{t_i}}(\boldsymbol{X_k} - \boldsymbol{t_i})^2 + \boldsymbol{d_i}^2 \frac{\delta(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\delta \boldsymbol{t_i}}) \tag{E.4}$$

Since $\boldsymbol{d_i}$ does not depend on $\boldsymbol{t_i}$ the term that is multiplied by its derivative can be cancelled.

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}} = \frac{1}{2}\boldsymbol{c_i}^{-1}(\boldsymbol{d_i}^2 \frac{\delta(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\delta \boldsymbol{t_i}}) \tag{E.5}$$

The following is obtained after deriving the term $(\boldsymbol{X_k} - \boldsymbol{t_i})^2$.

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}} = \frac{1}{2}\boldsymbol{c_i}^{-1}\Big(\boldsymbol{d_i}^2\big(-2(\boldsymbol{X_k} - \boldsymbol{t_i})\big)\Big) \tag{E.6}$$

Finally, the following is obtained after clearing the equation:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}} = -\frac{\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})}{\boldsymbol{c_i}} \tag{E.7}$$

E.1.2. Derivative of the Radial Function Output w.r.t. the Wavelet Dilation Parameter

The derivative of the radial function output w.r.t. the wavelet dilation parameter $\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}}$ (i.e. $\boldsymbol{d_i}'$) is defined by:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}} = \frac{\delta(\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2)^{\frac{1}{2}}}{\delta \boldsymbol{d_i}} \tag{E.8}$$

The following is obtained after applying the chain rule.

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}} = \frac{1}{2}\Big((\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2)^{-\frac{1}{2}}\Big) \cdot \frac{\delta \boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\delta \boldsymbol{d_i}} \tag{E.9}$$

The term $\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2\big)^{\frac{1}{2}}$ can be substituted by $\boldsymbol{c_i}$ as follows:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}} = \frac{1}{2}(\boldsymbol{c_i}^{-1}) \cdot \frac{\delta \boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\delta \boldsymbol{d_i}} \tag{E.10}$$

Eqs. E.11 - E.13 are obtained by using the chain rule:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}} = \frac{1}{2\boldsymbol{c_i}}\Big(\frac{\delta \boldsymbol{d_i}^2}{\delta \boldsymbol{d_i}}(\boldsymbol{X_k} - \boldsymbol{t_i})^2 + \boldsymbol{d_i}^2\frac{\delta(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\delta \boldsymbol{d_i}}\Big) \tag{E.11}$$

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}} = \frac{1}{2\boldsymbol{c_i}}\Big((2\boldsymbol{d_i})(\boldsymbol{X_k} - \boldsymbol{t_i})^2 + \boldsymbol{d_i}^2(2(\boldsymbol{X_k} - \boldsymbol{t_i}))\frac{\delta(\boldsymbol{X_k} - \boldsymbol{t_i})}{\delta \boldsymbol{d_i}}\Big) \tag{E.12}$$

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}} = \frac{1}{2\boldsymbol{c_i}}\Big((2\boldsymbol{d_i})(\boldsymbol{X_k} - \boldsymbol{t_i})^2 + \boldsymbol{d_i}^2(2(\boldsymbol{X_k} - \boldsymbol{t_i}))\frac{\delta(\boldsymbol{X_k} - \boldsymbol{t_i})}{\delta \boldsymbol{d_i}}\Big) \tag{E.13}$$

The term $\boldsymbol{X_k} - \boldsymbol{t_i}$ does not depend on $\boldsymbol{d_i}$ so the equation turns into the following:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}} = \frac{1}{2\boldsymbol{c_i}}\Big((2\boldsymbol{d_i})(\boldsymbol{X_k} - \boldsymbol{t_i})^2\Big) \tag{E.14}$$

Finally, the following is obtained after clearing the equation:

$$\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}} = \frac{\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\boldsymbol{c_i}} \tag{E.15}$$

E.2. Derivatives of the Parameters of the Gaussian Derivative Wavelet

Recall from Eq. 4.12 that the Gaussian Derivative wavelet function is defined by the following equation $\psi(\boldsymbol{c_i}) = -\boldsymbol{c_i} e^{-\frac{1}{2}\boldsymbol{c_i}^2}$.

E.2.1. Gaussian Derivative Translation Parameter Derivative

The derivative of the translation parameter $\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}}$ for the Gaussian Derivative wavelet (i.e. $\boldsymbol{t_i}'$) is defined by:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}} = \frac{-\delta \boldsymbol{c_i} e^{-\frac{1}{2}\boldsymbol{c_i}^2}}{\delta \boldsymbol{t_i}} \tag{E.16}$$

Eqs. E.17 - E.20 are obtained by using the chain rule.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}} = -\Big(\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{t_i}} e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \boldsymbol{c_i}\frac{\delta e^{-\frac{1}{2}\boldsymbol{c_i}^2}}{\delta \boldsymbol{t_i}}\Big) \tag{E.17}$$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = -(\frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2}\frac{-\delta\frac{1}{2}\boldsymbol{c_i}^2}{\delta\boldsymbol{t_i}}) \tag{E.18}$$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = -(\frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot -\boldsymbol{c_i} \cdot \frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}}) \tag{E.19}$$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = -(\frac{-\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot (-\boldsymbol{c_i}) \cdot \frac{-\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})}{\boldsymbol{c_i}}) \tag{E.20}$$

The following is obtained after solving the sign multiplications.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = \frac{\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} - \boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2}(\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})) \tag{E.21}$$

Finally, the following is obtained after clearing the equation:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot \boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i}) \cdot \left(\frac{1}{\boldsymbol{c_i}} - \boldsymbol{c_i}\right) \tag{E.22}$$

E.2.2. Gaussian Derivative Dilation Parameter Derivative

The derivative of the dilation parameter $\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{d_i}}$ for the Gaussian Derivative wavelet (i.e. $\boldsymbol{d_i}'$) is defined by:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{d_i}} = \frac{-\delta\boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2}}{\delta\boldsymbol{d_i}} \tag{E.23}$$

This equation has already been solved in Eq. E.19, leaving the following:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{d_i}} = -(\frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{d_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot -\boldsymbol{c_i}\frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{d_i}}) \tag{E.24}$$

The derivative of $\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}}$ was already calculated in Eq. E.15, so Eq. E.24 can be rewritten as follows:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = -\Big(\frac{\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2}(-\boldsymbol{c_i})\frac{\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2}{\boldsymbol{c_i}}\Big) \tag{E.25}$$

The next is obtained after solving the divisions of the last term.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = -\Big(\frac{\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} - \boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2}\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2\Big) \tag{E.26}$$

The following is obtained after solving the sign multiplications.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = -\frac{\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2}\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2 \tag{E.27}$$

The next can be obtained by substituting the term $\boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2}$ by $-\psi(\boldsymbol{c_i})$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = -\frac{\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} - \psi(\boldsymbol{c_i})\cdot\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2 \tag{E.28}$$

Finally, the following is obtained after clearing the equation:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = -\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2\cdot\big(\frac{1}{\boldsymbol{c_i}}e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \psi(\boldsymbol{c_i})\big) \tag{E.29}$$

E.3. Derivatives of the Parameters of the Morlet Wavelet

Recall from Eq. 4.13 that the Morlet wavelet function is defined by the following equation $\psi(\boldsymbol{c_i}) = e^{-\frac{1}{2}\boldsymbol{c_i}^2}cos(5\boldsymbol{c_i})$.

E.3.1. Morlet Translation Parameter Derivative

The derivative of the translation parameter $\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}}$ for the Morlet wavelet (i.e. $\boldsymbol{t_i}'$) is defined by:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = \frac{\delta\big(e^{-\frac{1}{2}\boldsymbol{c_i}^2}\cdot cos(5\boldsymbol{c_i})\big)}{\delta\boldsymbol{t_i}} \tag{E.30}$$

Eqs. E.31 - E.33 are obtained by using the chain rule.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = \frac{\delta e^{-\frac{1}{2}\boldsymbol{c_i}^2}}{\delta\boldsymbol{t_i}}\cdot cos(5\boldsymbol{c_i}) + e^{-\frac{1}{2}\boldsymbol{c_i}^2}\cdot\frac{\delta\big(cos(5\boldsymbol{c_i})\big)}{\delta\boldsymbol{t_i}} \tag{E.31}$$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = (e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot\frac{-\delta\frac{1}{2}\boldsymbol{c_i}^2}{\delta\boldsymbol{t_i}}\cdot cos(5\boldsymbol{c_i}) + e^{-\frac{1}{2}\boldsymbol{c_i}^2}\cdot\big(-sen(5\boldsymbol{c_i})\big)\cdot\frac{\delta 5\boldsymbol{c_i}}{\delta\boldsymbol{t_i}} \tag{E.32}$$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = (e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot(-\boldsymbol{c_i})\cdot\frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}}\cdot cos(5\boldsymbol{c_i}) + e^{-\frac{1}{2}\boldsymbol{c_i}^2}\cdot\big(-sen(5\boldsymbol{c_i})\big)\cdot\frac{\delta 5\boldsymbol{c_i}}{\delta\boldsymbol{t_i}} \tag{E.33}$$

The derivative of $\frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}}$ was already calculated in Eq. E.7, so Eq. E.33 can be rewritten as:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = (-\boldsymbol{c_i}e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot cos(5\boldsymbol{c_i})\cdot(-\frac{\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})}{\boldsymbol{c_i}}) + e^{-\frac{1}{2}\boldsymbol{c_i}^2}\cdot\big(-sen(5\boldsymbol{c_i})\big)\cdot 5(-\frac{\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})}{\boldsymbol{c_i}}) \tag{E.34}$$

The next can be obtained after clearing the equation and applying the sign multiplications that correspond.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = (e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot cos(5\boldsymbol{c_i})\cdot\big(\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})\big) + e^{-\frac{1}{2}\boldsymbol{c_i}^2}\cdot\big(sen(5\boldsymbol{c_i})\big)\cdot 5(\frac{\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})}{\boldsymbol{c_i}}) \tag{E.35}$$

The term $(e^{-\frac{1}{2}\boldsymbol{c_i}^2}) \cdot cos(5\boldsymbol{c_i})$ can be substituted by $\psi(\boldsymbol{c_i})$ as follows:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}} = \psi(\boldsymbol{c_i}) \cdot \big(\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})\big) + e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot \big(sen(5\boldsymbol{c_i})\big) \cdot 5(\frac{\boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i})}{\boldsymbol{c_i}}) \tag{E.36}$$

Finally, the following is obtained after clearing the equation:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}} = \boldsymbol{d_i}^2(\boldsymbol{X_k} - \boldsymbol{t_i}) \cdot \Big(\psi(\boldsymbol{c_i}) + \frac{5}{\boldsymbol{c_i}} e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot sen(5\boldsymbol{c_i})\Big) \tag{E.37}$$

E.3.2. Morlet Dilation Parameter Derivative

The derivative of the dilation parameter $\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{d_i}}$ for the Morlet wavelet (i.e. $\boldsymbol{d_i}'$) is defined by:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = \frac{\delta\big(e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot cos(5\boldsymbol{c_i})\big)}{\delta \boldsymbol{d_i}} \tag{E.38}$$

This has already been solved in Eq. E.33, so Eq. E.38 can be rewritten as follows:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = (e^{-\frac{1}{2}\boldsymbol{c_i}^2}) \cdot (-\boldsymbol{c_i}) \cdot \frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{d_i}} \cdot cos(5\boldsymbol{c_i}) + e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot \big(-sen(5\boldsymbol{c_i})\big) \cdot \frac{\delta 5\boldsymbol{c_i}}{\delta\boldsymbol{d_i}} \tag{E.39}$$

The derivative of $\frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{d_i}}$ was already calculated in Eq. E.15.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = (e^{-\frac{1}{2}\boldsymbol{c_i}^2}) \cdot (-\boldsymbol{c_i}) \cdot \frac{\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\boldsymbol{c_i}} \cdot cos(5\boldsymbol{c_i}) + e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot \big(-sen(5\boldsymbol{c_i})\big) \cdot 5(\frac{\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\boldsymbol{c_i}})) \tag{E.40}$$

The next can be obtained after clearing the equation.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = -\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2) \cdot (e^{-\frac{1}{2}\boldsymbol{c_i}^2}) \cdot cos(5\boldsymbol{c_i}) + e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot \big(-sen(5\boldsymbol{c_i})\big) \cdot 5(\frac{\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2}{\boldsymbol{c_i}})) \tag{E.41}$$

The term $(e^{-\frac{1}{2}\boldsymbol{c_i}^2}) \cdot cos(5\boldsymbol{c_i})$ can be substituted by $\psi(\boldsymbol{c_i})$ as follows:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = -\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2 \cdot \psi(\boldsymbol{c_i}) + \frac{5}{\boldsymbol{c_i}} e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot \big(sen(5\boldsymbol{c_i})\big) \cdot \big(- \boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2\big) \tag{E.42}$$

Finally, the following is obtained after clearing the equation:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = -\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2 \cdot \big(\psi(\boldsymbol{c_i}) + \frac{5}{\boldsymbol{c_i}} e^{-\frac{1}{2}\boldsymbol{c_i}^2} \cdot sen(5\boldsymbol{c_i})\big) \tag{E.43}$$

E.4. Derivatives of the Parameters of the Mexican Hat Wavelet

Recall from Eq. 4.14 that the Mexican Hat wavelet function is defined by the following equation $\psi(\boldsymbol{c_i}) = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}(1 - \boldsymbol{c_i}^2)e^{-\frac{1}{2}\boldsymbol{c_i}^2}$.

E.4.1. Translation Parameter Derivative

The derivative of the translation parameter $\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}}$ for the Mexican Hat wavelet (i.e. $\boldsymbol{t_i}'$) is defined by:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}} = \frac{\delta\big(\frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}(1 - \boldsymbol{c_i}^2)e^{-\frac{1}{2}\boldsymbol{c_i}^2}\big)}{\delta \boldsymbol{t_i}} \tag{E.44}$$

Eqs. E.45 - E.47 are obtained by using the chain rule.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{t_i}} = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}\Big(\frac{\delta(1 - \boldsymbol{c_i}^2)}{\delta \boldsymbol{t_i}} \cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + (1 - \boldsymbol{c_i}^2) \cdot \frac{\delta(e^{-\frac{1}{2}\boldsymbol{c_i}^2})}{\delta \boldsymbol{t_i}}\Big) \tag{E.45}$$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}\Big(- 2\boldsymbol{c_i} \cdot \frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}} \cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + (1-\boldsymbol{c_i}^2) \cdot (e^{-\frac{1}{2}\boldsymbol{c_i}^2}) \cdot \frac{\delta(-\frac{1}{2}\boldsymbol{c_i}^2)}{\delta\boldsymbol{t_i}}\Big) \tag{E.46}$$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}\Big(- 2\boldsymbol{c_i} \cdot \frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}} \cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + (1-\boldsymbol{c_i}^2) \cdot (e^{-\frac{1}{2}\boldsymbol{c_i}^2}) \cdot (-\boldsymbol{c_i}) \cdot \frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}}\Big) \tag{E.47}$$

The derivative of $\frac{\delta\boldsymbol{c_i}}{\delta\boldsymbol{t_i}}$ was already calculated in Eq. E.7, so Eq. E.47 can be rewritten as:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}\Big(-2\boldsymbol{c_i}\cdot(-\frac{\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})}{\boldsymbol{c_i}})\cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2}+(1-\boldsymbol{c_i}^2)\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot(-\boldsymbol{c_i})\cdot(-\frac{\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i})}{\boldsymbol{c_i}})\Big) \tag{E.48}$$

Clearing the equation results in the following:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}\Big(2(\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i}))\cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + (1-\boldsymbol{c_i}^2)\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot(\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i}))\Big) \tag{E.49}$$

Relocating some terms gives the following expression:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = (\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i}))\left(\frac{4}{\pi^{\frac{1}{4}}\sqrt{3}}\cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + (\frac{2}{\pi^{\frac{1}{4}}\sqrt{3}})\cdot(1-\boldsymbol{c_i}^2)\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2})\right) \tag{E.50}$$

Finally, the following is obtained after clearing the equation and substituting $(\frac{2}{\pi^{\frac{1}{4}}\sqrt{3}})\cdot(1-\boldsymbol{c_i}^2)\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2})$ by $\psi(\boldsymbol{c_i})$.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{t_i}} = (\boldsymbol{d_i}^2(\boldsymbol{X_k}-\boldsymbol{t_i}))\left(\frac{4}{\pi^{\frac{1}{4}}\sqrt{3}}\cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + \psi(\boldsymbol{c_i})\right) \tag{E.51}$$

E.4.2. Dilation Parameter Derivative

The derivative of the dilation parameter $\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}}$ for the Mexican Hat wavelet (i.e. $\boldsymbol{d_i}'$) is defined by:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = \frac{\delta\big(\frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}(1-\boldsymbol{c_i}^2)e^{-\frac{1}{2}\boldsymbol{c_i}^2}\big)}{\delta \boldsymbol{d_i}} \tag{E.52}$$

This has already been solved in Eq. E.47, so Eq. E.52 can be rewritten as follows:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}\Big(-2\boldsymbol{c_i}\cdot\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}}\cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + (1-\boldsymbol{c_i}^2)\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot(-\boldsymbol{c_i})\cdot\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}}\Big) \tag{E.53}$$

The derivative of $\frac{\delta \boldsymbol{c_i}}{\delta \boldsymbol{d_i}}$ was already calculated in Eq. E.15, so Eq. E.53 can be expressed as follows:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}\Big(-2\boldsymbol{c_i}\cdot\big(\frac{\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2}{\boldsymbol{c_i}}\big)\cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + (1-\boldsymbol{c_i}^2)\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot(-\boldsymbol{c_i})\cdot\big(\frac{\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2}{\boldsymbol{c_i}}\big)\Big) \tag{E.54}$$

The equation can be cleared as it is done in the next two equations.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = \frac{2}{\pi^{\frac{1}{4}}\sqrt{3}}\Big(-2\big(\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2\big)\cdot e^{-\frac{1}{2}\boldsymbol{c_i}^2} + (1-\boldsymbol{c_i}^2)\cdot(-e^{-\frac{1}{2}\boldsymbol{c_i}^2})\cdot\big(\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2\big)\Big) \tag{E.55}$$

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta \boldsymbol{d_i}} = \big(\boldsymbol{d_i}(\boldsymbol{X_k}-\boldsymbol{t_i})^2\big)\Big(-\frac{4}{\pi^{\frac{1}{4}}\sqrt{3}}\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2}) - (\frac{2}{\pi^{\frac{1}{4}}\sqrt{3}})\cdot(1-\boldsymbol{c_i}^2)\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2})\Big) \tag{E.56}$$

The following equation is obtained after substituting the term $(\frac{2}{\pi^{\frac{1}{4}}\sqrt{3}})\cdot(1-\boldsymbol{c_i}^2)\cdot(e^{-\frac{1}{2}\boldsymbol{c_i}^2})$ by

$\psi(\boldsymbol{c_i})$.

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{d_i}} = \big(\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2\big)\Big(- \frac{4}{\pi^{\frac{1}{4}}\sqrt{3}} \cdot (e^{-\frac{1}{2}\boldsymbol{c_i}^2}) - \psi(\boldsymbol{c_i})\Big) \tag{E.57}$$

Finally, the following is obtained after clearing the equation:

$$\frac{\delta\psi(\boldsymbol{c_i})}{\delta\boldsymbol{d_i}} = -\big(\boldsymbol{d_i}(\boldsymbol{X_k} - \boldsymbol{t_i})^2\big)\Big(\frac{4}{\pi^{\frac{1}{4}}\sqrt{3}} \cdot (e^{-\frac{1}{2}\boldsymbol{c_i}^2}) + \psi(\boldsymbol{c_i})\Big) \tag{E.58}$$