FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automatic signal characterization for the Dendritic Cell Algorithm

Vitor Pereira

Mestrado em Engenharia e Ciência de Dados

Supervisor: Prof. Dr. Rui Pinto

October 26, 2024

Automatic signal characterization for the Dendritic Cell Algorithm

Vitor Pereira

Mestrado em Engenharia e Ciência de Dados

October 26, 2024

Abstract

Artificial Immune Systems (AIS) are algorithmic frameworks inspired by immunological principles, integrating elements from computer science, engineering, and mathematics. The Dendritic Cell Algorithm (DCA), a prominent AIS, is noted for its capability in anomaly detection across diverse fields, including network intrusion and cybersecurity. However, DCA's pre-processing phase, traditionally reliant on expert manual intervention, poses challenges such as subjective bias and inefficiency, especially with large-scale datasets. This dissertation addresses a critical literature gap by proposing automated feature reduction techniques to streamline DCA's pre-processing phase, thereby enhancing its efficiency and accuracy. The study introduces three methods: Kernel Principal Component Analysis (KPCA), Autoencoders (AE), and a hybrid approach (AEkPCA). These techniques aim to reduce feature dimensions in large datasets, facilitating more effective anomaly detection.

The research methodology involved extensive experimental and validation testing across datasets from varied domains, including biology, finance, and cybersecurity. The experimental phase used seven datasets, applying Z-score normalization, feature reduction, and DCA implementation. Validation tests featured scenarios with varying proportions of positive and negative labels. Metrics such as elapsed time, Mean Cost Attribute Value (MCAV), anomaly counter, and accuracy were used to evaluate performance.

Results demonstrated variable success across different techniques and datasets. While KPCA showed consistent performance, particularly with Gaussian kernels, AE and AEkPCA yielded mixed outcomes, highlighting both the potential and challenges of deep learning in feature reduction. Despite not fully achieving its objectives, this study contributes significantly to AIS literature by providing insights into the integration of advanced feature reduction techniques with DCA, and suggests pathways for future research to refine these approaches.

Keywords: Dendritic Cell Algorithm, Artificial Immune Systems, Feature Engineering, Anomaly Detection, Computational Biology, Signal Characterization

Resumo

Os Sistemas Imunitários Artificiais (AIS) são estruturas algorítmicas inspiradas em princípios imunológicos, integrando elementos da informática, da engenharia e da matemática. O Algoritmo das Células Dendríticas (DCA), um AIS proeminente, é conhecido pela sua capacidade de deteção de anomalias em diversos domínios, incluindo a intrusão na rede e a cibersegurança. No entanto, a fase de pré-processamento do DCA, tradicionalmente dependente da intervenção manual de especialistas, coloca desafios como a parcialidade subjectiva e a ineficiência, especialmente com conjuntos de dados de grande escala.

Esta dissertação aborda uma lacuna crítica na literatura, propondo técnicas automatizadas de redução de características para simplificar a fase de pré-processamento da DCA, melhorando assim a sua eficiência e precisão. O estudo apresenta três métodos: Análise de Componentes Principais de Kernel (KPCA), Autoencoders (AE) e uma abordagem híbrida (AEkPCA). Estas técnicas visam reduzir as dimensões das características em grandes conjuntos de dados, facilitando uma deteção de anomalias mais eficaz.

A metodologia de investigação envolveu testes experimentais e de validação extensivos em conjuntos de dados de domínios variados, incluindo biologia, finanças e cibersegurança. A fase experimental utilizou sete conjuntos de dados, aplicando a normalização do Z-score, a redução de características e a implementação do DCA. Os testes de validação incluíram cenários com proporções variáveis de rótulos positivos e negativos. Métricas como tempo decorrido, valor médio do atributo de custo (MCAV), contador de anomalias e precisão foram usadas para avaliar o desempenho.

Keywords: Algoritmo das Células Dendríticas, Sistemas Imunitários Artificiais, Engenharia de Dados, Detecção de Anomalias, Biologia Computacional, Caracterização de sinais

Acknowledgements

There were many adversities to overcome during these two years. Learning was continuous and the challenge was always there. Above all, after completing this stage, I feel prepared for what lies ahead.

To my supervisor, Prof Dr Rui Pinto, for all his incredible dedication, concern and commitment. I would also like to thank him for the knowledge and constant support he always gave me, culminating in this work. Your contribution has been fundamental and has greatly enhanced this work. Thank you very much!

To my family, my grandparents and godmother, for their support over all these years, from a young little boy until the man I am today. Without them, nothing like this would have been possible. I will always be grateful.

To all the colleagues who attended the master's in Engineering and Data Science with me, from whom I had the pleasure of learning intellectually and personally.

Special thanks go to the 'FEUP-Force Team' group for their constant mutual support, indispensable socialising and commitment. You have undoubtedly made this process easier and more enjoyable. I'm sure the future will see more of us.

Last but not least, to all my close friends who have shared this journey with me. Their contribution, support and encouragement were essential keys to achieving this goal. You know are also part of this.

Vitor Pereira

"It is not the strongest of the species that survives,
nor the most intelligent that survives.
It is the one that is the most adaptable to change."

Charles Darwin

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Problem Description . . . 3
- 1.3 Aim & Scope . . . 4
  - 1.3.1 Research Questions . . . 4
  - 1.3.2 Objectives . . . 4
- 1.4 Adressing the SDGs . . . 4
- 1.5 Thesis Structure . . . 5

2 Bio-inspired Computing **6**
- 2.1 Human Immune System . . . 7
- 2.2 Artificial Immune Systems . . . 9
  - 2.2.1 Negative Selection Algorithms . . . 9
  - 2.2.2 Clonal Selection Algorithm . . . 11
  - 2.2.3 Immune Network Algorithms . . . 13
- 2.3 The Danger Theory . . . 14
- 2.4 The Dendritic Cell Algorithm . . . 17
- 2.5 Signal Categorization . . . 17
- 2.6 Summary . . . 18

3 Feature Engineering **20**
- 3.1 Feature Selection . . . 21
  - 3.1.1 Filter Methods . . . 22
  - 3.1.2 Wrapper Methods . . . 22
  - 3.1.3 Embedded Methods . . . 22
- 3.2 Feature Reduction . . . 23
  - 3.2.1 Kernel PCA . . . 23
  - 3.2.2 Autoencoders . . . 24
  - 3.2.3 AEKPCA . . . 25
- 3.3 Summary . . . 26

4 Scientific Methodology **27**
- 4.1 Research Design . . . 27
  - 4.1.1 Research Approach . . . 27
  - 4.1.2 Research Strategy and Context . . . 28
- 4.2 Data Collection and Pre-Processing . . . 28
- 4.3 Data Analysis . . . 28

5 Proposed Solution **30**
5.1 Architecture . . . 30
5.1.1 Feature reduction . . . 30
5.2 Technological Framework . . . 34
5.3 Dendritic Cell Algorithm . . . 34

6 Empirical Analysis of the DCA Pre-Processing Phase **36**
6.1 Experimental Design and Dataset Overview . . . 36
6.2 Baseline Performance . . . 37
6.3 Kernel PCA . . . 38
6.3.1 Linear Kernel . . . 39
6.3.2 Gaussian Kernel . . . 40
6.3.3 Polynomial Kernel . . . 41
6.3.4 Sigmoid Kernel . . . 42
6.4 Autoencoder . . . 43
6.5 AEKPCA . . . 45
6.5.1 AEKPCA - Linear . . . 45
6.5.2 AEKPCA - Gaussian . . . 46
6.5.3 AEKPCA - Polynomial . . . 47
6.5.4 AEKPCA - Sigmoid . . . 48
6.6 Summary . . . 48

7 Validation and Performance Analysis of DCA **50**
7.1 Validation Test Design and Methodology . . . 50
7.2 Performance on Network Analysis Datasets . . . 52
7.2.1 OPC UA - Original Proportions . . . 52
7.2.2 OPC UA - Balanced Proportions . . . 53
7.2.3 OPC UA - Inverted Proportions . . . 54
7.3 Performance on Cybersecurity Datasets . . . 54
7.3.1 Port Scan - Original Proportions . . . 55
7.3.2 Port Scan - Balanced Proportions . . . 55
7.3.3 Port Scan - Inverted Proportions . . . 56
7.3.4 DDoS - Original Proportions . . . 57
7.3.5 DDoS - Balanced Proportions . . . 57
7.3.6 DDoS - Inverted Proportions . . . 58
7.4 Performance on Biological Datasets . . . 59
7.4.1 Diabetes - Original Proportions . . . 59
7.4.2 Diabetes - Balanced Proportions . . . 60
7.4.3 Diabetes - Inverted Proportions . . . 60
7.5 Performance on Financial Datasets . . . 61
7.5.1 Credit Card - Original Proportions . . . 61
7.5.2 Credit Card - Balanced Proportions . . . 62
7.5.3 Credit Card - Inverted Proportions . . . 63
7.6 Summary . . . 64

8 Discussion and Analysis of Results **65**
8.1 KPCA . . . 65
8.1.1 Empirical Tests Analysis . . . 65
8.1.2 Validation and Performance Tests . . . 68
8.2 AE . . . 71
8.2.1 Empirical Tests Analysis . . . 71
8.2.2 Validation and Performance Tests . . . 73
8.3 AEKPCA . . . 75
8.3.1 Empirical Tests Analysis . . . 75
8.3.2 Validation and Performance Tests . . . 78
8.4 Addressing the Research Questions . . . 80
8.4.1 Research Question 1 . . . 81
8.4.2 Research Question 2 . . . 81
8.4.3 Research Question 3 . . . 82

9 Conclusion **84**
9.1 Impact on literature . . . 84
9.2 Limitations . . . 85
9.3 Future Work . . . 85

A Dendritic Cell Algorithm **86**
A.1 K . . . 87
A.2 CSM . . . 87

References **88**

List of Figures

2.1 An example of the main immunological elements that interact between the immune and the adaptive systems [1]. 8
2.2 Schematic representation of the selection of mature T-cells from thymocytes [2]. . 10
2.3 The theory of antibody production through clonal selection. There is just one kind of antibody receptor produced on the surface of each B cell. One B lymphocyte is recognized by an antigen from a broad repertoire. This causes the B cell to divide quickly and differentiate into a "plasma cell," which produces and secretes antibodies that are unique to the original antigen [3]. 12
2.4 Network activation occurs when an antibody recognises an antigen via a cellular receptor, and network suppression occurs when an antibody recognises an idiotope. [4]. 13
2.5 Development and role of DCs in the immune response: From peripheral tissue surveillance to T-cell activation in lymph nodes [5]. 15

3.1 Showing various feature extraction and feature selection methods for dimensionality reduction. 21
3.2 General architecture of an AE [6]. 24
3.3 Description of the method [7]. 25

4.1 DCA pipeline. 29

5.1 KPCA pipeline. 31
5.2 AE network architecture. 32
5.3 AEKPCA network architecture. 33
5.4 DCA architecture. 35

8.1 Elapsed time by dataset and feature reduction technique. 65
8.2 MCAV heatmap by dataset and feature reduction technique. 66
8.3 Anomaly Counter distribution by feature reduction technique. 67
8.4 Accuracy comparison by dataset and feature reduction technique. 67
8.5 Average elapsed time by dataset and feature reduction technique across the three scenarios. 68
8.6 Average MCAV by dataset and feature reduction technique across the three scenarios. 69
8.7 Average Anomaly Counter distribution by feature reduction technique. 70
8.8 Average accuracy comparison by dataset and feature reduction technique. 70
8.9 Elapsed time by dataset. 71
8.10 MCAV by dataset. 72
8.11 Anomaly counter distribution 72
8.12 Accuracy comparison by dataset. 73

8.13 Average elapsed time by dataset. 73
8.14 Average MCAV by dataset. 74
8.15 Average anomaly counter distribution 74
8.16 Average accuracy comparison by dataset. 75
8.17 Elapsed time by dataset and feature reduction technique. 76
8.18 MCAV heatmap by dataset and feature reduction technique. 76
8.19 Anomaly counter distribution by feature reduction technique. 77
8.20 Accuracy comparison by dataset and feature reduction technique. 77
8.21 Average elapsed time by dataset and feature reduction technique across the three scenarios. 78
8.22 Average MCAV by dataset and feature reduction technique across the three scenarios. 79
8.23 Average anomaly counter distribution by feature reduction technique. 79
8.24 Average accuracy comparison by dataset and feature reduction technique. 80

List of Tables

2.1 Main works on investigating pre-processing phase and signal categorization. . . . 18

6.1 Datasets' description. . . . 36
6.2 Results of the baseline test. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 38
6.3 Results of the Linear kernel-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 39
6.4 Results of the Gaussian kernel-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 41
6.5 Results of the Polynomial kernel-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 42
6.6 Results of the Sigmoid kernel-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 43
6.7 Results of the AE-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 45
6.8 Results of the AEKPCA-Linear-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 46
6.9 Results of the AEKPCA-Gaussian-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 47
6.10 Results of the AEKPCA-Polynomial-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 47
6.11 Results of the AEKPCA-Sigmoid-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter. . . . 48

7.1 Validation datasets. . . . 51
7.2 OPC UA first validation scenario. . . . 52
7.3 OPC UA second validation scenario. . . . 53
7.4 OPC UA third validation scenario. . . . 54
7.5 Port Scan first validation scenario. . . . 55
7.6 Port Scan second validation scenario. . . . 56
7.7 Port Scan third validation scenario. . . . 56
7.8 DDOS first validation scenario. . . . 57
7.9 DDOS second validation scenario. . . . 58
7.10 DDOS third validation scenario. . . . 59
7.11 Diabetes first validation scenario. . . . 59
7.12 Diabetes second validation scenario. . . . 60
7.13 Diabetes third validation scenario. . . . 61
7.14 Credit card first validation scenario. . . . 62

7.15 Credit card second validation scenario. 62
7.16 Credit card third validation scenario. 63

List of Algorithms

1 Generic Negative Selection Algorithm 11
2 Generic Clonal Selection Algorithm 13
3 Generic Immune Network Algorithm. 14
4 Generic Dendritic Cell Algorithm. 16

Abbreviations

AE	Autoencoder
AEkNN	Autoencoder with KNN
AEKPCA	Autoencoder with Kernel Principal Components Analysis
AIN	Artificial Immune Networks
AIS	Artificial Immune Systems
ANN	Artificial Neural Networks
APCs	Antigen Presenting Cells
CLONALG	The CLOnal ALGorithm
CSA	Clonal Selection Algorithm
DCA	Dendritic Cell Algorithm
DCs	Dendritic Cells
dDCA	deterministic Dendritic Cell Algorithm
DDoS	Distributed Denial-of-Service
pDCA	prototy Dendritic Cell Algorithm
rDCA	robotic Dendritic Cell Algorithm
FE	Feature Engineering
GA	Genetic Algorithms
HIS	Human Immune Systems
IQR	Interquartile Range
KPCA	Kernel Principal Components Analysis
LLE	Locally Linear Embedding
MCAV	Mature Context Antigen Value
ML	Machine Learning
MSE	Mean Squared Error
MVU	Maximum Variance Unfolding
NSA	Negative Selection Algorithms
PAMP	Pathogen-Associated Molecular Pattern
PCA	Principal Components Analysis
PS	Particle Swarm
SGDs	Sustainable Development Goals
SVM	Support Vector Machine
RBF	Radial Basis Function
RST	Rough Set Theory
aiNet	The Artificial Immune Network algorithm

Chapter 1

Introduction

Humans have long tended to replicate their behaviour in machines. Among many other solutions, Artificial Immune Systems (AIS) implements a technique inspired by the Human Immune System (HIS), which is an extraordinary natural defence mechanism that learns about invading pathogens [8]. Thus, from a computational perspective, HIS has several seductive properties that have aroused the interest of different areas [9]. Resources such as "pattern recognition", "noise tolerance", or "anomaly detection" are as powerful as they are rare, which increases HIS value [10]. Consequently, various approaches to AIS have emerged, bridging multiple areas such as immunology, engineering, computer science, and mathematics.

HIS is one of nature's most complex and robust systems. It can even be compared to the nervous system [11]. The main purpose of the Immune System is to protect the body from harmful entities, mostly invaders, but in some cases, damage can be triggered by the body itself [12]. The range of threats is vast, meaning several actions may be needed to repel them. However, most of these actions are still not fully understood by the scientific community [13]. On the other hand, the complexity of the processes represents a source for developing new approaches to bio-inspired computing.

From a computational point of view, several advantages make this algorithm family interesting to the computer science community. Due to a comprehensive collection of properties mapped from the natural system, it is thought to have an advantage over other conventional approaches in some situations. One of the most appreciated advantages is that several of them come across as lightweight algorithms, especially in classification tasks, when compared to Machine Learning (ML) algorithms [14]. Also, some other applications showed that AIS are advantageous in detection performance over conventional techniques [15].

Dendritic Cell Algorithm (DCA) [16] was designed to replicate the functions of Dendritic Cells (DCs) in the innate immune system in classification problems. This algorithm is an innovative approach and has some interesting features. It can perform data fusion, removing noise due to its filtering property, temporal correlation of anomalies with their respective causes and multi-scale aggregation [17]. To take advantage of these useful properties, there are several applications that are worth mentioning since they were successful, such as network intrusion [18], robotic [14],

port scan detection [19], botnets detection [20], wireless sensor networks [21], operating systems [22] or cloud model [23].

Despite the success of these applications, there are still many unresolved issues. As this algorithm is relatively new, most of the existing work is not yet sufficient to have a complete understanding of how it works and to indicate all its strengths and weaknesses. Unlike conventional ML algorithms, which can often generalise well across different domains with little human assistance, DCA requires much input from domain experts. Because of its dependence on human expertise, the algorithm is less scalable and generic when promoting its use as a learner. It is, therefore, necessary to propose and test new approaches to dealing with the algorithm and integrating it into different pipelines.

1.1 Motivation

In the human body, the HIS acts as a protective agent whose behaviour is influenced by the context and the surrounding environment. With this in mind, an immunological response is triggered. Either the innate immune system, which acts as a first barrier, or the adaptive immune system, which gives a more specific response, will be activated based on the alert type [24].

At the same time, humans have been inspired by these biological organisms to devise new innovative computational approaches. Artificial neural networks or genetic algorithms are good examples of successful applications of the biological metaphor to computer problems [25]. AIS has emerged as one of these approaches that are trying to mimic a certain human behaviour; in this particular case, it exploits the mechanisms of HIS. Within these mechanisms, two different categories are appropriate to classify this family of algorithms.

In the first category, called "first generation", the negative and the clonal selection algorithms are examples. Both are derived from the "self/non-self" theory of immunity [25]. As the theory, both algorithms also consider any non-self substance to be harmful [12, 25]. Stibor *et al.* have found that these algorithms are vulnerable to problems with scalability, and their outcome generated a high rate of false positives [26]. This became an issue since the algorithm's performance was far from what he was capable of. To overcome this, more up-to-date approaches were designed in collaboration with immunologists. They believe that the results are promising in what concerns replicating the robustness that humans can experience [27].

Based on the "Danger Theory", the DCA was created to reproduce the functions of DCs, inserted in the second category and representing the "second generation" [28, 16]. Unlike the previous theory, this one considers the surrounding environment and context rather than just a specific compound. Using a collection of input signals, the DCA algorithm fuses multi-sensor data and correlates the results with "antigens" [20]. The DCA is very effective in many applications with promising results due to its characteristics, including bots detection [20], network intrusion detection [18], temporal correlation on wireless data [21], or categorization of signals based on basic statistical analysis [29]. There are many other areas where this algorithm can be implemented,

even for more general scientific applications. Due to its ability to analyse time-dependent data in real-time, it may be useful in predicting earthquakes [30].

Regardless of all these successful applications, there are some weaknesses that are the root cause of some limitations. In order to solve some of these issues, this work has found a proposal. Out of the associated limitations, data pre-processing phase arises as one of the most impactful on the algorithm's behavior, it's either performed based on users expertise (manually) or with some feature extraction method (automatically) [31]. If the data is pre-processed manually, algorithm independence is not possible. If the pre-processing is carried out using an automatic technique, this will destroy the meaning behind the features which were present in the original data [31]. This is indeed a problem because it is essential to categorize each signal, in order to know the source where it comes from. An appropriate mapping between the resulting features and the respective signal categories should be found to meet this criterion. To do this, either experts in a given field are asked to map the features or a suitable classification is devised to be applied to the selected features, as in [32].

Thus, within the aforementioned gaps, this dissertation finds motivation to propose and test different approaches to solving the problem. Initially, feature reduction will be performed on the original dataset in order to feed the algorithm without losing meaningful information, if possible. After that, the result will also be mapped to the original set of features.

1.2 Problem Description

The problem that this dissertation aims to solve can be described through different key aspects. As in many other domains, this algorithm also depends on human expertise to carry out the pre-processing phase and select the features that will be used in the classification phase. This manual intervention introduces subjectivity and bias into the process since it is carried out by a human who may or may not be an expert in the field. Thus, it is clear that each feature selected will play a crucial role in characterising the signal, making it possible to map the features and the respective signals. Therefore, choosing the appropriate feature selection technique has become even more important.

Another crucial aspect is the need for adaptability to the problem domain, which means that different domains may require a distinct set of features to characterize the signals. Without this adaptability, the algorithm may have difficulty generalising and correctly classifying whether or not there is an anomaly. Based on that, automated mechanisms for adapting to the problem domain are essential. It is also important to mention that this dissertation is conducted to enhance DCA's performance in detecting anomalies, which means that the feature reduction process will play a key role in it.

Taking into account what was said above, this will probably lead to an increase in the algorithm's performance, which has become a problem since it shows some variability in performance, particularly in terms of false positives.

1.3 Aim & Scope

The aim of this dissertation is to investigate, test and propose a technique that better adapts to the most well-known limitation of DCA, which is the pre-processing stage. Furthermore, to map each pre-processed signal to its respective triggered response, establishing a cause-and-effect relationship. After that, new data will be generated and ingested by the DCA to classify the presence or absence of an anomaly. Different metrics will be considered to evaluate the process.

1.3.1 Research Questions

According to the aim of this dissertation, three research questions arose during the development of this work. The research questions are listed as follows:

- **RQ1:** Is the DCA performance dependent on the effectiveness of data pre-processing?
- **RQ2:** Can the DCA be applicable to a larger set of problems when integrated with feature reduction techniques, such as KPCA, AE and AEKPCA?
- **RQ3:** How can KPCA, AE and AEKPCA enhance the data pre-processing phase of the DCA to make the system automated and adaptable to a given problem domain?

1.3.2 Objectives

In order to answer these research questions and consequently meet the goal of this dissertation, several objectives were set with regard to the study of the AIS in general and the DCA in particular:

- Integrating feature reduction techniques to automate signal characterization in the initial phase of DCA;
- Evaluate and compare the performance of the various existing versions of the DCA with the versions proposed in this work;
- Evaluate which feature reduction technique produces the best DCA performance results;
- Evaluate the computational efficiency of feature reduction techniques and DCA variants;
- Explore the scalability of feature reduction techniques and DCA variants for large-scale datasets.

1.4 Adressing the SDGs

Based on the definitions provided by the United Nations, this dissertation aims at three different Sustainable Development Goals (SDGs). The development and validation of a solution to address the pre-processing phase of the DCA aligns with SDG 3: Good Health and Well-being, SDG 9: Industry, Innovation and Infrastructure, and SDG 16: Peace, Justice and Strong Institutions.

Regarding SDG 3, which relates to ensuring healthy lives and promoting well-being for every living creature, the application of this study to health-related datasets, such as heart failure, stroke or diabetes, contributes directly to improving patient monitoring systems and healthcare technology.

Moreover, this dissertation also supports SDG 9, which highlights the development of robust infrastructures, the promotion of equitable and sustainable industrialisation and encouraging innovation. The sophisticated methods created in this dissertation improve the security and reliability of critical infrastructure systems by strengthening their ability to identify anomalies.

SDG 16, which focuses on promoting inclusive and peaceful communities, is also supported by this initiative in mitigating and detecting cyber security threats and contributing to the security and resilience of institutions.

This alignment highlights the relevance of this dissertation to global issues. It provides a framework for evaluating and improving its contributions to sustainable development, ultimately maximising its beneficial effects on society.

1.5 Thesis Structure

This dissertation is structured into 9 different chapters. Chapter 1 introduces the study and the context that motivates this investigation, as well as the research questions and the SDGs. Chapter 2 presents the biological theoretical concepts and context that inspired this type of algorithm. It also provides a literature overview. Chapter 3 provides an analysis of the existent feature reduction techniques. It also states a more detailed explained of each technique that is employed during this study.

Moreover, Chapter 4 describes the scientific methodology that this dissertation intends to follow to address the research questions. Chapter 5 states the proposed solution to the referred literature gap. Chapter 6 delivers an extensive set of tests to evaluate DCA's pre-processing phase. Chapter 7 is another group of tests designed to validate DCA's performance. Chapter 8 shows an extensive and detailed results analysis.

Finally, Chapter 9 concludes this document by summarising the key aspects found during the study, reflecting the most impactful limitations, and emphasising the importance of further investigations concerning this domain.

Chapter 2

Bio-inspired Computing

Biologically inspired computing has emerged as one of the most active interdisciplinary research areas since the early 1990s. Typically, biology, computer science, engineering, and mathematics are involved. Scientists working in this field are inspired by nature to build computer systems that mimic the desirable characteristics of biological systems. Because biologically inspired computing has had such a profound impact on numerous machine learning approaches, it is closely linked to the study of artificial intelligence.

Among all the implementations made over the years, some have become successful approaches. Inspired by the evolutionary process of living organisms, the Genetic Algorithms (GA) were designed. These GA are useful both as search methods for solving problems and for modelling evolutionary systems [33]. Binary strings are stored in a computer's memory and are modified over time, imitating the evolution of populations under natural selection.

Another successful approach was the cellular automata, which are mathematical idealizations of physical systems in which space and time are discrete and physical quantities take on a finite set of discrete values. The intention is to mimic cells evolving through a number of discrete steps based on the state of their neighbouring cells [34].

Taking inspiration from the sophisticated functionality of the human brain, where hundreds of billions of interconnected neurons process information in parallel, Artificial Neural Networks (ANN) are built out of densely interconnected sets of simple units [35]. It consists of an input layer of neurons, one or even more hidden layers, and to produce an output, a final layer of output neurons [36].

More recently, at the end of the 20th century, Particle Swarm (PS) was conceived based on the collective behaviour of groups of organisms such as fish schooling, insects swarming or birds flocking. It is used for problems where the function to be optimized is discontinuous and non-differentiable with too many non-linearly related parameters [37]. A swarm is an enormous number of homogeneous, decentralized agents that interact locally with one another and their surroundings in an effort to find a global solution, which is frequently optimal [38].

Along with these, a new research area called AIS has emerged. The focus of this area is drawing inspiration from theoretical immunology, principles and models to solve real-world problems

with computer systems [10]. The HIS evolved to defend against a variety of invasive microorganisms; similarly, artificial intelligence systems are created to offer similar defensive capabilities within a computing environment. Initially, a significant number of immune-inspired algorithm applications were focused on computer security, particularly intrusion detection [39]. Because AIS has a comprehensive collection of properties mapped from the natural system, it is thought to have an advantage over other conventional approaches in some situations.

2.1 Human Immune System

The HIS is frequently called an organization of cells and molecules with specialized roles in defending against harmful micro-organisms, such as bacteria and viruses, named pathogens [40]. To this end, there are two different types of systems that respond to invading microbes, namely the innate immune system and the adaptive immune system, each with a different set of components and characteristics [41]. The innate immune system evolves and is passed down from generation to generation, whereas the adaptive immune system develops during the course of an individual's lifetime. Furthermore, the innate immune system frequently reacts to invasive pathogens immediately, sometimes even hours after the first contact. On the other hand, it can take days or even months for the adaptive immune system to provide a more focused response [42].

The innate immune system uses phagocytic cells (neutrophils, monocytes and macrophages), cells that release inflammatory mediators (basophils, mast cells and eosinophils) and natural killer cells [43]. Acute-phase proteins, complement, and cytokines like interferons are among the molecular elements of innate reactions. When antigen binds to these cells' surface receptors, antigen-specific B and T cells proliferate, resulting in acquired responses. Antigen-presenting cells are specialized cells that help lymphocytes respond to antigens by displaying the antigen to them. Immunoglobulins, which are antigen-specific antibodies that eradicate extracellular microbes, are secreted by B cells. T cells stimulate macrophages and destroy virally infected cells in addition to assisting B cells in the production of antibodies and eliminating intracellular infections. Usually, innate and acquired defences combine to eliminate infections [40, 43].

The spleen, lymph nodes, and mucosa-associated lymphoid tissue all produce adaptive immune responses. The secondary lymphoid tissues are what we call these. Different B- and T-cell compartments of lymphoid tissue in the spleen and lymph nodes are where antigen-activated lymphocytes occur. The secondary follicle housing the germinal centre, where B-cell responses occur within a network of follicular DCs, is a remarkable morphologic characteristic of the B-cell region. Mucosal surfaces are protected by the lymphoid tissues connected to the mucosa, such as the tonsils, adenoids, and Peyer's patches. There are diffuse collections of lymphoid cells in the gut wall's lamina propria as well as the lung [40, 44].

Because the innate and adaptive immune systems must work together to protect the body against a variety of invasive infections, they are inextricably linked and cannot exist apart. The group of Antigen-Presenting Cells (APCs) known as DCs is responsible for maintaining the connection between these two immune systems [28]. DCs need to engage with T-cells in order to

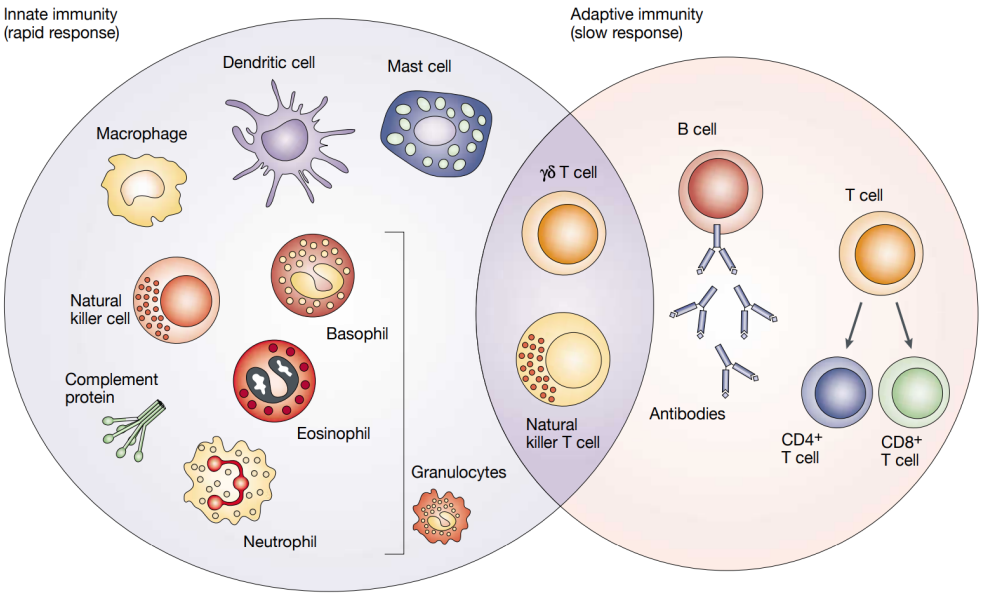

Figure 2.1: An example of the main immunological elements that interact between the immune and the adaptive systems [1].

trigger immune responses against particular infections. B-cells could potentially be involved, depending on the sort of antigens. Figure 2.1 shows the main immunological elements that are engaged in the interaction between the innate and adaptive immune systems.

The computationally viable characteristics of HIS motivate the creation of artificial systems with comparable capabilities. The immune system's ability to make decisions is directly linked to this type of computing. Decision-making is accomplished through a mechanism called "immune learning", which involves long-term evolutionary learning at the species level and short-term reinforcement learning at the individual level [45]. Long-term learning is defined as the process of developing a collection of detection mechanisms used as pattern recognition. An individual inherits this from their parents, who in turn inherit from their parents, and so on, from generation to generation. The time span to achieve this process could be extensive, as the innate immune system shows. In contrast, the innate immune system develops short-term learning through exposure to various proteins during an individual's life.

To respond quickly and effectively to both known and unknown threats, negative selection and clonal selection are the mechanisms for short-term learning [46]. Negative selection generates a population of T-cells that react to non-self substances, which frequently are considered dangerous, as stated in the self/non-self theory [12]. Thus, from the exposed cells, those that react to their own substances are removed, leaving those that would only respond to pathogens. Moreover, the purpose of clonal selection is to generate B-cells, which can effectively produce antibodies to recognise antigens. It is suggested that the immunological memory, which is in charge of boosting the potency and efficacy of second-encountered immune responses, "remembers" the knowledge of pathogens [47]. As in [48], the state change of the HIS displays computational characteristics

akin to those of a Turing machine. Therefore, the immune system can be thought of as a computing device that converts body-state input into immune-system data and then provides feedback in real-time to alter the body's condition and reestablish its health [49].

2.2 Artificial Immune Systems

As mentioned earlier, AIS are a family of bio-inspired computational approaches that emerged in the 1990s, merging various domains, including immunology, computer science and engineering. Immunology is typically modelled mathematically or computationally; then those models are abstracted into algorithms, which are then designed and implemented in the context of computer science and engineering to create immune-inspired algorithms [50]. There are three different types of AIS researchers, proposed by Cohen, namely literal school, metaphorical school and modellers [51]. Regarding the first, they are people who actually mimic the exact behaviours of the immune system. Metaphorical school finds inspiration in the immune system to build computer systems with similar functionalities. Lastly, modellers use computational or mathematical modelling to understand the HIS. In parallel, there are four major immunological topics that have influenced AIS architectures, specifically Negative Selection Algorithms (NSA) [52], Clonal Selection Algorithms [53], Artificial Immune Networks [54, 55]. Danger Theory [56, 57]. AIS techniques have previously proven to be effective in solving computer science and engineering real-world problems. Their nature show unique qualities, which Aldhaheri *et al.* [58] divided as the following:

1. **Adaptive**: can acquire new skills and adaptable habits over time.
2. **Robust**: can handle processing in situations with ambiguous and imprecise data.
3. **Resiliency**: using a bottom-up methodology and an agent-based paradigm, the immune system is designed to operate in dynamic, heterogeneous, and dispersed contexts.
4. **Self-tolerance**: can inhibit the immune system's reaction to a specific self-antigen, i.e., benign pathogens.
5. **Lightweight**: has a low computational complexity of processing.
6. **Distributed**: can operate without centralized management or control by processing in a distributed fashion.

Since the main objective of this dissertation is to address the pre-processing phase and map the remaining features with the original set of features, these characteristics may be useful to accomplish what is intended.

2.2.1 Negative Selection Algorithms

Initially proposed by Forest *et al.* [52], the self/non-self immunological theory is the base of the Negative Selection Algorithm (NSA), which implies the T-cells maturation, known as "negative selection", which occurs in the thymus [2].

Firstly, a population of T cells with a unique binding affinity to a specific type of antigen is created. After this, in order to select only fully developed T cells, the T cells produced are then exposed to self-proteins. Since a fully developed T cell doesn't stick to the self-proteins, it can be released from the thymus and perform immunological functions within the body. On the other hand, T cells that react to self-proteins are eliminated [12]. Figure 2.2 helps to visualize the selection process.

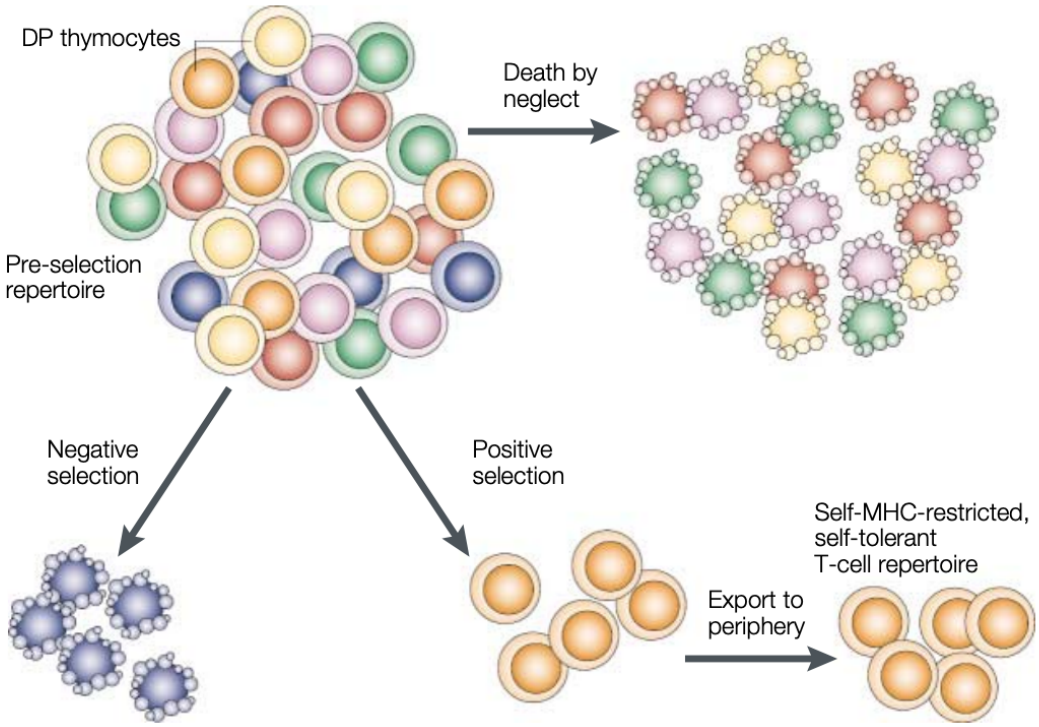

Figure 2.2: Schematic representation of the selection of mature T-cells from thymocytes [2].

Although the algorithm was proposed in 1994 and some new approaches have been tested since then, the algorithm still consists of three phases, namely the representation of self-information, the creation of detectors and the classification of anomalies [59]. More recently, alternative NSA architectures have been presented, which rely on various techniques for matching criteria, detector creation, and data encoding [60]. With regard to network security, Hofmeyer and Forrest proposed a new architecture based on the negative selection that has shown great potential [61]. Other methods were tested to improve the algorithm's properties, in this case Gonzales and Dasrupta [62] or Stibor *et al.* [26, 63] followed by Stibor's thesis [64]. The former increased the algorithm's computational complexity. The latter emphasised the enormous challenge of creating detectors without considering the type of component representations.

As Algorithm 1 shows, a population of "detectors", which resemble T-cells, is initially produced by the process. Next, a random set of generators is created, which can be expressed using various data types (binary, character strings, among others). A metric is used to compare the data points with the cells themselves to determine the affinity of each element. This process is negative

self-selection, which means that the detector is immediately destroyed or not, based on its reaction with self-cells (T-cell maturation).

Algorithm 1 Generic Negative Selection Algorithm

Require: S_{seen} = set of known self elements
Ensure: D = set of generated detectors
1: begin
2: **repeat**
3: Randomly generate a set P of potential detectors;
4: Determine the affinity of each element of P with each element of S_{seen};
5: **if** one element in S_{seen} recognizes a detector in P **then**
6: Remove the detector from P;
7: **else**
8: Add the detector to D;
9: **end if**
10: **until** stopping criteria has been reached;
11: end

2.2.2 Clonal Selection Algorithm

The theory of clonal selection and immune memory [24], proposed by Burnett, was the inspiration behind Clonal Selection Algorithm (CSA) [65]. Based on another population of immune cells, more precisely B cells, the theory states that these cells can produce antibodies triggered by the stimulation of invading pathogens, known as antigens. The mechanism is described as the maturation of B cells into plasma cells, activated by these antigens, which, after activation, release antibodies and memory cells. These memory cells can "remember" previous exposures to a particular antigen, forming the immune memory. Subsequently, the mechanism only maintains antibodies with a higher affinity for the antigens. In the end, after several exposures, the B cells will increase the affinity of their antigen-antibody bond, which means that a higher affinity bond will contribute to the production of more B cell clones. On the other hand, B-cell mutation is inversely proportional to antigen-antibody binding. This process is called hypersomatic mutation, which occurs at an extremely high rate. Figure 2.3 shows a clear visualisation of the theory of clonal selection.

Although CSA and evolutionary algorithms are closely linked [66], they differ in some fundamental aspects. For example, the affinity of the antibody-antigen bond controls the selection and mutation mechanisms, evolutionary algorithms do not have a crossover mechanism and it is usually possible to find several optimal solutions.

The idea behind CSA is that only cells with the ability to recognise antigens will proliferate. Castro and Von Zuben, therefore, devised what is known as CLOnal AlGorithm (CLONALG) [67, 68]. This was the first algorithm to consider clonal selection and affinity maturation to recognise patterns. One of its main features is exploring the search space, using affinity as a criterion. Initially, there is a randomly selected sample representing antibodies, which will be compared

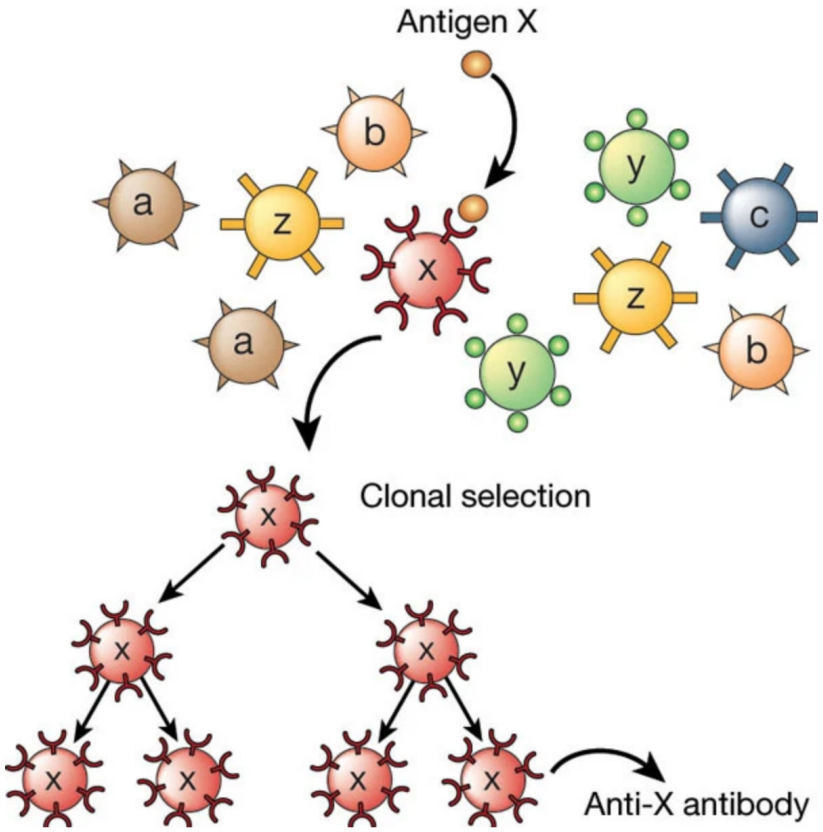

Figure 2.3: The theory of antibody production through clonal selection. There is just one kind of antibody receptor produced on the surface of each B cell. One B lymphocyte is recognized by an antigen from a broad repertoire. This causes the B cell to divide quickly and differentiate into a "plasma cell," which produces and secretes antibodies that are unique to the original antigen [3].

for each antigen using a metric representing affinity. In this context, the pattern is the computational representation of the antigen. According to the criteria, various antibodies are selected and replicated until the optimum state is reached.

A variety of works with respect to the CSA have been developed, with new characteristics and concepts when compared to the previous. Timmis and Neal proposed a way to overcome limited resources [69]. Their solution was artificial recognition balls that contain a number of identical B-cells, represented now as clusters. As an extension of CLONALG, the adaptive immune mechanisms of B-cells are set along with the immune network theory [70]. B-cell properties also were computationally traduced through an algorithm [71]. This algorithm was used to perform function optimisation challenges using a single contiguous hypermutation operator. It demonstrated a considerable reduction in evaluations compared to a hybrid genetic algorithm without sacrificing the quality of the solutions obtained. A technique based on clonal selection known as MISA [72] has been shown to converge using Markov chain theory in preliminary work carried out by VillalobosArias *et al.* [73]. A generic CSA is presented in Algorithm 2.

Algorithm 2 Generic Clonal Selection Algorithm

Require: S = set of patterns to be recognized, n = the number of worst element to remove
Ensure: M = set of memory detectors capable of recognizing unseen patterns

```
Begin
Create an initial random set of antibodies A;
for all the patterns in S do
    Determine the affinity with each antibody in A;
    Generate clones of a subset of the antibodies in A with the highest affinity, the number of
    clones is inversely proportional to its affinity;
    Mutate attributes of these clones inversely proportional to its affinity;
    Add these clones to the set A, and place a copy of the highest affinity antibodies in A into
    the memory set M;
    Replace the n lowest affinity antibodies in A with new randomly generated antibodies;
end for
End
```

2.2.3 Immune Network Algorithms

The clonal theory of immunity also provided the basis for developing the Artificial Immune Network (AIN), which describes the dynamic interactions between immune cells. Proposed by Jerne [74], the immune network theory, also known as the idiotypic network theory, was developed in 1974. This theory implies that immunological entities interact with each other in the immune system, even in the absence of antigens, functioning as a kind of network. This means that these interactions can occur only between antigens and antibodies, but they can also occur only between antibodies, as Figure 2.4 shows. As a result, they can induce both stimulatory and suppressive responses, causing immunological behaviours such as tolerance or the emergence of memory. This stimulation of B cells can be affected by the suppression of neighbouring B cells, the contribution of antigen binding and the contribution of neighbouring B cells [75].

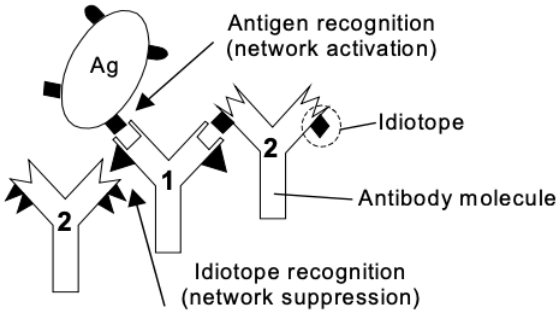

Figure 2.4: Network activation occurs when an antibody recognises an antigen via a cellular receptor, and network suppression occurs when an antibody recognises an idiotope. [4].

There are two approaches to modelling immune networks, and they can be either discrete or continuous. The first method is based on iterative adaptation processes to solve problems, while differential equations inspire the second to support the behaviour of immune networks. Algorithm 3 shows a generic immune network.

The idea of an immunological network is not widely accepted by immunologists. The theory of immunological networks aims to represent a complex system in an even more sophisticated way than existing models, which abstract a simplified version of biological events. The ability to take into account interactions between immunological components, such as interactions between antibodies, distinguishes algorithms based on immune networks from other AIS approaches.

Algorithm 3 Generic Immune Network Algorithm.

Require: S = set of patterns to be recognised, nt = network affinity threshold, ct = clonal pool threshold, h = number of highest affinity clones, a = number of new antibodies to introduce
Ensure: N = set of memory detectors capable of recognising unseen patterns

```
Begin
Create an initial random set of network antibodies N;
repeat
    for all the patterns in S do do
        Determine the affinity with each antibody in N;
        Generate clones of a subset of the antibodies in N with the highest affinity, the number
        of clones is proportional to its affinity;
        Mutate attributes of these clones inversely proportional to its affinity;
        Place the h number of highest affinity clones into a clone memory set C;
        Eliminate all elements of C whose affinity with the antigen is less than ct;
        Determine the affinity amongst all the antibodies in C and eliminate those antibodies
        whose affinity with each other is less than ct;
        Incorporate the remaining clones in C into N;
    end for
    Determine the affinity between each pair of antibodies in N and eliminate all antibodies
    whose affinity is less than nt;
    Introduce a number a of new randomly generated antibodies into N;
until a stopping condition is met;
End
```

As mentioned earlier, AIN was inspired by the theory of clonal selection. The Artificial Immune Network algorithm (aiNet) is, possibly, the best-known approach that was inspired by AIN [76]. This algorithm is a modified version of CLONALG that takes into account the suppressive interactions between antibodies. As demonstrated in [77], the aiNet technique has become very popular for solving optimisation problems. However, the recent theoretical analysis of aiNet carried out by Stibor and Timmis [78] revealed that the algorithm's shortcomings apply to situations of non-uniformly distributed data clustering.

2.3 The Danger Theory

DCA is based on DCs, whose behaviour and properties play a part in the body's first line of defence against pathogens [18, 79]. These immune cells are able to combine different sources of information to detect patterns in order to translate this information to T cells. Based on this, DCs

are responsible for both detecting invaders and triggering immune responses. The type of immune response depends on the process of revealing antigens from the DCs to the T cells and the signals collected from the surrounding environment. Signals are a "metric" interpreted by DCs to measure tissue health.

Initially, as Figure 2.5 show, DCs present themselves as immature to the body. The context where they are inserted and the signals grabbed from there induce differentiation in those cells. Depending on that, DCs can differentiate into semi-mature or mature cells. A DC is more likely to develop into a semi-mature DC if it is exposed to a combination of signals produced by a stable state or a healthy tissue environment, such as one in which no tissue damage occurs. On the other hand, a DC is more likely to develop into a fully mature DC if it receives a combination of signals from a damaged tissue environment, such as the presence of uncontrolled cell death [80].

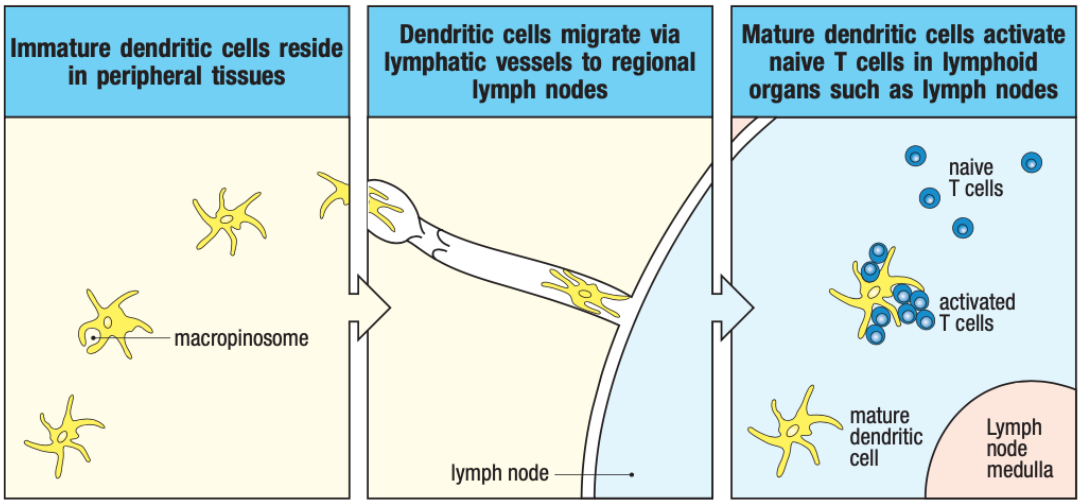

Figure 2.5: Development and role of DCs in the immune response: From peripheral tissue surveillance to T-cell activation in lymph nodes [5].

Since DCs are heavily influenced by the aforementioned signals, there is a different type of receptor for each detected signal, namely Pathogen-Associated Molecular Pattern (*PAMP*), Safe signals (*Ss*) and Danger signals (*Ds*). Each type represents the scale of the threat. *PAMP* reveals danger with a high confidence level; *Ss* shows the lack of danger with medium confidence, and *Ds* detects the threat but with some reservations [40].

In parallel with the above, DCA was conceived as an abstraction of these DCs [16] by Greensmith. The algorithm accepts two types of inputs as DCs, such as "signals" and "antigens". Computationally, signals are measurements of the state of a monitored system at specific time intervals and are expressed as vectors of real numbers. Antigens are categorical variables that can be things of interest linked to a monitoring system or different states of a problem area. While signals tend to refer to a more generalised context, based on the environment and the interaction between elements, antigens refer to a more specific part of a problem domain - which problem in an element causes the overall result [32, 14].

Algorithm 4 Generic Dendritic Cell Algorithm.

Require: antigen and signal instances
Ensure: antigen types and anomaly metric K_α

```
set DC population size;
initialise DCs;
while data do do
    if antigen then then
        agCounter++;
        cellIndex = agCounter mod populationSize;
        DC of cellIndex assigned antigen;
        update DC's antigen profile;
    end if
    if signal then then
        calculate csm and k;
        for each DC do do
            DC.lifespan-- = csm;
            DC.sumK+ = k;
            if DC.lifespan ≤ 0 then then
                record antigens and DC.sumK;
                reset DC;
            end if
        end for
    end if
end while
for each antigen type do do
    calculate K_α;
end for
```

Just as in the human body, the DCA also detects and alerts the system using the same concepts mentioned above: *PAMP*, *Ss* and *Ds*. Computationally, *PAMP* can be seen as an increase in value when unusual behaviour is detected. This is a reliable sign of an anomaly, which typically manifests itself as indicators of occurrences that are certain to cause systemic damage. For its part, *Ss* shows increases in values which, together with the observed normal behaviour, are a reliable indicator of the normal, steady-state or predictable behaviour of the system. Finally, *Ds* shows possible anomalies; as the values increase, so does the monitored system's confidence that its state is aberrant. As demonstrated in the natural system, increasing the value of the safety signal reduces the impact of the *PAMP* and *Ds* on the algorithm.

Since the DCA is a population-based algorithm, it begins with a random population of DCs with the aim of detecting invaders. As a result, DCA can combine many input signals to determine the current context of the environment. As there are many random-based components in the original version of DCA, the method is extremely stochastic. A dynamic time window effect is created in the population, assigning a specific lifetime limit to each DC [81]. After this, the algorithm does a temporal correlation between signals and antigens, allowing the capture of a causal relationship. An individual DC internally stores the antigens sampled during detection in order to

update its antigen profile. A new immature DC is established with default values in place of the mature DC once it has supplied the processed data. Algorithm 4 illustrates a generic version of what was stated.

2.4 The Dendritic Cell Algorithm

Over the last few years, the DCA was considered and modified through several works. This is a demonstration of the diversity associated with the algorithm and how powerful it can be. Initially, the DCA was created and applied as an anomaly detection method. More specifically, DCA was created with the intention of using it to solve network intrusion detection problems in the future.

Various works have been carried out in the literature to extend the standard version of the DCA. Some academics have focused on simplifying the principles of the DCA by eliminating or replacing parts of its elements. There are two different DCA implementations. The first version is the generic approach, as previously mentioned, and the second is the deterministic version of the DCA (dDCA) [82].

Before arriving at the deterministic version of the algorithm, different architectures were tested. After the first abstract model, a prototype system (pDCA) was used to show the applicability of DCA [57]. To show that this population-based approach was computationally capable of discriminating binary classes in an ordered data set, the pDCA was applied to a common ML problem. In this application, three categories of signals were composed using an antigen formed by a timestamp and a mixture of pre-processed features. The next version (ltDCA) [18] was stochastic in nature, with many highly unpredictable components such as randomly assigned lifetimes, signal degradation and random sampling of antigens during exposure to the received antigen.

Several adjustments to ltDCA were necessary to obtain an acceptable mapping of signal and antigen data. This involved integrating a re-implemented ltDCA into a robotic version [83], known as robotic DCA (rDCA), and segmenting the output information to provide localisation information for a discovered anomaly. The method had a significant number of random elements removed to simplify the evaluation and analysis of the system. Later, a simplified variant known as deterministic DCA (dDCA) was created [82].

2.5 Signal Categorization

Numerous research papers have been created on the feature selection/reduction and signal categorisation sub-steps of the DCA data preparation phase. In fact, in the beginning, specialised or domain knowledge was required for data pre-processing, and DCA did not rely on a training data stage. It was said that this task manually adjusted the data to the algorithm, which was not what was desired.

In [32], PCA statistical method was presented for an automatic pre-processing task. By creating new features to keep and performing their categorization to their respective signal kinds (*Ss*, *Ds*, *PAMP*), PCA is used to reduce data dimension automatically. Information gain and correlation

coefficient were also tested and discussed in [31]. However, the use of dimensionality reduction methods, and more precisely, data reduction methods such as PCA, during the data pre-processing phase of PCA destroys the intrinsic meaning of the features that were originally included in the input database. This is considered a negative feature of various data preparation methods, including PCA.

Regarding the second sub-step of the DCA data pre-processing phase, signal classification, the study of [31] suggested using the mean squared error (MSE), which is the predicted value of one particular utility function known as the quadratic utility function less than one. Approximate DCA versions [84, 85, 86] are more recent additions to the DCA data preparation phase. The application of Rough Set Theory (RST) [87] to the problem of DCA data preparation is the basis of the approximate DCA algorithms that have been suggested. As an initial sub-step of the DCA data pre-processing phase, feature selection is carried out using RST concepts. The algorithms eliminate superfluous attributes and retain only the most informative ones, a subset known as reduct, which retains almost the same classification power as the original training dataset. The second sub-stage of the DCA data preparation phase, signal classification, is based on the reduct and core ideas of RST.

Despite all of these applications, the biggest limitation in the DCA is still linked to its data pre-processing phase, which is performed either manually by users based on their expert knowledge of a given problem domain or automatically using some feature reduction methods. Table 2.1 shows the main works on investigating the pre-processing phase.

Table 2.1: Main works on investigating pre-processing phase and signal categorization.

Approach	**Overview**
PCA, information gain and correlation coefficient [31, 32]	Feature reduction
PCA and MSE [32]	Signal categorization
RST [84]	Feature selection
RST [85]	Feature selection
QuickReduct Algorithm [86]	Signal categorization
Fuzzy RST [88]	Feature selection
Fuzzy RST [89]	Signal categorization

2.6 Summary

In this chapter, different bio-inspired approaches were described, and their impact on artificial intelligence and ML was highlighted. The literature overview made during this chapter shows the interdisciplinary nature of these algorithms, describing successful approaches such as GA or ANN.

Moreover, an extensive review of the HIS and its detailed components and mechanisms are also part of this chapter. This foundation provides knowledge for exploring AIS, which draws inspiration from HIS principles to solve complex computational problems.

Furthermore, the Danger Theory and the DCA are also mentioned. Firstly, the theory and its contextual concepts are described. After that, inspired by the dendritic cells of the immune system, DCA was presented as a reliable technique for identifying anomalies. Its versatility and effectiveness were demonstrated by discussing many iterations and adjustments.

Lastly, the chapter discussed the importance of data pre-processing and signal classification in DCA. To highlight their contributions and limits in improving DCA performance, various methodologies and methods were examined, including PCA, RST and other feature selection techniques.

Chapter 3

Feature Engineering

Feature Engineering (FE) is one of the most nuclear components when it comes to artificial intelligence, more specifically ML. In practice, it is the process of creating useful features from pre-existing features in order to increase models' performance. The process involves the application of transformations to the original set of features, either with arithmetic or aggregation operators, to generate a new dataset to feed the model. Transformations make it easier to scale a feature or convert a non-linear relationship into a more easily learnt linear relationship between a feature and a target class.

Typically, a human performs FE using their domain knowledge, iterative trial and error and model evaluation. Some current methods use guided search in the feature space with heuristic measurements of feature quality (such as information gain) and other surrogate performance metrics to perform automatic FE [90]. This process is applied to raw data collected from different sources. However, the ability to distinguish between features and raw data is essential for making the kind of decisions necessary for successful FE [91].

From all the techniques used to mutate features, dimensionality reduction is one of the most important phases of the data pre-processing phase [92]. It involves creating a new and lower-dimensional set of features that accurately reflects the initial characteristics. Two approaches can be used to reduce the dimensionality of the dataset: one is feature selection, which involves keeping only the most significant variables from the original dataset; the other is feature reduction, which involves taking advantage of redundancy in the input data to find a smaller set of new variables that are combinations of the input variables and contain essentially the same information as the input variables [93].

As far as feature selection techniques are concerned, they can be categorised into three different branches, namely filtering methods, wrapping methods and embedded methods. As a pre-processing step separate from the selected classifier, filters choose subsets of features. Using a "black box" approach, wrapping methods evaluate feature subsets based on their ability to anticipate results. Usually tailored to specific classifiers, embedded methods select features during the training process [94].

With respect to feature reduction, the type of data determines the adopted technique. The

data can either be linear or nonlinear, which means that traditional linear approaches may struggle with real-world data since it is likely to be nonlinear. Previous research has shown that non-linear methods perform better on complex artificial problems than their linear counterparts [95]. Over the years various contributions have been made to this field, both for dealing with linear data, but above all for non-linear data, namely Principal Components Analysis (PCA), Locally Linear Embedding (LLE) or Maximum Variance Unfolding (MVU) [96].

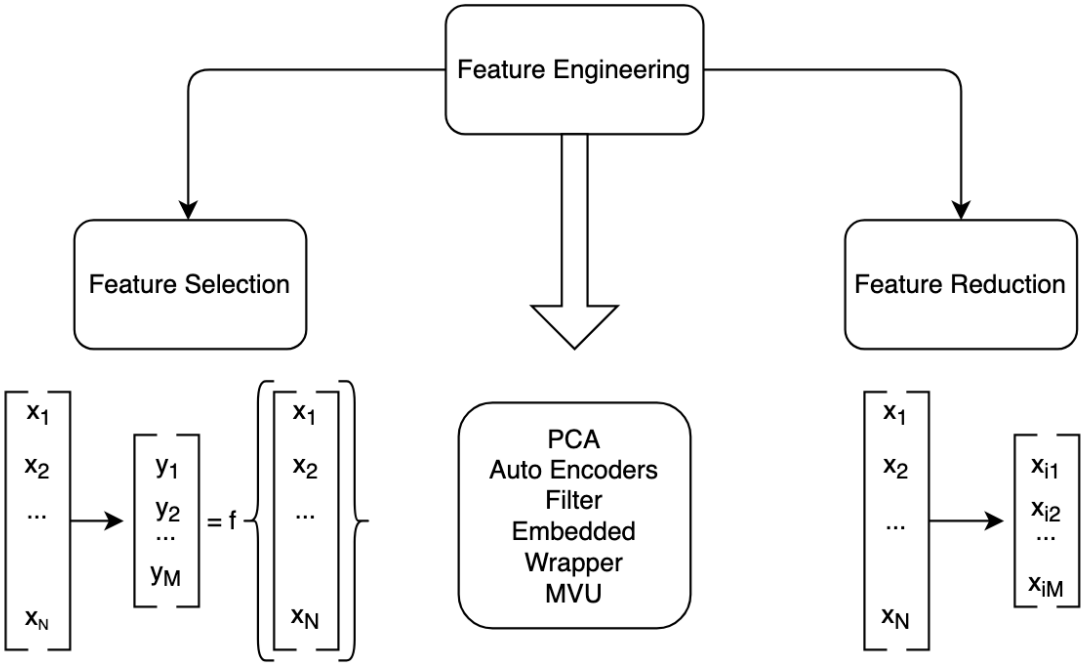

Figure 3.1: Showing various feature extraction and feature selection methods for dimensionality reduction.

3.1 Feature Selection

Since, for many real-world problems, most features decrease model performance, both in terms of speed (due to high dimensionality) and prediction accuracy (due to redundant and irrelevant information), it is important to choose the appropriate technique for selecting those features that describe the target concept and feed into the model. As already stated, there are three general approaches to selecting features, such as filtering methods, wrapping methods and embedded methods [97].

3.1.1 Filter Methods

As mentioned above, filter methods determine a score for each feature based on the dataset without considering the ML algorithm used. They rank the features according to their estimated merit based on this scoring system or select a subset of the top features [98]. This method is computationally efficient and quick to calculate. However, certain features that are not useful on their own but which become extremely useful when combined with other characteristics can be ignored by filter methods.

The feature filtering measures developed can be broadly categorised into information, distance, consistency, similarity, and statistical measures. Multivariate feature filters evaluate an entire subset of features, while univariate feature filters evaluate (and usually classify) a single feature [99]. Among all the algorithms devised, information gain, correlation, and chi-square are examples of techniques that can be used [100, 101, 102].

3.1.2 Wrapper Methods

The wrapper and filter methods differ only in the way the evaluation is carried out. In wrappers, evaluation is carried out using a learning algorithm, which means that the model selects the set of features that best suits the objective [103]. Consequently, in classification tasks, a wrapper evaluates subsets according to the performance of a classifier (such as Naïve Bayes or Support Vector Machine (SVM)) [104, 105], but in clustering tasks, a wrapper evaluates subsets according to the effectiveness of a clustering algorithm (such as K-means) [106].

Like filters, the search method determines the generation of the subset and the evaluation is repeated for each subset. As wrappers depend on the resource requirements of the modelling process, they identify sufficiently suitable subsets much more slowly than filters. It is possible to create a wrapper from almost any combination of modelling algorithm and search strategy, but it is only practical to use wrappers with fast and greedy modelling algorithms, such as Naïve Bayes [107] or linear SVM [108].

3.1.3 Embedded Methods

Compared to the wrapper approach, this method interacts with the learning algorithm at a reduced computational cost. It records feature dependencies as well. In addition to considering the relationships between a single input feature and the output feature, it also looks locally for characteristics that improve local discrimination. A given cardinality determines the ideal subsets using independent criteria. The ultimate optimal subset among the optimal subsets across various cardinalities is then chosen using the learning process [103].

Some embedded techniques require the feature coefficients to be tiny or exactly zero while weighting the features based on regularisation models with objective functions that minimise the fitting errors. These techniques based on Lasso [109] and Elastic Net [110] usually employ linear classifiers (SVM or others) and penalise features that add nothing to the model.

3.2 Feature Reduction

In order to reduce the amount of resources needed for processing without losing any relevant features of the datasets, the feature extraction approach is very useful for extracting new features from the original dataset. The feature reduction method reduces the high dimensionality of the feature vector, generating new features that are based on the original input feature set, as previously stated.

These dimensionality reduction strategies preserve the original relative distance between features and cover the potential structure of the original data in an effort to minimise the amount of information lost during the feature translation process [111]. One advantage over feature selection methods is that feature extraction is less susceptible to overfitting and performs well in terms of classification accuracy. However, after transformation, the description of the data is occasionally lost, and, in some datasets, the cost of this operation is high [112, 113].

3.2.1 Kernel PCA

The aim of PCA is to model the linear variability of high-dimensional data. On the other hand, many high-dimensional data sets are non-linear. In these situations, PCA cannot accurately represent the variability of the data because the high-dimensional data is located on or near a non-linear manifold rather than in a linear subspace.

In Kernel Principal Components Analysis (KPCA), the principal components can be efficiently calculated in high-dimensional feature spaces that are connected to the input space by a non-linear mapping using kernels. By performing PCA in the space created by the non-linear mapping, where the low-dimensional latent structure should be simpler to localise, KPCA finds principal components that are non-linearly related to the input space [114]. This non-linear variation of PCA, KPCA [115], is effective and can better reflect the information by capturing some of the higher-order statistics [116].

Since this approach is more flexible than the traditional PCA, it allows the discovery of patterns and relationships that linear methods might miss. However, selecting the appropriate kernel must be considered because different kernels can capture different types of non-linear relationships [117].

To perform a kernel-based PCA, known as KPCA, the kernel matrix must first be calculated. All pairs of data points are submitted to the kernel function selected to create this matrix. Then, using the calculated kernel matrix, an eigenvalue problem must be solved. The algorithm diagonalises the matrix to find the eigenvalues and the corresponding eigenvectors. When the product of each eigenvector by itself is equal to 1, the expansion coefficients of the resulting eigenvectors must be normalised. Finally, to find the principal components, the algorithm projects each test point onto the eigenvectors acquired in the previous step [118].

3.2.2 Autoencoders

An Autoencoder (AE) is an artificial neural network that can use unsupervised learning to acquire new encodings of information [119]. This architecture uses a sequence of hidden layers to learn how to reproduce an input in the output. Through the hidden layers, AEs can compress the original representation to obtain a reliable representation of the original data. These properties and performance show that AE is well suited to dimensionality reduction tasks [119, 120].

The fundamental structure of an AE is very similar to that of a multilayer perceptron. Since an AE has no cycles, it works like a feedforward neural network in which data always flows in one direction. A sequence of layers, including an input layer, several hidden layers and an output layer - the units of each layer coupled to those of the subsequent layer - usually make up an AE. Since the main purpose of AEs is to replicate the input at the output during the learning process, the output and input have the same number of nodes [121].

An AE can be divided into two sections: the encoder and the decoder. The first section comprises the input layer and the first segment of hidden layers. The second section comprises the output layer and the second half of the hidden layers. This architecture is symmetrical, as shown in Figure 3.2.

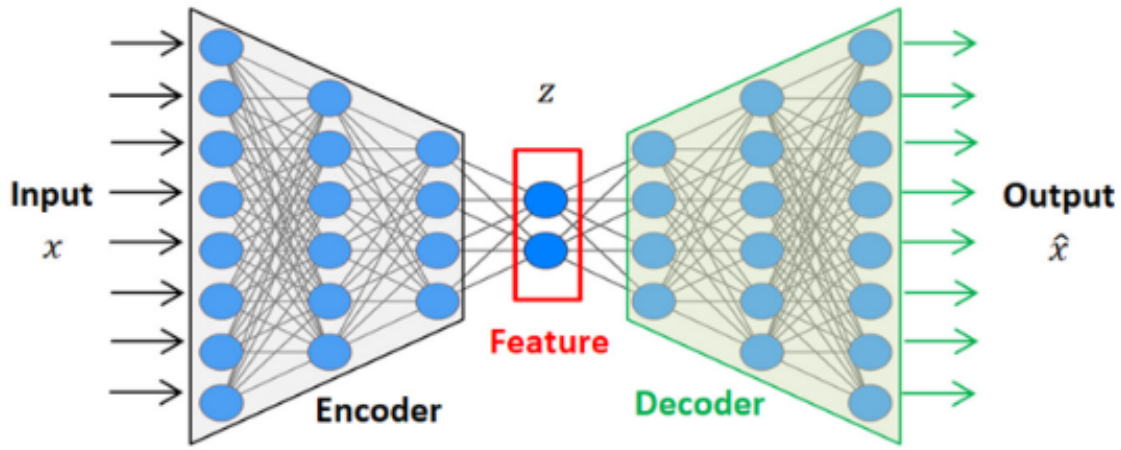

Figure 3.2: General architecture of an AE [6].

AE tries to reduce the reconstruction error during training. The weights are modified after the error has been back-propagated through the network to reduce the error obtained. At first glance, learning to reproduce the input in the output may seem pointless. However, what matters more than the output is a new representation of the original data that has been learnt in the hidden layers. This makes it possible to obtain very useful features [122].

The number of hidden layers is the criterion for classifying the two types of AEs. On this basis, there are those that have fewer units in their hidden layer than in their input and output layers and those that have more units in their hidden layers than in their input and output layers. The first type is called undercomplete. Its main purpose is to force the network to extract new and more sophisticated features, forcing it to learn a condensed version of the input. The other type is

called overcomplete. As the name suggests, in this case, the network can learn to copy the input to the output instead of learning something useful [7].

3.2.3 AEKPCA

As already mentioned, automating the pre-processing phase with dimensionality reduction is necessary for DCA applications. As a result of this problem, different methods have been tested and implemented that are capable of resolving this issue. However, there is still work to be done.

Autoencoder with kNN (AEkNN) [7] is a new approach that presents a variation of an AE to address this problem of selecting the best set of features without losing meaningful information. This type of neural network is suitable for this task due to its structure and capabilities. As mentioned before, AEs learn to reproduce the input at the output through a sequence of hidden layers. In AEs, the input and output layers often have more units than the intermediate layer. Consequently, this bottleneck forces the network to acquire a more condensed and sophisticated representation of the data. In the classification phase, the coding generated by this intermediate layer can be retrieved and used as a new collection of features. Based on that, some works have demonstrated that it is possible to obtain better results approaching the problem with AEs than with traditional methods such as PCA or multidimensional scaling [123].

This algorithm is an instance-based approach designed to deal with classification problems involving many variables. Firstly, it starts by working with a feature space and projects that space onto a new feature space, smaller than the first one, creating new features. These updated features are expected to record more significant data than their predecessors. After creating this new representation of the data, the algorithm calculates the distances between the data points considering only the new features generated. This process has become useful since dealing with high-dimensional data can be challenging. In the last phase, the prediction phase, the algorithm estimates the result based on the test sample provided.

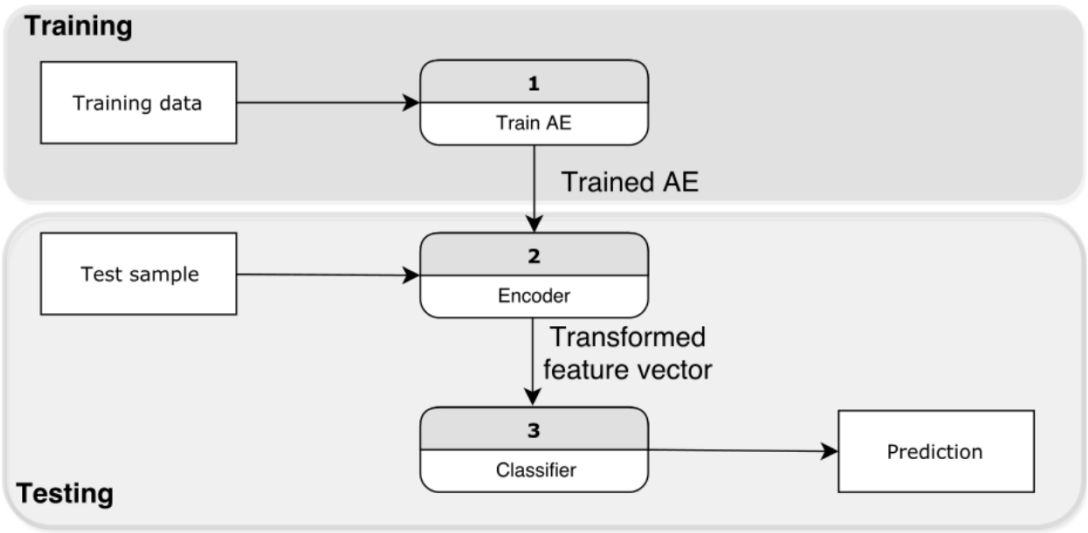

Figure 3.3: Description of the method [7].

AEkNN is not a lazy algorithm, unlike some other techniques; instead, it starts by building a model to produce the new features. This method can improve runtime and prediction performance, especially when working with large feature datasets. This is an advantage since DCA is also lightweight, and the solution to the problem can retain this characteristic.

There are two main steps that define these algorithms. To create the AE model, which enables the creation of a new data encoding, the learning phase is first completed using the training data. The next step is the classification run. The test data is resubmitted using the model created in the first phase, and the class of each instance is then predicted using the neighbouring instances. Figure 3.3 illustrates the general process.

Since this approach uses the kNN to evaluate the performance of the AE, it does not serve our purposes. However, this combination was the inspiration behind the third technique used in this dissertation. Instead of combining the AE with the kNN algorithm, this study combines the AE to generate a different representation of the data with the same shape of the input data with the KPCA to reduce the number of features. This combination allows us to embed both techniques that this study aims to use. Additional details will be stated further. The technique will be called Autoencoder with Kernel Principal Components Analysis (AEKPCA).

3.3 Summary

This chapter provided a comprehensive analysis of the FE process, which is a crucial step of any ML project. The impact of transforming and manipulating pre-existing features, as well as creating new ones, is linked to models' performance. Within FE, different techniques are covered, including feature selection and feature reduction, and their importance in reducing dataset size without losing meaningful information is also stated.

Based on that, there are three main types of feature selection techniques that were categorized in this chapter: filter methods, wrapper methods and embedded methods. Each of these was explained, and their applications on feature selection were highlighted. While wrapper approaches use learning algorithms to evaluate subsets of features, filtering methods rely on statistical measures and embedded methods include feature selection in the training process.

There was a detailed discussion of feature reduction approaches, including KPCA and AEs. Modelling non-linear variability in high-dimensional data is a useful use of KPCA, a non-linear extension of PCA. The ability of AEs - neural networks built to acquire compressed representations of data - to reduce dimensionality through unsupervised learning was emphasised.

An innovative method called AEKPCA was presented. By combining the best features of the two approaches, this strategy produces a more concise and insightful representation of the data that makes categorisation more accurate and efficient. By automating the pre-processing phase of DCA applications, the AEKPCA technique seeks to overcome the difficulties associated with feature selection and reduction, while preserving the robustness and lightness of DCA.

Chapter 4

Scientific Methodology

The following chapter details the scientific methodology that this dissertation followed and adopted to assess the DCA's performance, integrated with the proposed solution previously mentioned. In addition, this chapter is divided into three sections, namely research design, data collection and pre-processing, and data analysis. Over the three sections, the foundations and the setup behind the scientific methodology followed are described, which establishes the foundations of the tests and results further discussed. The last section emphasizes the alignment of this study with the Sustainable Development Goals (SDGs).

4.1 Research Design

Several parameters were considered to settle equal conditions during the research, ensuring the study is repeatable and reproducible. As mentioned before, there is a gap in the literature concerning the pre-processing phase associated with the DCA.

This dissertation proposes a solution to this issue using three feature reduction techniques: KPCA, AE and AEKPCA. Besides this, the DCA has been written from scratch to propose a version of this particular algorithm that suits the needs of the AIS community. Each technique was fed different dataset types, creating a new and lower-dimension data representation.

Throughout this process, it's important to mention that the goal was to reduce the number of features without losing meaningful information. Once the new data representation was made, the DCA ingested it to classify the presence of anomalies. The DCA's performance was evaluated through classic ML metrics, and his metrics were stated by Greensmith [16]. The study was also conducted to answer the three research questions formulated in the chapter 1.

4.1.1 Research Approach

This dissertation used qualitative and quantitative data across different areas to extensively comprehend the problem and ensure various results. Some transformations were necessary to prepare the data for the models that were covered in 5.1. Since the DCA is a lightweight algorithm that

can perform quickly, the time was considered. So, besides the evaluation metrics previously mentioned, time was also a metric to evaluate DCA's performance.

A dataset created with a high correlation between some features and the predictable variable was used to write and debug the algorithm. This was the benchmark test used to compare the performance of the feature reduction techniques. A detailed explanation will be further provided.

4.1.2 Research Strategy and Context

The strategy used implies the same parameters over the different tests to ensure comparisons, which means that controlled experiments were conducted to test the performance of each feature reduction technique. Moreover, a comparative analysis was performed to evaluate the effectiveness of these techniques.

It's important to note that this dissertation was conducted in the context of anomaly detection through computational biology algorithms. The datasets were selected to cover different application areas, such as network intrusion data, Distributed Denial-of-Service (DDoS) attacks, wireless sensor network data and biomedical signals. This variety of tests provides a different range of results to test the versatility and robustness of the DCA.

4.2 Data Collection and Pre-Processing

Different data sources from various domains were considered for testing the algorithm. After some research, it was decided to include data from port scanning, DDoS attacks, credit card fraud detection, stroke and heart failure in the study.

Since all the data was gathered, cleaning and transforming the data was essential before feeding the feature reduction algorithms. There were no missing values in the selected datasets, which makes things easier. Each dataset's categorical data was encoded and transformed into numerical using different techniques, namely one-hot encoding, frequency encoding, or IP converter. Once all the data became numerical, all the features were normalized using Z-score standardization, which scales the data based on the mean and the standard deviation, ensuring that all the features have the same weight in the feature reduction process.

4.3 Data Analysis

To interpret the results and to answer the research questions that this study proposes, it's crucial to define metrics that will allow us to conclude. This step has the same importance to the final goal as the ones mentioned so far.

Thus, it was defined using four different metrics: running time, anomaly counter, accuracy and Mature Context Antigen Value (MCAV).

The running time is collected to maintain the algorithm's lightweight property; this metric works as a control. Finding a way to pre-process the data to give the algorithm autonomy is

important, but keeping the running time the lowest possible is also important. So, a time variable starts with the DCA and stops when the algorithm also stops. The difference is the time that it took to run the algorithm.

The anomaly counter is a metric that gives us the absolute number of anomalies found. Every time the DCA finds an anomaly one unit is added to the variable previously initiated. The final number is the total number of anomalies.

To compute the accuracy, the ML accuracy formula is considered which is the ratio between the sum of the true negative and true positives over the total number of instances.

The last metric, MCAV, was stated in DCA's formulation paper [16]. This metric returns a value between 0 and 1, which allows for applying an anomaly threshold to detect if the anomaly is present; in this particular case, it was 0. It is the ratio between the number of antigens presented by the DCs and the total number of antigens.

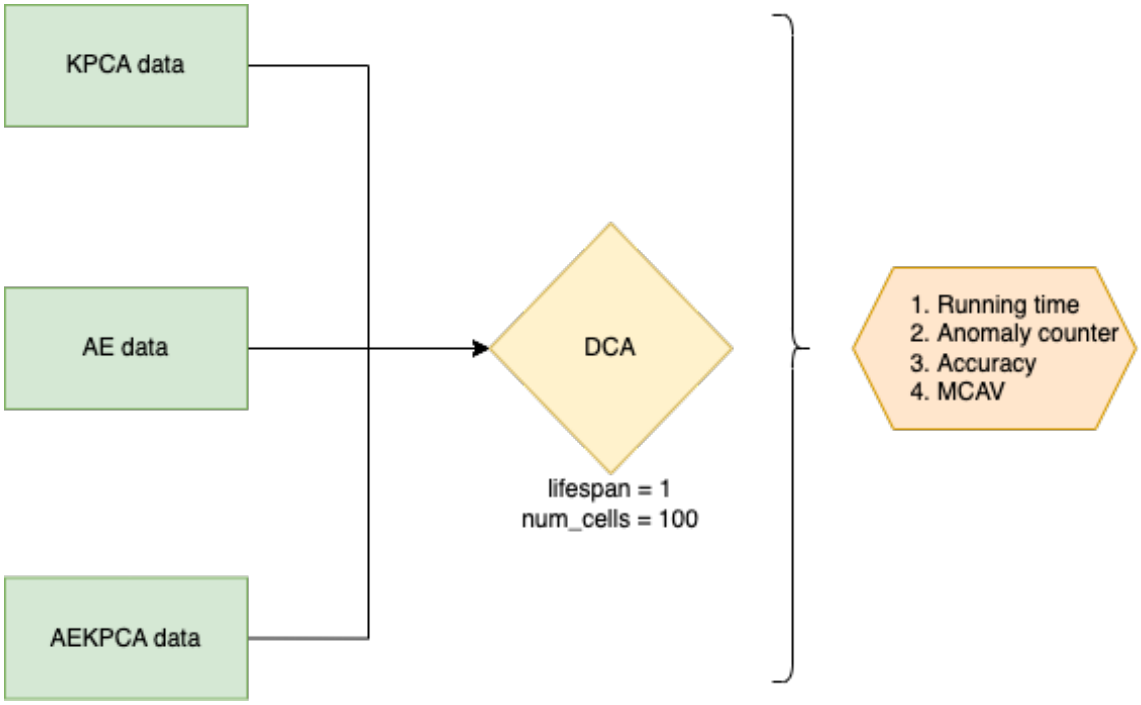

Figure 4.1: DCA pipeline.

Concerning the DCA itself, it was developed with two parameters as the dDCA. The *num_cells*, which represents the number of DCs present, was set to 100, whereas the *lifespan*, which represents the "duration" of each cell, was set to 1 to match the data normalization. Figure 4.1 illustrates the DCA's running pipeline.

Chapter 5

Proposed Solution

In this chapter, the proposed solution idealized to address the previously mentioned literature gap is presented. This solution focuses on incorporating advanced data pre-processing techniques to improve the efficiency and accuracy of DCA in detecting anomalies in various domains.

This chapter is a two-section chapter. Firstly, detailed architectures, tools and parameters within each technique are explained and detailed. A deep analysis of each feature reduction technique describes the adopted configurations. The last section refers to the hardware and software used during the dissertation, which ensured the consistency and reproducibility of these tests.

5.1 Architecture

An experimental setup was designed to create a workable and dynamic pipeline to test the effectiveness of the DCA in anomaly detection. This experimental setup includes both procedures, tools and parameters used to perform the experiments, ensuring that all the tests were performed within the same environment.

5.1.1 Feature reduction

As mentioned before, three approaches to address the literature gap that motivated this dissertation were implemented. As it should be, these techniques were not tried before, which makes the proposed solution even more interesting and challenging. Every configuration and parameter used to test the feature reduction solution will be detailed further.

5.1.1.1 Kernel Principal Components Analysis

KPCA is more flexible than the traditional PCA, which allows the discovery of patterns and relationships that linear methods might miss. This approach implies the selection of the appropriate kernel since different kernels can capture different types of non-linear relationships.

As the other techniques that will be further detailed, this one was also used to reduce the number of features without losing meaningful information, which means that no matter the size of

the input, the model will be trained to produce two features that will represent the danger signal and the safe signal later ingested by the DCA.

The Scikit-learn Python ML library was used to train the KPCA and produce the aforementioned new data representation. Regarding the hyperparameters that had to be trained, it's important to mention that the goal was to keep the models as simple as possible so they could be applicable in different domains. The *n_components* was set to two, which means that the KPCA will produce two features, and the kernel type was passed as an array with the different kernels, such as *linear*, *poly*, *rbf* and *sigmoid*.

All the selected datasets were used to feed this model, and each had a new representation generated by each specific kernel. Figure 5.1 represents the pipeline used.

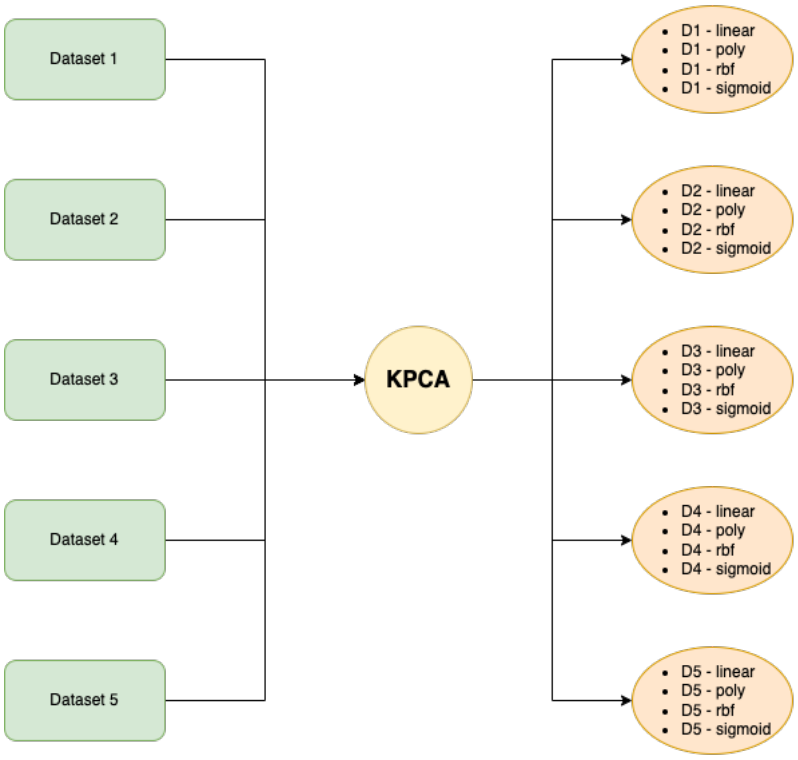

Figure 5.1: KPCA pipeline.

Once the data had been created, the KPCA's efficiency was evaluated not through ML traditional metrics but through DCA's performance. In this specific case, it's not the performance of the KPCA itself that matters but the accuracy of the DCA when it comes to classifying anomalies with pre-processed data with this technique. The results will be analyzed further.

5.1.1.2 Autoencoder

AEs are powerful for feature reduction because, like KPCA, they are well known for capturing the non-linear relationships within the data. Besides that, with their symmetrical architecture, they learn a new representation of the data by minimizing the reconstruction error, and the encoder part can be used as a dimensional reduction tool. In the context of the DCA, the subject of this study,

AEs should help pre-process high-dimensional data without losing meaningful information and enhance the DCA's capabilities to detect anomalies.

The Pytorch Python ML library was used to train the different AEs treated in this dissertation. Similarly to the previous approach, there was an intention to have a general approach that could suit the different domains, so the AE configuration was kept the same in all the different tests.

The network architecture is composed of symmetric parts, as mentioned before. The *num_in* parameter is the number of neurons of the input layer, which may vary according to the chosen dataset. Within the encoder part, the first hidden layer is connected to the input layer, composed of 128 neurons. Consequently, the second hidden layer is composed of 256 neurons and connected to the first hidden layer. The bottleneck layer, composed of 2 neurons to retrieve the two desired signals, represents the latent space. On the other side, the decoder part, the bottleneck layer, is connected to the first hidden layer, composed of 256 neurons. Therefore, the previous layer is connected to the second hidden layer, composed of 128 neurons. In parallel, *num_out* parameter has the same size as *num_in* and composes the output layer. The activation function applied to all the layers is ReLU. Figure 5.2 shows the architecture.

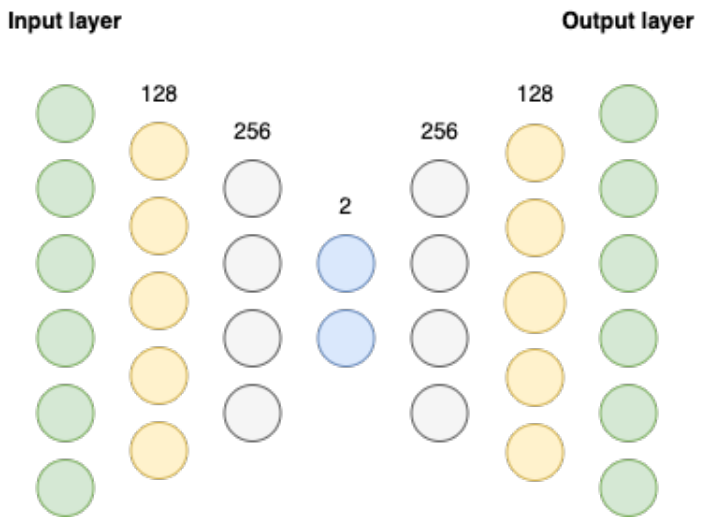

Figure 5.2: AE network architecture.

Regarding the training process, the number of epochs used to train the network was 100. The optimizer applied to the AE was Adam, with a learning rate of 0.001. Each batch size was 64. The MSE loss function was used to measure the error.

As in the KPCA approach, the ML metrics were used as control. In fact, although the performance of the AE might be as expected with the ML metrics, the evaluation only became complete when the newly generated data was ingested into the DCA. If DCA's performance is good with that data, it is possible to confirm the usability of AE as a feature reduction tool in the context of DCA.

5.1.1.3 Autoencoder with KPCA

Inspired by [7], the approach that uses an AE with the kNN algorithm to reduce the number of features, this dissertation also aims to test a new variation of the AE. Instead of using a second algorithm as a classifier and, thus, evaluating the performance through kNN, the intention here is to use the AE to create a new representation of the data with the same size as the input data. Once the new data is generated, the KPCA is applied to reduce the features of the new representation.

As in the previously detailed approach, Pytorch Python ML library was also used. The configuration replicates the AEkNN by Pulgar [7] when it comes to the AE itself. The intention here was to propose and test a feature reduction for the DCA and incorporate it with the AE, as they have done with the AEkNN. As before, the configuration will be the same in all datasets.

For the architecture itself, symmetry is maintained, and each part is composed of three hidden layers. In the encoder part, the input layer is composed of the number of neurons set in *num_in* parameter and differs from dataset to dataset. The first hidden layer, connected to the previous layer, consists of 1.5 x *num_in* neurons. The second hidden layer consists of 0.5 x *num_in* neurons. The last hidden layer, the bottleneck layer, has 1.5 x *num_in* neurons. Concerning the decoder part, the process is repeated with the same number of neurons. As expected, the output layer has the same size as the input layer. The ReLU activation function is applied to induce linearity.

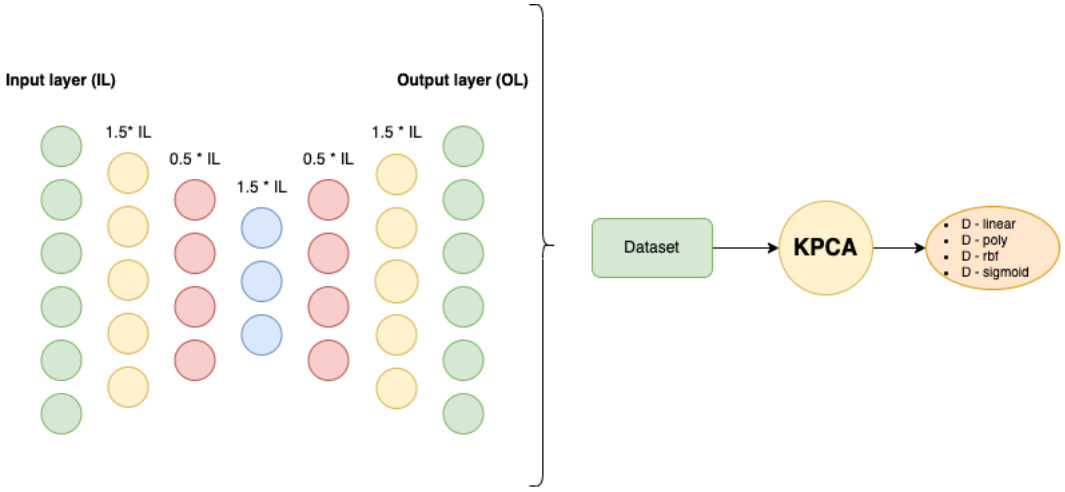

Figure 5.3: AEKPCA network architecture.

To keep the tests within the same context, the number of epochs was 100, as in the previous approach. The Adam optimizer was used once again with a learning rate of 0.001. The batch size and loss function were kept the same.

The new representation of the data, with 1.5 times the size of the input data, fed the KPCA to reduce the number of features to two. Python's library and the pipeline were the same, as Figure 5.3 shows. Each kernel was tested to generate the new compressed data for all the datasets.

This solution allowed us to mix both techniques and to do several comparisons between them. All the results will be detailed further.

5.2 Technological Framework

The experimental environment comprises all the hardware and software used during this dissertation. It is important to have a suitable and constant configuration to guarantee repeatability and reproducibility. Moreover, some changes had to be made to adapt to some issues, particularly concerning the hardware.

Regarding the software, Python 3.11 [124] was the programming language. Within Python, several libraries were considered, some of which have already been mentioned. Besides them, Pandas [125], Numpy [126] and Matplotlib [127] were also used. Jupyter notebooks [128] were used to develop ML-related solutions, whereas the DCA was developed in a Python file.

With respect to hardware, here some implications have arisen when computing the KPCA. Firstly, the machine used to compute all the models has 16 GB of RAM and is equipped with 2,2 GHz Quad-Core Intel Core i7 processor. Although these specifications were enough to compute the AEs, it struggled to compute more than 50% of data with the KPCA. To overcome this issue, DIGI2 Digital and Intelligent Industry Lab [129], from FEUP, provided an external server.

Once all the models were trained and the generated data gathered, the DCA was fed with the different types of datasets to classify anomalies. Each dataset classification resulted in different metrics that have been detailed in Section 4.3.

5.3 Dendritic Cell Algorithm

As part of the proposed solution, there is a new approach to the DCA itself that this dissertation intends to propose. After some intensive research and preliminary analysis, a new architecture emerged. Embedded with this approach are the feature reduction techniques previously mentioned. Three phases are usually involved in computing the DCA, namely pre-processing, detection, and analysis.

Regarding the pre-processing phase, the one that motivated this dissertation, there is a clear difference when it is compared to the DCA's version found in the literature. This difference is the aggregation of an automatic pre-processing phase, which aims to substitute human intervention and give autonomy to the algorithm.

This first phase of the DCA is performed before the data ingestion and can become exhaustive in certain aspects. This exhaustive process behind the pre-processing phase is due to the necessity of selecting the appropriate signals and categorising them. In practice, it is extracting the most interesting features from a given dataset and calculating the signals to increase DCA's performance. To avoid this, this dissertation proposes an addition to the DCA's architecture, combining ML with the DCA.

The second phase of the DCA involves two distinct steps, namely anomaly detection and anomaly attribution. In the detection phase that this dissertation considered, it was one-stepped. Anomaly attribution was not considered since each dataset only contained one antigen type. However, the most important step was considered, which allowed the DCA to detect anomalies.

Two output signals are considered after each iteration to perform this anomaly detection. As mentioned before, one data point is processed by the DCA, consisting of a *Ds* and a *Ss* and the outcome of this process is two output signals that will be impactful in detecting anomalies, namely K and Csm. The K signal is a measure that shows the bias towards abnormality or normality or how to make judgments. In contrast, the CSM signal represents the amount of information that a DC has digested or when to make decisions [31]. This value is the sum of both *Ds* and *Ss*. It will allow the DCA to determine whether a certain DC is mature.

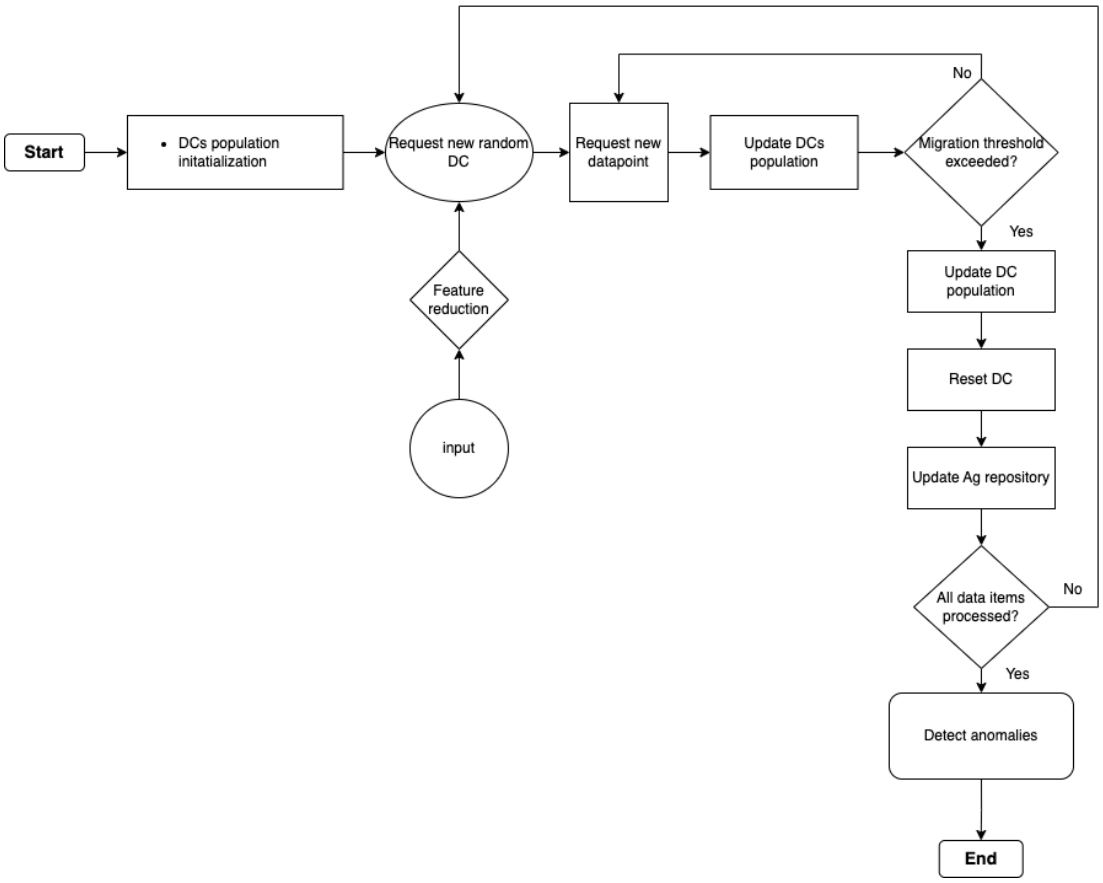

Figure 5.4: DCA architecture.

The last phase considered to build the DCA proposed by this dissertation is usually performed offline. The analysis phase starts when the algorithms finish to pre-process all the data. Over this last step, the output produced by each DC is aggregated to perform the referred analysis.

Once all the outputs are gathered, the value of K, which is the anomaly metric, is calculated. If the value of K is higher than the anomaly threshold, settled to 0, the DCA has found an anomaly. Otherwise, that specific antigen is not an anomaly. The value of k is the difference between the *Ds* and twice the *Ss*. After this process is over, the DCA ends. Figure 5.4 shows the architecture behind this approach. DCA's implementation is presented in Appendix A, whereas K and CSM implementations are presented in Appendix A.1 and Appendix A.2, respectively.

Chapter 6

Empirical Analysis of the DCA Pre-Processing Phase

In this chapter, which is composed of several sections, the experimental tests conducted to verify the viability of the proposed solutions are detailed. These tests provide a comprehensive analysis of applying the proposed feature reduction techniques to pre-process the data before feeding the DCA. By using different datasets that respect a wide domain spectrum, such as cyber security, biology, or finance, it is possible to evaluate the robustness and performance of these techniques. The first section relates to the design of the pipeline, whereas the following ones refer to the proposed techniques.

6.1 Experimental Design and Dataset Overview

The experiments were made employing datasets from several sources. As mentioned before, these datasets included a dataset that allowed some benchmark tests due to the correlation between some features and the predictive variable. Besides that, different datasets referring to different domains were tested, such as port scan, DDoS attacks, network traffic, credit card fraud, heart failure and stroke. Through this wide range of areas, it was possible to test the effectiveness and robustness of the DCA.

Table 6.1: Datasets' description.

Dataset	Instances	Features	Type
OPC UA [130]	107 634	32	Benchmark
Port Scan [131]	286 096	78	Cyber security
DDOS [131]	225 711	79	Cyber security
PCAP [131]	225 711	79	Cyber security
Credit Card Fraud [132]	190 911	79	Finance
Heart Failure [133]	918	12	Biology
Stroke [134]	5 110	12	Biology

Table 6.1 shows the different datasets considered for the purpose of this dissertation. It's important to mention that the shape of the datasets is quite heterogeneous, which allows for diversification in the analysis of results. Beyond the shape, the type and the context are different, granting us a wide range of tests regarding the applicability of the DCA. The following list provides a clear description of each dataset, its source and its usability.

- **OPC UA**: This benchmark dataset was provided by [130]. It includes 107634 instances and incorporates 32 features. It was used to develop the algorithm due to its correlation between variables and to provide a baseline comparison between all the tests.
- **Port Scan**: This is an intrusion detection dataset provided by [131]. The shape of the dataset comprises 286089 instances over 78 features. This dataset provides context from port scanning activities.
- **DDOS**: This dataset has the same creator as the one previously mentioned [131]. The shape is almost the same, with 225711 instances and 79 features. It gives context about DDOS attacks, another cyber security domain.
- **PCAP**: This dataset also created by [131]. The shape is 190911 instances and 79 features. It gives context about traffic that passes over a digital network, another cyber security domain.
- **Credit Card Fraud**: This dataset was collected during a project that aimed to improve the existing fraud detection [132]. These 284807 instances and 31 features were used to test DCA's performance in detecting fraudulent transactions in financial data.
- **Heart Failure**: This dataset was provided in [133]. This healthcare dataset consists of 918 instances and 12 features. It allows to predict heart failure in humans.
- **Stroke**: This dataset was provided in [134]. It is a biological dataset that comes up with 5110 instances and 12 features. It is used to predict whether someone is likely to get a stroke, testing DCA's performance to detect biological anomalies.

As stated, each dataset was submitted to several pre-processing steps before being ingested by the models used to reduce the number of features. Since they only accept numerical variables, it was necessary to encode categorical variables. Missing values were deleted, and the data was normalized using the Z-score. After that, the data was piped to the DCA to classify anomalies. Different metrics were considered to evaluate DCA's performance.

6.2 Baseline Performance

The baseline test was performed without any associated feature reduction. Instead, the feature *b_pktTotalCount* was selected because it had the highest correlation (≈ 1) with the predictive variable. Since we needed at least two features to feed the DCA (*Ds* and *Ss*), it was necessary to calculate both signals.

Different processes were considered to calculate both signals. Firstly, the mean value of *b_pktTotalCount* was calculated. After that, two different features were instantiated, *Danger* and *Safe*, both with 0.

For each value of *b_pktTotalCount* was subtracted the mean previously calculated. If the outcome of the subtracted value was lower than the mean, that value was assigned as an absolute value to the feature called *Danger*, and the feature *Safe* remained 0. If the outcome was higher than the mean, that value was assigned as an absolute value to the feature *Safe*, and the *Danger* feature remained 0. Therefore, the signals were calculated, and the respective label was mapped with *b_pktTotalCount* for later metrics calculations. Both signals were ingested by the DCA, and the anomalies were classified.

Table 6.2: Results of the baseline test. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	107 633	W/ FR	74 079	0.688	1.00	20.901

It is possible to confirm what was previously stated from Table 6.2. The correlation between the mentioned variable and the predictive variable is almost perfect, which means that one explains the other. This is represented by the accuracy shown in the table. Regarding the MCAV, the closer the 0, the better the algorithm performance. This value, 0.688, is subjective since it is treated the same way as values of 0 and -100. For that reason, it's important to include accuracy in the evaluation. The elapsed time is another important measure, and here, the benchmark is almost 21 seconds.

The baseline model is critically important when evaluating DCA's performance when applying the proposed feature reductions. Understanding this particular test allows us to assess the impact of those proposed solutions better to embed with the DCA.

6.3 Kernel PCA

The first technique to reduce the dimensionality of the data was the KPCA. The KPCA is a widely used technique that helps to capture non-linear relationships between features since it maps data into a higher-dimensional space. This approach aimed to reduce the number of features besides enhancing DCA's performance.

This technique was applied to several datasets using the different available kernels, such as Linear, Gaussian, Polynomial and Sigmoid. The model was trained to be useful in a wide range of domains, making it simpler than it should be. Each kernel was trained with different parameters to optimize the model's performance. The hyperparameters passed to the model are described in the following list:

- **Gamma**: Kernel coefficient for Gaussian, Polynomial and Sigmoid kernels - range (0, 1, 0.1)
- **Degree**: Degree for the polynomial kernel - [2, 3, 4, 5]
- **Coef0**: Independent term for the Polynomial and Sigmoid kernels - [2, 3, 4, 5]
- **Eigen Solver**: Eigenvalues solver to optimize the process - [‘auto’, ‘dense’, ‘arpack’, ‘randomized’]

After the hyperparameter tuning process, each kernel was used to generate a dataset with the best parameters found by the model. As mentioned before, the default values of the hyperparameters were maintained due to computational issues and to keep the models as simple as possible.

6.3.1 Linear Kernel

The Linear kernel is the simplest function implemented in kernel-based techniques such as KPCA. As a result of its lack of nonlinear mapping, the Linear kernel turns the KPCA almost as the normal PCA. This means it can be very advantageous when dealing with mostly linear data. Despite being used in a technique that aims to map the data into a higher-dimensional space, the Linear kernel does not act that way; it keeps the data in its original space.

This kernel has some advantages, such as simplicity, interpretability, efficiency and straightforward use. The first refers to the fact that it only involves dot product, which makes the process computationally less expensive. This is correlated with the associated efficiency since, for large datasets, it takes less time and memory usage than the other kernels. The results provided are from the original feature space, so it contributes to making the model more interpretable. Unlike other kernels, there is no need for hyperparameter tuning, so it is straightforward to use.

The fact that it can only capture linear relationships makes this kernel more prone to perform poorly if the underlying structure of the data is nonlinear. This is a clear disadvantage of this kernel; there is a trade-off between simplicity and usability. Another disadvantage linked to the previous one is that it can not deal with complex datasets, which can lead to sub-optimal performance.

Table 6.3: Results of the Linear kernel-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	64 580	Linear	20 963	0.001	0.431	6.400
Port Scan	62 941	Linear	5 694	0.026	0.453	4.722
DDOS	63 199	Linear	19 648	0.449	0.473	5.011
PCAP	64 910	Linear	4 857	0.028	**0.916**	4.728
Heart Failure	918	Linear	780	0.218	0.521	0.066
Stroke	5 110	Linear	2 998	0.384	0.422	1.295
Credit Card	62 658	Linear	7 405	0.506	**0.880**	5.815

Table 6.3 shows the metrics computed across different datasets during the application of the Linear kernel to pre-process the data before feeding the DCA. Since KPCA is computationally expensive, some adjustments had to be made, mainly in the dataset size. In fact, it was necessary to use samples from the original datasets, except for the biological datasets, namely heart failure and stroke, because their length is 918 and 5110 instances, respectively.

As far as time is concerned, it is clear that this version of the algorithm retains a property of lightness when processing the data provided by the Linear kernel. The data sets within the same dimension took approximately the same time. Due to their small size, the heart failure and stroke datasets showed a decrease in the time needed to compute the DCA.

On the other hand, the MCAV metric shows a variety in the number of fully mature antigens presented by the DCs. The same behaviour is observed in the anomaly counter metric. Regarding accuracy, the PCAP and credit card datasets show the highest values achieved, 0.916 and 0.880, respectively. This may suggest that these datasets have more linear relationships than the others. Detailed analysis and deeper insights will be taken in Chapter 8.

Compared to the baseline test, both results show a huge difference. The time consumed is within the expected range, so this is clearly a positive point. The same applies to MCAV, as you can see some improvements from the baseline to the linear kernel tests. As expected, accuracy dropped significantly, especially for the OPC UA datasets used in both tests.

6.3.2 Gaussian Kernel

One of the most widely used kernels in kernel-based techniques, such as KPCA, is the Gaussian kernel, sometimes called the Radial Basis Function (RBF) kernel. It has a great capacity for extracting intricate and non-linear relationships from data. Contrary to the previously mentioned kernel, this allows the mapping of the data into an infinite-dimensional feature space, which enables its capacity to model highly complex data and nonlinear relationships. This kernel has the *gamma* hyperparameter that determines the width of the Gaussian function.

The Gaussian kernel has three important advantages. The capacity to capture nonlinearity within the data can be applied to various applications. Its inherent flexibility is due to the *gamma* parameter, which can approximate the Gaussian kernel to other kernels. Lastly, the infinite-dimensional space provides robustness since it can capture various degrees of nonlinearity.

Since the computation of KPCA with this kernel is more complex, it can be more computationally expensive than Linear or Polynomial kernels, which is a disadvantage. The need for hyperparameter tuning and the risk of overfitting are also a disadvantage. The first can be very time-consuming because it often requires cross-validation. The second can occur due to the capture of noise within the data with large *gamma* values.

Table 6.4 presents the results of applying the DCA to the reduced data provided using the Gaussian kernel. The time necessary to run the DCA was almost the same as the previous test, which is a positive aspect.

The MCAV generally increased compared to the last test. This value reflects a decrease in the number of fully mature antigens presented by the DCA, which means that the algorithm could not

find the same amount of danger as in the previous test. Accuracy also points in this direction since all the datasets show lower values than the Linear kernel test. The overall number of anomalies found has been lower, with the same behaviour observed in the other metrics.

Table 6.4: Results of the Gaussian kernel-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	64 580	Gaussian	15 179	0.260	0.399	4.725
Port Scan	62 941	Gaussian	23 858	0.536	0.488	5.235
DDOS	63 199	Gaussian	17 558	0.547	0.470	5.665
PCAP	64 910	Gaussian	21 385	0.350	0.667	4.920
Heart Failure	918	Gaussian	744	0.192	0.536	0.145
Stroke	5 110	Gaussian	2 622	0.334	0.486	0.663
Credit Card	62 658	Gaussian	16 697	0.201	**0.733**	4.961

For the credit card dataset, the highest achieved accuracy, the value was 0.733 and may be indicative of the ability of the Gaussian kernel to find non-linear relationships.

Comparing the Gaussian-based results with the baseline test, where pre-processing was not automatic, the results were not good enough. We saw a decrease in important metrics such as accuracy and the anomaly counter. The latter saw the biggest decrease.

6.3.3 Polynomial Kernel

The Polynomial kernel used in this set of tests allows the modelling of different interactions between features up to a degree. In fact, the kernel maps the input data into a higher-dimensional space where polynomial relationships can be found. Since the mapping is implicit, the kernel is used to perform calculations. To train this model, it was necessary to optimize three different hyperparameters: *Degree*, *coef0* and *gamma*. The first determines the degree of the polynomial, the second controls the influence of higher-order terms and the last is the scaling factor applied to the dot product.

It presents two main advantages. Its capacity to capture polynomial interactions is effective and useful in certain domains. Besides, its flexibility by adjusting the *degree* and *coef0* parameters allows the model to learn more complex polynomial patterns and other higher-order interactions.

With flexibility comes two disadvantages, which are computationally demanding and the risk of overfitting, which are both related. For higher-degree polynomials and large datasets, this kernel could be computationally intensive and can capture noise within the can, which can lead to overfitting.

The data from applying the Polynomial kernel to the PCA was gathered and passed to the DCA, and the results are presented in Table 6.5. The time necessary to compute the algorithm saw some improvements, which means that the DCA took less time to compute the same datasets that were computed before. However, that's the only positive aspect of this particular test.

From the table, it is possible to confirm that this approach found a huge decrease in the number of anomalies classified, although the accuracy was almost the same as seen before. An increase in the number of false negatives explains this.

Table 6.5: Results of the Polynomial kernel-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	64 580	Polynomial	101	0.500	0.309	4.600
Port Scan	62 941	Polynomial	1	0.000	0.446	4.430
DDOS	63 199	Polynomial	49	0.000	0.433	4.979
PCAP	63 199	Polynomial	49	0.000	0.433	4.979
Heart Failure	918	Polynomial	721	0.247	0.544	0.086
Stroke	5 110	Polynomial	1 283	0.681	0.719	0.559
Credit Card	62 658	Polynomial	465	0.559	**0.991**	4.837

The credit card dataset showed the top performance after pre-processing through this kernel. The overall accuracy metric shows that this kernel found some relationships with the abovementioned problems within the data. MCAV metric decreased, particularly in the Port Scan, DDoS and PCAP datasets, which can be seen as a positive. A detailed analysis of the results will be presented next.

Compared with the baseline test, the results from the Polynomial-based data are similar to the ones mentioned previously, except for the anomaly counter metric. Here, we can see a huge drop in the number of anomalies found by the DCA, which may suggest that this kernel is not what we are looking for since the intention behind the algorithm is to find anomalies.

6.3.4 Sigmoid Kernel

The last set of KPCA tests was done with the Sigmoid kernel. This kernel, known as the hyperbolic tangent kernel, is inspired by the neural network theory and can be seen as an activation function. Applying the hyperbolic tangent transformation can capture nonlinearities by mapping the data into a higher-dimensional space similar to the neural network's behaviour. Two hyperparameters were optimized during this training: *alpha* and *coef0*. The first controls the scaling of the input data. The second is the bias term added to the dot product.

Since neural networks inspire it, it is a good choice for problems where neural networks are well suited. Moreover, it can capture nonlinearities like the ones mentioned before, which are two advantages.

In contrast to the advantages, there are three disadvantages. Improper optimisation of the parameters can lead to poor performance, as it is highly sensitive to the choice of parameters. There is also the risk of saturation for large values of the scalar product, which can lead to loss of information. Finally, since the kernel is not completely positive, this may not always guarantee a value kernel matrix.

Table 6.6: Results of the Sigmoid kernel-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	64 580	Sigmoid	21 754	0.175	0.438	6.487
Port Scan	62 941	Sigmoid	1 9201	0.552	0.480	6.035
DDOS	63 199	Sigmoid	7 324	0.153	0.448	4.842
PCAP	64 910	Sigmoid	17 678	0.346	**0.723**	5.373
Heart Failure	918	Sigmoid	708	0.196	0.510	0.069
Stroke	5110	Sigmoid	3 072	0.349	0.406	0.673
Credit Card	62 658	Sigmoid	18 859	0.353	0.698	4.756

All the datasets processed by the last kernel used in this first set of tests passed the DCA, just like the previous ones. The results from these computations are presented in Table 6.6.

Although this is not a real problem, as the impact is insignificant, it was the first time we found a generalised increase in the time needed to run DCA. Two sets of data took around 6 seconds, with values only observed in the Linear-based tests, where this technique was the simplest applied due to the kernel used.

Regarding the anomaly counter metric, through the pre-precessing with this kernel, the results were within the same pattern that we found before, except with the Polynomial kernel. The heart failure dataset registered the lowest value compared to the previous tests, a negative point.

As far as MCAV is concerned, the Sigmoid-based results are slightly better than all of the above, which indicates that a higher number of mature antigens were presented by the DCs. The results from the accuracy calculation demonstrate that the Sigmoid kernel had some meaningful impact in reducing the number of features since the general outcome was a little better than before. It is worth mentioning that the PCAP, heart failure and credit card datasets showed values above 0.5.

It is clear from comparing the Sigmoid-based results with the baseline test that the results were worse, as expected. The time necessary to run the algorithm was almost the same. The difference is found in the other two metrics, with one positive and two negative aspects. Regarding the positive aspect, the MCAV metric slightly improved compared to the baseline test. This is important because more mature antigens are present in the DCs. MCAV's behaviour was almost the same during this set of tests, always showing improvements when compared with the baseline. On the other hand, the accuracy and the anomaly counter metric significantly decreased their values, which is clearly a negative aspect of this test.

6.4 Autoencoder

The second approach used to reduce the number of features was the AE. AEs are neural networks that, among many other applications, can be used to reduce the dimensionality of data. It comprises two main parts: the encoder and the decoder. The first part compresses the input into the latent space, whereas the second part reconstructs the data from that compressed representation.

Although the two parts must be trained, only the encoder is used when necessary to create the new and lower dimensional data.

During this dissertation, the AE was trained on each dataset to learn a compact and meaningful representation of the data. The architecture was designed to be simple to suit the different dataset domains and balance between avoiding overfitting and losing meaningful information within the data. The intention behind the process was to keep all the approaches as similar as possible, so a few processes were repeated before training the model.

Firstly, all the datasets had categorical variables that needed to be transformed, so the transformations applied were identical. This allows us to establish comparisons throughout the study. Once the transformations were done, Z-score transformation was applied to the datasets to normalize the data before feeding the AE. After that, the data loader, provided by the Pytorch Python library, was created since it was necessary to pass the data as tensors to the model. These were the key steps in processing the data before entering into the model. The last step was training the model, with the architecture mentioned in Section 5.1.1.2, to create the new data representation.

This approach has several advantages, namely flexibility, feature learning, data denoising or dimensionality reduction. The first refers to the ability to suit different data types, whereas the second shows the capacity to learn useful features from raw data. The last two emphasise the power to remove noise from data and reduce the number of features, which is the purpose of this dissertation.

On the other hand, these techniques can be computationally expensive, especially for deep networks. It can also be very time-consuming due to its parameter sensitivity. It is important to keep in mind, during the training process, that AE can easily overfit, which is also another disadvantage.

A machine with a GPU was used to calculate these models, which allowed for greater computing resources and time savings. Once the data had been created and collected, the same machine was used to calculate the DCA as had been used previously in order to keep the tests as similar as possible. It is important to mention that due to the increase in computational power, during this test, the whole size of each dataset was used.

After feeding the DCA with the data provided by the AE, all the metrics were calculated as before. The results are presented in Table 6.7. The time that it took to run the algorithm increased, although it did not increase as expected, which is a clear positive point. The results from the AE-based approach may suggest that this technique improves the time necessary to compute the DCA. It is important for this study to keep the lightweight property of the DCA.

With regard to MCAV, it is clear that there has been an improvement compared to the values resulting from the KPCA-based tests. In general, AE performed better in this particular metric. The other two remaining metrics, anomaly counter and accuracy, also improved compared to the previous tests. Accuracy increased significantly in the PCAP and credit card datasets and also increased in the other datasets, although not as significantly. The anomaly counter is higher, too, but with that comes an increase in the number of false positives.

Table 6.7: Results of the AE-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	107 633	AE	24 731	0.397	0.400	12.633
Port Scan	286 096	AE	10 657	0.407	0.448	27.236
DDOS	225 711	AE	40 888	0.128	0.457	21.394
PCAP	190 911	AE	10 627	0.385	**0.935**	16.367
Heart Failure	918	AE	714	0.427	0.459	0.066
Stroke	5 110	AE	3 020	0.254	0.436	0.494
Credit Card	284 807	AE	22 508	0.284	0.920	27.012

Compared to the baseline model, the results are almost the same as the ones viewed before. The more notable improvement is with respect to the MCAV, where the results from this technique showed many mature antigens presented to the DCs. Otherwise, the accuracy is below of what is found in the baseline test, as expected. The same behaviour can be seen in the anomaly counter metric, where the number of false positives also increased.

6.5 AEKPCA

The last approach that was tested as a solution for the problem that motivated this dissertation is the AEKPCA. The AEKPCA is a combination of both techniques previously used and tested and was inspired in [7]. This combination uses the AE to create a new representation of the data, using its ability to capture non-linear relationships and the KPCA to reduce the number of features of the new data representation.

The architecture followed and implemented in AEKPCA was described in Section 5.1.1.3. Once the new data representation had been generated using AE, the KPCA was applied in the same way as on its own. This KPCA made the feature reduction in each dataset mentioned before.

As with the other approaches, there are advantages and disadvantages that must be considered. Combining both techniques can enhance the feature representation since both methods can be very effective in capturing linear and non-linear relationships. This technique also presents itself with the capacity to handle high-dimensional datasets, which is also an advantage. Conversely, this combination increases the computational time, requiring more resources than using either method alone. Both methods together also increase the complexity.

6.5.1 AEKPCA - Linear

As mentioned before, the test was subdivided by the kernels within the AEKPCA. The first kernel applied to data was the Linear kernel. Table 6.8 presents all the results from applying this method to various datasets. As the previous tests, this also includes the same metrics to allow comparisons.

Table 6.8: Results of the AEKPCA-Linear-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	53 816	AEKPCA - Linear	22 006	0.068	0.462	4.459
Port Scan	62 941	AEKPCA - Linear	8 758	0.405	0.460	5.399
DDOS	53 816	AEKPCA - Linear	16 834	0.067	0.429	4.257
PCAP	61 092	AEKPCA - Linear	8 120	0.311	**0.859**	4.800
Heart Failure	918	AEKPCA - Linear	737	0.261	0.431	0.076
Stroke	5 110	AEKPCA - Linear	3 094	0.383	0.427	0.614
Credit Card	59 809	AEKPCA - Linear	22 195	0.181	0.629	6.165

Since this method uses KPCA as the first one, there was a need to use samples of each dataset due to computational requirements, except for the heart failure and stroke datasets. This procedure will be the same during the different kernels applied.

With regard to the time metric, there was a slight improvement compared to the KPCA results without AE incorporated. However, this improvement had no impact on the process. The number of anomalies found increased in almost all datasets except for the PCAP dataset. Even so, this did not translate into increased accuracy, meaning that more false positives were found. The MCAV showed the worst results, meaning that less mature antigens were present in DCs.

Compared to the baseline test, the results are much worse, as you would expect. The only positive note goes to the MCAV, which improved its results compared to the baseline test. The time necessary to run the algorithm was very similar to the one seen before.

6.5.2 AEKPCA - Gaussian

The Gaussian kernel was also applied during this set of tests, and this was also a combination of both previous techniques. The results of this test are exhibited in Table 6.9. It is clear from the table that there was a small improvement in the accuracy compared to the previous subtest. Since the number of anomalies found also decreases, as it is possible to confirm from the anomaly counter metric, this might suggest that the number of false positives also decreases with the increase in accuracy, which is clearly interesting.

The MCAV metric is lower than the ones registered before, so the number of mature antigens presented by the DCs is higher in this case. The time consumed to run the algorithm is bigger than the time necessary to run with the data from the AEKPCA-Linear. However, this time does not have an effect on the computations.

Compared to the baseline test, two metrics have improved, while the other two have decreased. The positive metrics are the MCAV and the time necessary to run the algorithm, which are both lower than the registered in the baseline test. In the MCAV's case, lower values mean more mature antigens presented by the DCs, which is positive as a decrease in the time consumed. On the other hand, the accuracy and the number of anomalies found also decreased, which means that the algorithm did not perform as in the baseline test.

Table 6.9: Results of the AEKPCA-Gaussian-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	53 816	AEKPCA - Gaussian	18 264	0.141	0.436	5.863
Port Scan	62 941	AEKPCA - Gaussian	16 435	0.162	0.476	4.968
DDOS	53 816	AEKPCA - Gaussian	15 699	0.142	0.421	4.242
PCAP	61 092	AEKPCA - Gaussian	16 065	0.405	**0.732**	6.482
Heart Failure	918	AEKPCA - Gaussian	773	0.202	0.464	0.104
Stroke	5 110	AEKPCA - Gaussian	2 408	0.000	0.523	0.768
Credit Card	59 809	AEKPCA - Gaussian	18 204	0.307	0.695	4.865

6.5.3 AEKPCA - Polynomial

A certain amount of feature interactions is captured by AEKPCA, since the polynomial relationships in the AE-transformed data were modelled using the polynomial kernel. This behaviour was also observed in the Polynomial-based KPCA without the data transformed by the AE.

This kernel makes it possible to combine the non-linear transformation of AE with the extraction of polynomial characteristics from KPCA, which is useful for data sets in which polynomial relationships predominate.

The results of this test are shown in the Table 6.10. This may be the test where the technique didn't work at all. As you can see from the table, the PCAP and Credit Card datasets recorded accuracy values of 0.990 and 0.865, respectively. This may seem like an interesting value, although the number of anomalies found in the PCAP dataset was 0, and a large decrease was also observed in the Credit Card dataset. The result for the MCAV metric was also zero for the PCAP dataset, along with the OPC UA dataset.

In general, the number of anomalies found was way lower than in the previously mentioned tests. From all the datasets, the one that seems to perform better under this technique is the Credit Card. All the metrics improved in comparison to the tests mentioned before. The time necessary to run the model was almost the same as before, meaning no complexity was introduced in the data during the pre-processing phase.

Table 6.10: Results of the AEKPCA-Polynomial-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	53 816	AEKPCA - Polynomial	0	0.000	0.309	4.209
Port Scan	62 941	AEKPCA - Polynomial	101	0.200	0.446	4.601
DDOS	53 816	AEKPCA - Polynomial	101	0.500	0.309	4.403
PCAP	61 092	AEKPCA - Polynomial	0	0.000	**0.990**	4.933
Heart Failure	918	AEKPCA - Polynomial	667	0.501	0.471	0.068
Stroke	5 110	AEKPCA - Polynomial	2 473	0.270	0.517	0.633
Credit Card	59 809	AEKPCA - Polynomial	7 980	0.168	**0.865**	5.038

In comparison to the baseline test, the results are considerably worse. With the data from the

AEKPCA-Sigmoid approach, the DCA could not find anomalies, which is a very strange result. The accuracy also decreased, and it is a consequence of the previous metric. Conversely, the MCAV metric improved, although, with more mature antigens presented to the DC, the DCA still did not perform with this data. The time necessary to run the algorithm improved.

6.5.4 AEKPCA - Sigmoid

The hyperbolic tangent kernel was applied within the AEKPCA model as the last subset of tests. As it was mentioned above, this kernel was inspired by the neural networks, so in this specific test, there is an alignment between the first and the second phase of this pre-processing technique. Combining AE with sigmoid KPCA is hoped to utilise both the non-linear mapping of the sigmoid kernel and the ability of AE to extract non-linear features.

Table 6.11: Results of the AEKPCA-Sigmoid-based tests. DL: Dataset length; FR: Feature reduction technique; AC: Anomaly counter.

Dataset	DL	FR	AC	MCAV	Accuracy	Elapsed Time
OPC UA	53 816	AEKPCA - Sigmoid	15 468	0.001	0.419	4.510
Port Scan	62 941	AEKPCA - Sigmoid	22 651	0.165	0.485	5.244
DDOS	53 816	AEKPCA - Sigmoid	9 601	0.000	0.377	4.265
PCAP	61 092	AEKPCA - Sigmoid	20 108	0.361	**0.667**	5.489
Heart Failure	918	AEKPCA - Sigmoid	760	0.247	0.459	0.119
Stroke	5 110	AEKPCA - Sigmoid	3 692	0.334	0.343	0.738
Credit Card	59 809	AEKPCA - Sigmoid	23 917	0.143	0.599	4.706

The results of the AEKPCA-Sigmoid-based tests are presented in Table 6.11. In comparison to the previous Sigmoid-based test, this one clearly underperformed. The results from the MCAV are better, and this behaviour was observed during the whole process, although the other metrics are slightly below what was expected. Even the time underperformed in this test.

In comparison to the baseline test, this test only surpasses it in the MCAV metric, which states that the DCA could present more mature antigens than in the baseline test. The other metrics clearly were not good enough.

6.6 Summary

The feature reduction techniques suggested to improve the DCA in identifying anomalies in various datasets were analysed and empirically validated in this chapter. Several datasets from different domains, including biology, finance and cybersecurity, were used to evaluate the resilience and effectiveness of KPCA, AE and AEKPCA. The aim of these methods was to reduce the dimensionality of the data while preserving or improving the detection powers of DCA.

As a basis for comparison, the baseline mode achieved 100% accuracy using a single strongly correlated feature without any feature reduction. When evaluated with Linear, Gaussian, Polynomial and Sigmoid kernels, the KPCA performed differently on different datasets. For the PCAP and credit card datasets, the Linear kernel outperformed the Gaussian kernel in terms of accuracy, but at a slightly higher computational cost. The results of the Sigmoid and Polynomial kernels were inconsistent, with the Polynomial kernel outperforming the others for the credit card dataset.

With the best results for the PCAP and credit ard datasets, the AEs showed greater computational efficiency and detection performance. When compared to KPCA, the AE method demonstrated a notable decrease in computation time and an improvement in accuracy. The aim of combining AE and KPCA was to maximise the benefits of each method. The results of AEKPCA using different kernels varied; for the PCAP dataset, the Linear and Gaussian kernels produced the best results, while the Polynomial and Sigmoid kernels produced inconsistent results.

Chapter 7

Validation and Performance Analysis of DCA

This chapter gives an extensive analysis of the validation tests carried out to reinforce and verify the results previously obtained. Based on these findings, this chapter aims to evaluate DCA's performance and robustness in detecting anomalies in data pre-processed through the proposed solutions. This chapter deals with the design of the validation test and the results per data set.

7.1 Validation Test Design and Methodology

To validate the results previously described, this dissertation proposes three sets of tests that will allow deeper analysis and further conclusions. Thus, there are a few changes in this group of tests compared to the ones mentioned before.

Firstly, the intention to keep a wide range of domains within the tests was also considered. In this particular test, three datasets were transited from the first group of tests: OPC UA, port scan, and DDoS. These datasets all belong to the same category, which is network analysis. The chosen datasets differ from those used before in the other categories, Biology and Finance. With respect to Biology, this refers to diabetes, whereas another credit card dataset has been chosen to represent the Finance domain.

Once the datasets were found, some parameters were set before performing the tests. Since we had computational issues running the KPCA before, some improvements had to be made to optimize the tests and keep them similar during the process. These improvements refer to the samples that needed to be taken from the original datasets, which resulted in datasets with 60000 instances.

Moreover, three validation scenarios were created to have different comparisons. The first validation scenario included a sample with 60000 instances, as mentioned, of each dataset. Each sample contained the same proportion of positive and negative labels as the original dataset. In the second validation scenario, once again, each dataset was sampled with 60000 instances, although,

this time, each had half positive and half negative labels. In the last validation scenario, the proportions of the first scenario were inverted, and the 60000 instances were maintained.

The last step of the project was similar to the previous test. Since the new data sets did not have categorical variables, there was no need for label coding. The same label coding was applied to the other data sets. Once the necessary transformations had been made, the same normalisation technique was applied, and each dataset was fed the same three feature reduction techniques, namely KPCA, AE, and AEKPCA.

Table 7.1: Validation datasets.

			Validation 1		**Validation 2**		**Validation 3**	
Dataset	**Instances**	**Features**	1	0	1	0	1	0
OPC UA [130]	53 816	32	0.69	0.31	0.5	0.5	0.31	0.69
Port Scan [131]	62 941	78	0.55	0.45	0.5	0.5	0.45	0.55
DDOS [131]	53 816	79	0.57	0.43	0.5	0.5	0.43	0.57
Diabetes [135]	70 692	18	0.55	0.45	0.5	0.5	0.45	0.55
Credit Card [136]	1 000 000	8	0.09	0.91	0.5	0.5	0.91	0.09

Table 7.1 shows each validation scenario's different datasets and label proportions. It is important to mention that, once again, there the shape of the datasets is quite heterogeneous, which allows different types of results and conclusions. The following list provides a clear description of each dataset, its source and its usability.

- **OPC UA**: This benchmark dataset was provided by [130]. It includes 107634 instances and incorporates 32 features. It was used to develop the algorithm due to its correlation between variables and to provide a baseline comparison between all the tests.
- **Port Scan**: This is an intrusion detection dataset provided by [131]. The shape of the dataset comprises 286089 instances over 78 features. This dataset provides context from port scanning activities.
- **DDOS**: This dataset has the same creator as the one previously mentioned [131]. The shape is almost the same, with 225711 instances and 79 features. It gives context about DDOS attacks, another cyber security domain.
- **Diabetes**: This dataset has diabetes indicators and allows us to predict whether there is a disease [135]. These 70692 instances and 18 features were used in the validation scenarios.
- **Credit Card**: This dataset was provided in [136]. This is another finance dataset and consists of 1000000 instances and 8 features. It allows us to predict credit card fraud.

7.2 Performance on Network Analysis Datasets

As mentioned before, this section presents the results of the validation tests and the DCA performance with the integrated feature techniques. Each subsection corresponds to a dataset, where a detailed analysis is made. The results are presented with respect to the three feature reduction approaches proposed across three validation scenarios. The same metrics as before are considered, which include accuracy, MCAV, anomaly counter and elapsed time.

The OPC UA dataset was sampled according to the conditions stated before, which resulted in three datasets with 60000 instances and 32 features. Each sampled dataset was used to feed the models that were trained to produce a lower-dimensional and meaningful new representation of the data. The results for the three validation scenarios are shown below.

7.2.1 OPC UA - Original Proportions

In the first validation scenario, the sampled dataset keeps the original proportion of positive and negative labels. Its unique difference is the number of instances.

Table 7.2: OPC UA first validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	22 684	0.001	0.455	4.851
KPCA-Gaussian	24 250	0.253	**0.464**	4.816
KPCA-Polynomial	217	0.682	0.311	4.320
KPCA-Sigmoid	18 254	0.000	0.425	4.716
AE	18 358	1.000	0.428	4.519
AEKPCA-Linear	15 783	0.271	0.408	4.500
AEKPCA-Gaussian	22 164	0.319	0.447	4.474
AEKPCA-Polynomial	0	0.000	0.310	4.221
AEKPCA-Sigmoid	20 699	0.031	0.445	4.985

From Table 7.2, it is possible to confirm the different results obtained from the ingestion of the datasets by the DCA. In this set of tests, the KPCA-Gaussian-based approach achieved the highest accuracy of 0.464. This technique also presented a value of MCAV close to 0, meaning that the DCs presented a high number of mature antigens.

The AE presented the worst results regarding MCAV, which was not expected due to the reasonable accuracy found. The AEKPCA-Polynomial-based technique did not find any anomaly, so it clearly underperformed.

Regarding the time, the results were quite similar across the different approaches, so there are no important insights to mention. Compared to the previously mentioned results, the time is also similar.

Overall, the results from this scenario show that the KPCA with the Gaussian kernel was more effective than the other techniques to approach this dataset. This may suggest that there are non-linear relationships that this kernel was able to capture. The combination of both techniques

underperformed in comparison to the other techniques, as the lowest results were found in the AEKPCA-Polynomial-based approach.

7.2.2 OPC UA - Balanced Proportions

The sampled dataset differed from the previous one in this second validation scenario. Here, instead of having the original proportion of positive and negative labels, it was changed to have both levels with the same proportion. The number of instances was 60000.

Table 7.3: OPC UA second validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	18 978	0.006	0.496	4.809
KPCA-Gaussian	17 123	0.073	0.498	4.592
KPCA-Polynomial	0	0.000	0.500	4.198
KPCA-Sigmoid	20 492	0.006	**0.504**	4.630
AE	16 044	0.587	0.499	4.720
AEKPCA-Linear	20 176	0.067	0.495	4.530
AEKPCA-Gaussian	16 965	0.071	0.500	4.418
AEKPCA-Polynomial	0	0.000	0.500	4.218
AEKPCA-Sigmoid	16 761	0.070	0.500	5.266

Regarding the KPCA-based approach, it is perceptible from Table 7.3 that the accuracy metric increased compared to the previous scenario. The highest accuracy value was again found in the KPCA-based technique, particularly in the KPCA-Sigmoid-based approach, with 0.504. The total number of anomalies found is similar to the ones seen before, except for the KPCA-Polynomial-based, since the DCA did not find any anomalies. Instead, it classified all the data points as non-anomaly. The values of MCAV are quite good, and it shows that a higher number of antigens were presented to the DCs.

The other two approaches, AE and AEKPCA, had behaviours similar to those of the first. With respect to the AE, the accuracy achieved was not the highest in the scenario, although the number of false positives was. It also recorded the highest value of MCAV, with 0.587, which is clearly an outlier within this scenario. In the AEKPCA tests, three tests recorded an accuracy of 0.500. However, the AEKPCA-Polynomial-based approach did not find any anomaly, which was the purpose behind the algorithm.

Concerning the time necessary to run the DCA, the results were similar to those recorded. The AEKPCA-Sigmoid-based test recorded 2.266 seconds, although this is not an increase that did not keep the lightweight property of the DCA.

Overall, this scenario recorded better results than the previous scenario, and it is clear an improvement in the direction that this dissertation aims to address.

7.2.3 OPC UA - Inverted Proportions

The last validation scenario concerning the OPC UA dataset differed from the previously mentioned ones. In this set of tests, the sampled dataset had the original proportions of positive and negative labels inverted. The number of rows was maintained similarly.

From Table 7.4, it is clear that there was a general improvement in the DCA's accuracy. The time necessary to run the algorithm also improved, whereas the MCAV are similar. The negative point of this scenario is the decreased number of anomalies recorded.

Table 7.4: OPC UA third validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	19 256	0.011	0.536	3.975
KPCA-Gaussian	5 226	0.126	0.614	3.749
KPCA-Polynomial	101	0.500	**0.643**	4.513
KPCA-Sigmoid	20 274	0.005	0.529	4.304
AE	5 540	0.036	0.613	4.444
AEKPCA-Linear	16 371	0.069	0.552	3.882
AEKPCA-Gaussian	21 317	0.271	0.526	4.111
AEKPCA-Polynomial	0	0.000	**0.643**	3.801
AEKPCA-Sigmoid	18 638	0.068	0.538	3.944

Regarding the KPCA-based approach, some results are important to mention. Firstly, one of the highest accuracy values recorded belongs to this approach. However, this value results in fewer anomalies detected, leading to an increase in false negatives. The Linear and the Sigmoid-based approaches did not record the highest accuracy value. On the other hand, they found more anomalies, presented a better value of MCAV, and even took less time to run.

Similar to the KPCA-based results were the AE and AEKPCA results, where even the same issue can be identified. AE's results are above 0.600, which is clearly an improvement compared to the previous scenarios, but the number of anomalies found is not satisfactory. The AEKPCA-based approaches did not improve much in comparison to the KPCA-based approaches. The polynomial kernel presents the same issue, whereas the Gaussian performed better than the others.

Regarding the time necessary to run the DCA, some approaches generated data that took less than 4 seconds to run the algorithm. This is the lowest value found.

Overall, this scenario presents the best results of this dataset since it took less time to run and recorded better accuracy, and the MCAVs are similar. The negative point is the anomaly counter metric, which decreased considerably.

7.3 Performance on Cybersecurity Datasets

The Port Scan dataset was used again to validate the results previously obtained. The conditions were replicated to maintain the equality of the results. This dataset was sampled thrice, resulting in 60000 instances over 78 features.

7.3.1 Port Scan - Original Proportions

In the Port Scan's validation scenario 1, the original proportion of positive and negative labels was maintained, which provided a trustworthy representation of the dataset.

From Table 7.5, it is possible to confirm that the MCAV had some bad values regarding the number of antigens presented to the DCs. The table also shows the accuracy recorded did not improve much compared to the values seen before across all the tests. The time necessary to run the DCA increased generally; on the other hand, the anomaly counter metric decreased.

Table 7.5: Port Scan first validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	5 368	0.032	0.459	4.983
KPCA-Gaussian	14 461	0.530	0.471	4.460
KPCA-Polynomial	111	0.910	0.450	4.274
KPCA-Sigmoid	17 508	0.522	0.479	5.264
AE	4 751	0.331	0.457	4.580
AEKPCA-Linear	10 778	0.049	0.469	4.415
AEKPCA-Gaussian	11 941	0.355	0.470	5.166
AEKPCA-Polynomial	448	0.008	0.450	4.830
AEKPCA-Sigmoid	12 136	0.200	**0.472**	4.972

Regarding accuracy, the highest values were found in KPCA-Gaussian-based and AEKPCA-Gaussian-based. It is clear that the Gaussian kernel performed better than the other kernels. However, the DCA found more anomalies with data from the KPCA-Gaussian-based approach.

Concerning the anomaly counter metric, both Polynomial-based approach data underperform, which led to an increasing number of false negatives. The KPCA-Polynomial-based also recorded the highest value in MCAV, which is a negative aspect.

Overall, the tests from this scenario indicate that the Gaussian kernel may suit this type of data better than the other approaches.

7.3.2 Port Scan - Balanced Proportions

The second validation scenario stands for the same proportion for both labels. In this particular group of tests, the number of positive and negative labels is the same.

From Table 7.6, some results improved from the last scenario, especially in the accuracy metric. Almost all the results are very close to 0.500, which is clearly an improvement. The highest accuracy value was found in the KPCA-Gaussian-based approach, with 0.503. Regarding anomaly counter metric, the lowest value was found again in the Polynomial-based techniques, with 102 and 47, respectively. The AEkPCA-Gaussian-based approach took more than 2 seconds than the other tests for the same dataset shape. The anomaly counter metric had the same behaviour, which clearly states that the DCA found fewer anomalies than it should.

Overall, the results of this scenario are similar to the ones registered in the previous scenario, so there were no improvements made, and the results are not good enough.

Table 7.6: Port Scan second validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	3 606	0.020	0.499	4.318
KPCA-Gaussian	12 276	0.543	**0.503**	4.510
KPCA-Polynomial	102	0.502	0.500	4.737
KPCA-Sigmoid	11 068	0.563	0.497	4.594
AE	11 953	0.452	0.499	4.600
AEKPCA-Linear	13 139	0.030	0.500	4.760
AEKPCA-Gaussian	16 667	0.312	0.500	6.633
AEKPCA-Polynomial	47	0.012	0.500	4.220
AEKPCA-Sigmoid	16 778	0.208	0.497	4.613

7.3.3 Port Scan - Inverted Proportions

In the last scenario of this dataset, the conditions pre-defined for scenario 3 were considered. In Table 7.7, there is a common issue for both three scenarios: the Polynomial-based approaches. Some improvements were found in the anomaly counter metric. Although, the KPCA-Polynomial and the AEKPCA-Polynomial techniques recorded the lowest values, with 8 and 214, respectively. This relates to the number of false positives that automatically increased.

Table 7.7: Port Scan third validation scenario.

Feature Reduction	Anomaly Counter	MACV	Accuracy	Elapsed Time
KPCA-Linear	7 288	0.032	0.463	4.606
KPCA-Gaussian	23 944	0.529	0.489	4.637
KPCA-Polynomial	8	0.018	0.450	4.703
KPCA-Sigmoid	17 295	0.525	0.481	4.613
AE	10 259	0.399	0.467	5.113
AEKPCA-Linear	9 169	0.016	0.461	4.476
AEKPCA-Gaussian	23 755	0.342	**0.490**	4.530
AEKPCA-Polynomial	214	0.273	0.450	4.202
AEKPCA-Sigmoid	18 128	0.224	0.478	4.520

Concerning accuracy, the results are similar to scenario 1 and lower than scenario 2. The AEKPCA-Gaussian-based approach recorded the highest accuracy value, with 0.490, whereas both Polynomial-based approaches recorded the lowest values, with 0.450. In most kernels, the results were worse.

Regarding the time consumed to run the DCA, almost all the results are within 4 and 5 seconds, which is the pattern in all these tests. The Gaussian-based approaches also overperformed in the anomaly counter metrics, with the highest values again. The result of the AE-based approach is not good enough.

Overall, this scenario did not benefit the data. The same behaviour was observed in the first scenario. From the three sets of tests, the second scenario, where the proportions between both

labels were the same, had the best results in the validation tests. The Gaussian-based approaches, presented better results than the others.

7.3.4 DDoS - Original Proportions

The last dataset from the Cybersecurity domain considered was the DDoS. This dataset was sampled into three different datasets, the same way as before. In the first validation scenarios, the results do not differ greatly from those we have seen. The time necessary to run the DCA is still between the same interval. The highest value of MCAV was recorded by the KPCA-Gaussian based, with 0.539, which already happened before. In terms of accuracy, there are no values above 0.500, which is a negative aspect. The number of anomalies found is quite low in some tests.

Table 7.8: DDOS first validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	23 975	0.443	**0.484**	4.607
KPCA-Gaussian	12 759	0.539	0.460	4.587
KPCA-Polynomial	0	0.000	0.430	4.230
KPCA-Sigmoid	15 941	0.180	0.466	4.506
AE	13 444	0.141	0.460	4.450
AEKPCA-Linear	4 232	0.283	0.440	4.563
AEKPCA-Gaussian	7 699	0.132	0.449	5.047
AEKPCA-Polynomial	59	0.338	0.430	4.290
AEKPCA-Sigmoid	20 255	0.295	0.477	4.953

Regarding the accuracy, there are no values below 0.430. The highest value recorded was 0.484, using the KPCA-Linear-based technique, which is the simplest kernel applied. However, the difference between the worst and the best results is very low. The KPCA-Linear-based approach was also the one that found the most anomalies, which is a positive result with the highest accuracy achieved. Table 7.8 shows the results.

Once again, the Polynomial-based tests were not very effective in finding anomalies. The KPCA-based approach recorded 0 anomalies, whereas the AEKPCA-based approach classified 59 data points as anomalies.

Overall, the KPCA-Gaussian-based approach showed better performance regarding anomalies found and their respective accuracy, whereas both Polynomial-based approaches did not perform as expected. It is important to mention that neither of the techniques surpassed the 0.500 in terms of accuracy, indicating challenges that must be addressed in this dataset.

7.3.5 DDoS - Balanced Proportions

The second validation scenario demonstrated an improvement in terms of MCAV and accuracy. Regarding the MCAV, all the values were below 0.500, whereas concerning the accuracy, all the values were greater or equal to 0.500, except one. The time necessary to run the DCA is within the same pattern, with all the values below 5 seconds.

From Table 7.9, it is possible to confirm the improvement found in the accuracy values. The KPCA-Gaussian-based approach overperformed again when compared to the other approaches. The value of accuracy recorded was 0.501. Only the AEKPCA-Sigmoid-based approach recorded a value below 0.500, although the difference to the highest value recorded is insignificant.

Table 7.9: DDOS second validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	13 155	0.412	0.501	4.433
KPCA-Gaussian	20 080	0.142	**0.504**	4.614
KPCA-Polynomial	91	0.231	0.500	4.668
KPCA-Sigmoid	14 203	0.192	0.502	4.554
AE	5 357	0.345	0.501	4.264
AEKPCA-Linear	9 266	0.215	0.500	4.390
AEKPCA-Gaussian	16 094	0.392	0.502	4.469
AEKPCA-Polynomial	47	0.018	0.500	4.320
AEKPCA-Sigmoid	8 809	0.194	0.498	4.498

Regarding the number of anomalies found, the results are below expectations. Since these datasets have 30000 anomalies, due to the proportions of labels chosen, only the KPCA-Gaussian-based approach found two-thirds, although some of them are false positives. The Polynomial-based approaches struggled again with this type of data.

Overall, the results are slightly better than in the previous scenario in terms of accuracy. This time, the KPCA-Gaussian-based technique outperformed the others, which differs from the previous test. This was the best technique for this scenario.

7.3.6 DDoS - Inverted Proportions

In the last validation scenario of this domain, the original proportions of the labels were inverted.

In Table 7.10, presented above, it is demonstrated that some metrics improved and are better than the ones recorded before. As in other datasets, when the proportions were inverted, the results improved. In terms of accuracy, all the values are higher than 0.500, except the one from AE-based data, whereas the number of anomalies found is slightly higher, which gives trustworthy results. The time necessary to run the DCA is almost the same, and the MCAV pattern keeps the same way.

Concerning the different techniques, the KPCA-Polynomial-based recorded the highest accuracy value, with 0.570. Despite that, the number of anomalies found is 4, which does not meet the DCA purposes. On the other hand, the AE-based approach presented the lowest accuracy values, with 0.474.

Overall, the Gaussian-based approaches performed better than the others. The KPCA-Gaussian-based approach found lower values of MCAV and acceptable values of anomalies found and accuracy.

Table 7.10: DDOS third validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	15 390	0.389	0.533	4.788
KPCA-Gaussian	22 715	0.058	0.519	4.572
KPCA-Polynomial	4	0.000	0.570	4.979
KPCA-Sigmoid	23 903	0.569	0.515	5.983
AE	18 479	0.267	0.474	5.307
AEKPCA-Linear	11 155	0.299	0.544	4.801
AEKPCA-Gaussian	19 394	0.428	0.522	4.438
AEKPCA-Polynomial	436	0.441	**0.569**	4.279
AEKPCA-Sigmoid	19 170	0.252	0.530	4.534

7.4 Performance on Biological Datasets

As it was already mentioned, testing the DCA's range of applicability is also an intention of this study. A dataset from another domain, like Biology, was chosen to effectively serve that purpose. To this dataset, the same sampling conditions were applied to have similar tests.

7.4.1 Diabetes - Original Proportions

In the first validation scenario, there is a difference from the previous group of tests, which is the number of anomalies found. In this scenario, the results are quite homogeneous, with no outliers as in the previous datasets. Regarding accuracy, the pattern seen before is still present, whereas the MCAV follows the same behaviour. The running time is between the same interval.

Table 7.11: Diabetes first validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	12 115	0.169	0.470	4.713
KPCA-Gaussian	26 751	0.366	**0.496**	4.697
KPCA-Polynomial	14 205	0.582	0.473	4.908
KPCA-Sigmoid	21 398	0.191	0.490	4.600
AE	15 167	0.770	0.477	4.427
AEKPCA-Linear	12 204	0.110	0.468	5.142
AEKPCA-Gaussian	20 137	0.129	0.482	4.430
AEKPCA-Polynomial	12 971	0.042	0.472	4.859
AEKPCA-Sigmoid	18 150	0.114	0.479	4.537

From Table 7.11, is possible to confirm the results. The results of the KPCA-based approaches are slightly better than those of the other approaches. The highest number of anomalies was found, with 26751, and the highest accuracy was achieved, with 0.496. These values belong to the Gaussian-based approach.

Regarding the other two approaches, the DCA also performed better than in other tests. AE's results showed improvements in accuracy and anomalies found compared to the previous datasets.

The AEKPCA-based techniques showed the lowest values of MCAV and the same pattern regarding the number of anomalies and accuracy.

Overall, there is an improvement in this type of data when compared to the Cybersecurity type. The KPCA-Gaussian-based outperformed the others.

7.4.2 Diabetes - Balanced Proportions

Concerning this validation scenario, the conditions were changed from the previous test to have the same proportions of both labels. This adjustment provided a balanced dataset to evaluate the proposed feature reduction techniques under these circumstances.

Table 7.12: Diabetes second validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	18 476	0.168	0.497	4.502
KPCA-Gaussian	12 125	0.361	0.503	4.595
KPCA-Polynomial	9 172	0.513	**0.504**	4.378
KPCA-Sigmoid	17 853	0.201	0.498	5.283
AE	7 204	0.283	0.500	4.541
AEKPCA-Linear	19 111	0.166	0.498	4.540
AEKPCA-Gaussian	16 929	0.213	0.497	5.165
AEKPCA-Polynomial	16 586	0.229	0.499	4.823
AEKPCA-Sigmoid	19 734	0.165	0.503	4.757

The results from this test are quite different than in the previous test, as it is presented in Table 7.12. The number of anomalies found decreased in general, whereas the accuracy had some improvements. The time necessary to run the DCA keeps the same pattern. The same behaviour is found in the MCAV metric.

Regarding the KPCA-based approaches, the Linear-based approach outperformed the others with an accuracy value of 0.497 and 18476 anomalies found. It is not the best in terms of accuracy, but it is the one that has the highest value of accuracy and anomalies found together.

Regarding the other two feature reduction techniques applied, the results are similar. The highest accuracy value can be found in the AEKPCA-Sigmoid-based approach, with 0.503. The AE did not perform as expected since it found the least anomalies within this test.

Overall, the AEKPCA-Sigmoid-based approach outperformed the other techniques. This approach found the highest number of anomalies and recorded the second-highest accuracy value.

7.4.3 Diabetes - Inverted Proportions

The last validation scenario concerning the Biology data type is presented below. In this validation scenario, the proportions were inverted, as in all validation scenarios 3. In this set of tests, there are some improvements compared to the two previously mentioned scenarios. The time taken to perform the DCA and MCAV were within the standard.

In the Table 7.13, presented below, is stated the improvement seen in the accuracy, whereas some tests presented fewer anomalies found. In terms of accuracy, the AEKPCA-Polynomial-based approach recorded the highest accuracy, with 0.539. This value is also the highest using the diabetes dataset. The AE-based approach presented the lowest value in this set of tests, which was a pattern in AE's results during diabetes validation scenarios.

Table 7.13: Diabetes third validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	11 634	0.167	0.532	4.482
KPCA-Gaussian	17 026	0.368	0.519	4.584
KPCA-Polynomial	7 064	0.140	0.537	4.357
KPCA-Sigmoid	17 722	0.203	0.516	4.526
AE	22 368	0.365	0.484	4.570
AEKPCA-Linear	19 049	0.152	0.519	4.797
AEKPCA-Gaussian	15 372	0.211	0.525	4.825
AEKPCA-Polynomial	8 301	0.047	**0.539**	4.766
AEKPCA-Sigmoid	18 707	0.099	0.520	4.737

Regarding the number of anomalies found, the AE presented the best result in this set of tests, with 22368. This might suggest that the lowest accuracy value is not as bad as it might seem. The lowest value of MCAV was also registered by the AEKPCA-Sigmoid, with 0.099.

Overall, this scenario seems to be the best configuration for this dataset compared to the other scenarios. This is sustained by the improvement that the results show. The AEKPCA-Polynomial-based technique outperformed the others.

7.5 Performance on Financial Datasets

Another credit card fraud dataset was the last dataset considered for the validation test. This dataset is different from the one used in the first tests, as the shape also confirms. DCA's robustness in another domain was tested with this dataset.

7.5.1 Credit Card - Original Proportions

In the first validation scenario, relating to the credit card dataset, the results were better than those recorded previously. None of the aforementioned scenarios recorded an accuracy of more than 0.600. This problem was overcome with this dataset. The highest number of anomalies found is similar to the highest numbers recorded before, and the time necessary to run the algorithm also increased. There were two values that did not meet the standard criteria.

Regarding the elapsed time metric, there are two values above 5 seconds. The first belongs to the KPCA-Linear-based approach, with 5.503 seconds, whereas the second belongs to the AEKPCA-Sigmoid-based approach, with 6.853 seconds. These values are higher than the ones recorded before, although this does not affect the algorithm's performance.

With respect to the MCAV metric, the values are higher than expected, especially in the approaches with higher accuracies. The highest value was recorded using the AEKPCA-Polynomial-based approach, with 0.704.

Table 7.14: Credit card first validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	21 389	0.549	0.619	5.503
KPCA-Gaussian	8 610	0.346	0.793	4.775
KPCA-Polynomial	144	0.044	**0.908**	4.386
KPCA-Sigmoid	14 801	0.612	0.708	4.933
AE	14 627	0.403	0.712	4.978
AEKPCA-Linear	15 623	0.396	0.696	4.580
AEKPCA-Gaussian	15 623	0.396	0.696	4.580
AEKPCA-Polynomial	240	0.704	0.907	4.396
AEKPCA-Sigmoid	18 010	0.371	0.665	6.853

Concerning the anomaly counter and the accuracy metrics, the results have some positive aspects. The KPCA-Linear-based approach classified 21389 data points as anomalies, the highest number in the set of tests. Moreover, the highest accuracy value was 0.908, recorded by the KPCA-Polynomial-based approach. This result would be a great result if the algorithm had found more anomalies than the 144 that it found.

Overall, the AE-based and the AEKPCA-Sigmoid-based approaches outperformed the other approaches. The first recorded an accuracy of 0.712 over 14627 anomalies, whereas the second recorded an accuracy of 0.665 over 18010 anomalies, as shown in Table 7.14.

7.5.2 Credit Card - Balanced Proportions

In the second validation scenario, the quality of the results decreased slightly. The time needed to run the DCA was within the standard. The MCAV and the accuracy generally decreased, whereas the anomaly counter metric showed some improvements.

Table 7.15: Credit card second validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	21 280	0.336	0.504	4.843
KPCA-Gaussian	24 049	0.548	0.502	4.715
KPCA-Polynomial	318	0.290	0.500	4.457
KPCA-Sigmoid	32 196	0.105	0.500	5.273
AE	17 289	0.236	**0.674**	4.692
AEKPCA-Linear	17 330	0.484	0.502	4.665
AEKPCA-Gaussian	13 896	0.180	0.502	5.192
AEKPCA-Polynomial	113	0.247	0.500	4.578
AEKPCA-Sigmoid	12 666	0.109	0.499	4.641

From Table 7.15, is possible to confirm that the AEKPCA-Sigmoid-based approach found the highest number of anomalies so far, although the accuracy recorded is below expected. In this metric, almost all the techniques improved their results, except for both Polynomial-based approaches.

Regarding accuracy, the highest value computed belongs to the AE-based approach, with 0.674, whereas the lowest belongs to the AEKPCA-Sigmoid-based approach. Generally, the accuracy values decreased compared to the previous scenario. Tha MCAV pattern is better than the one mentioned before, which is a positive aspect.

Overall, the AE-based approach outperformed the other approaches with 17289 anomalies and an accuracy value of 0.674. The results in the first scenario were better than these.

7.5.3 Credit Card - Inverted Proportions

The last group of tests made to validate the results obtained before had some interesting results. In general, the accuracy measured is the highest of the three tests. There are two outliers, although these tests had an increase of false negatives. Regarding the time necessary to run the DCA, the results are again within the same pattern. The MCAV metric is similar to the one recorded in the first scenario, whereas the anomaly counter metric inserts itself in the middle.

Table 7.16: Credit card third validation scenario.

Feature Reduction	Anomaly Counter	MCAV	Accuracy	Elapsed Time
KPCA-Linear	14 391	0.558	0.714	5.641
KPCA-Gaussian	20 933	0.343	0.624	5.266
KPCA-Polynomial	170	0.252	**0.908**	4.709
KPCA-Sigmoid	12 601	0.617	0.738	4.602
AE	20 119	0.171	0.637	4.722
AEKPCA-Linear	14 677	0.142	0.711	4.657
AEKPCA-Gaussian	11 813	0.118	0.749	4.928
AEKPCA-Polynomial	2	0.000	**0.910**	4.392
AEKPCA-Sigmoid	13 501	0.111	0.725	5.169

Concerning the anomaly counter metric, there are two values that do not follow the same pattern. These values, 170 and 2, belong to the KPCA-Polynomial-based and AEKPCA-Polynomial-based approaches, respectively. With data pre-processed by these two techniques, the DCA failed to classify the anomalies as it should have. The highest number of anomalies found was recorded with the KPCA-Gaussian-based approach, as Table 7.16 shows.

With respect to accuracy, the two outliers used polynomial-based techniques. Apart from this, the AEKPCA-Gaussian-based achieved the highest accuracy value, with 0.749. The MCAV metric of this technique also has a positive value, with 0.343, considering the accuracy computed.

Overall, the KPCA-Gaussian-based and the AEKPCA-Linear-based approaches outperformed the other approaches, considering all the metrics. The results from using this dataset may suggest that de DCA can also suit better finance data.

7.6 Summary

This chapter evaluates the effectiveness and resilience of the Dendritic Cell Algorithm (DCA) in identifying anomalies. It provides an exhaustive analysis of the validation tests carried out to confirm the results obtained previously. To ensure an exhaustive examination, datasets from various domains, including network analysis, biology and finance, were included in the design of the validation tests. For each dataset, three validation scenarios were used: the original proportions of positive and negative labels were maintained, the positive and negative labels were balanced, and the proportions were reversed.

Validation using various feature reduction strategies produced inconsistent results when applied to network analysis datasets such as OPC UA, Port Scan and DDoS. In general, the KPCA-Gaussian-based method produced the highest accuracy and anomaly identification rates, particularly on the OPC UA dataset, where it achieved an accuracy of 0.464. However, polynomial-based methods struggled and often showed extremely low anomaly detection rates. Compared to the original and inverted situations, the balanced ratio scenario produced better overall accuracy and MCAV scores. KPCA-Gaussian routinely outperformed other kernels, demonstrating its applicability in anomaly detection applications involving networks.

Using the AE and AEKPCA-based methods resulted in notable increases in accuracy and anomaly identification concerning to the Diabetes dataset. With a maximum accuracy of 0.496 in the original proportions scenario, the KPCA-Gaussian-based method was used. With an accuracy of 0.503, the AEKPCA-Sigmoid-based method, however, performed better in the balanced proportions scenario. The scenario with inverted proportions had the best overall accuracy of 0.539, as indicated by the AEKPCA-Polynomial-based method.

From all the datasets, the financial credit card dataset showed the best accuracy results, especially in the initial proportions scenario. Despite detecting a few anomalies, the KPCA-Polynomial method produced an exceptional accuracy of 0.908. The KPCA-Gaussian-based method had superior anomaly detection capabilities, but the AE-based technique provided the highest accuracy of 0.674 in the balanced proportions scenario. Despite the low anomaly detection rates, the AEKPCA-Polynomial-based method achieved an accuracy of 0.910 in the inverted proportions scenario.

Chapter 8

Discussion and Analysis of Results

Once all the tests have been carried out and validated, discussing and analysing the results is important. This chapter aims to present an exhaustive analysis and discussion of the results of various experiments and validations carried out in other chapters. The research questions mentioned previously will also be addressed here. This chapter will be divided into four sections - KPCA, AE and AEKPCA - each of which deals with a different approach to feature reduction. Finally, the research questions mentioned in the introduction will be addressed in this chapter.

8.1 KPCA

Some visualisations will be presented to analyse the results of the tests that used KPCA as a feature reduction technique. Firstly, the preliminary results will be shown, and after that, validation results will be presented. Lastly, comparisons between both techniques will be employed.

8.1.1 Empirical Tests Analysis

Figure 8.1 shows the time variation per dataset and feature reduction technique.

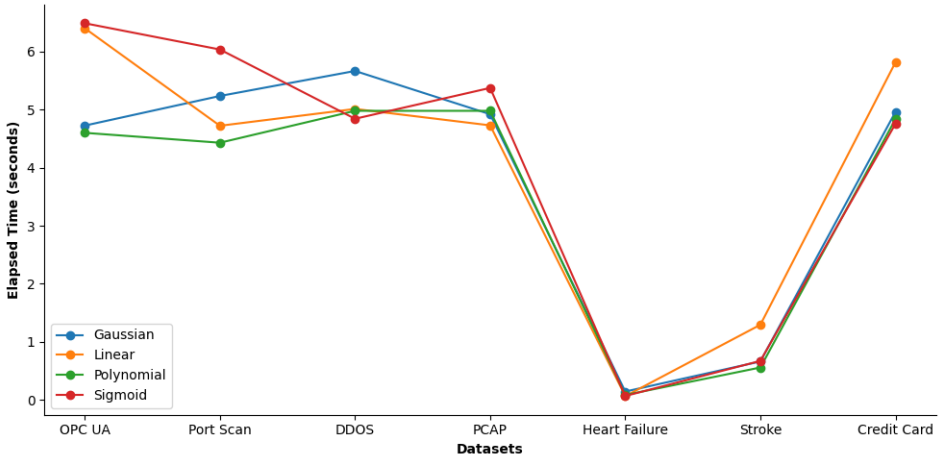

Figure 8.1: Elapsed time by dataset and feature reduction technique.

As it is clear by looking at the figure, there is an interval within which almost all the datasets are. This interval is between 4.5 and 5.5 seconds. The drop visualized is due to the decrease in the number of instances that the biological datasets had. Observing the line plot presented, it is possible to confirm that the lightweight property of the DCA is maintained with data reduction using the KPCA. OPC UA data pre-processed with the KPCA-Sigmoid-based technique demanded the most running time.

Regarding the MCAV metric, some comparisons can be made. As it is possible to confirm in Figure 8.2, the Polynomial-based technique tends to show the highest values compared to the other techniques.

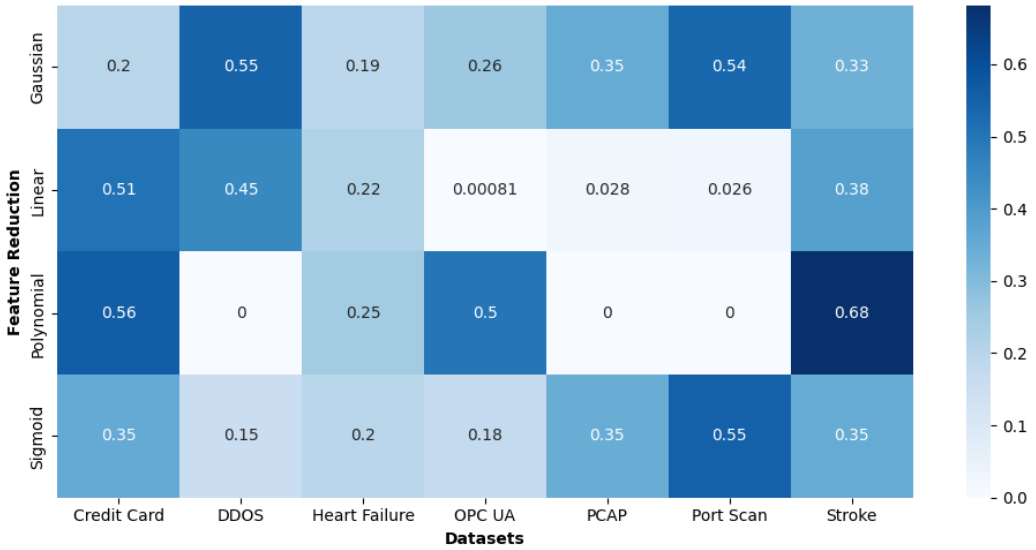

Figure 8.2: MCAV heatmap by dataset and feature reduction technique.

Linear-based and Sigmoid-based approaches present a high variability when applied across the different datasets. In terms of consistency, the Gaussian-based approach had similar values during these tests. Some values are presented as 0, a positive aspect since more mature antigens were presented to the DCs.

The box plot 8.3 provides an idea about the anomaly counter distribution across the different feature reduction techniques. The interquartile range (IQR), which includes the centre 50% of the data, is shown by each box. The median anomaly counter value is shown by the line that appears within each box. Any points outside the whiskers are regarded as outliers. The "whiskers" that extend from the boxes represent variability outside the upper and lower quartiles.

With recourse to the box plot, it is clear the anomaly counter metric varies from technique to technique. As the plot also shows, the Polynomial-based technique presents the widest results, which means that there is variability in its performance. The Linear-based method, in particular, shows an extended IQR, allowing to conclude that the middle 50% of its values span a large range. On the other hand, the Gaussian-based technique presents more consistent results than the other

techniques. This consistency may suggest that it is a more reliable technique to apply to these datasets since it produces data that the DCA can easily process.

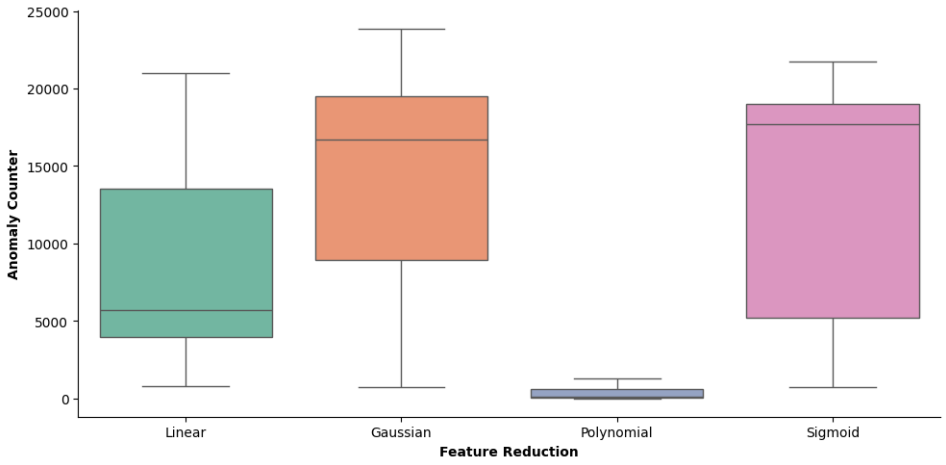

Figure 8.3: Anomaly Counter distribution by feature reduction technique.

Overall, the box plot presented can contribute to finding important insights into the distribution and the consistency of the anomaly counter metric across the different datasets tested. The robustness observed in the performance of DCA when the data was pre-processed through the Gaussian-based approach can be a good starting point for feature studies.

Concerning the accuracy, Figure 8.4 provides an insightful representation of the diversity found in applying the different KPCA-based techniques across the different datasets.

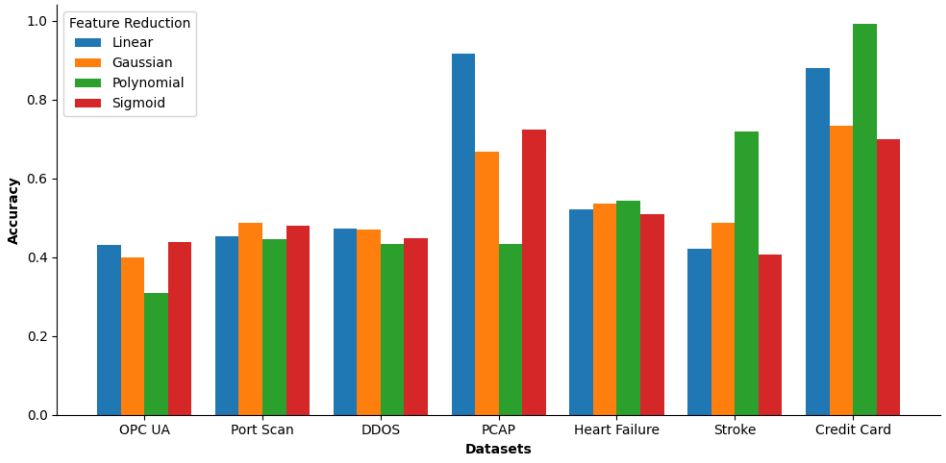

Figure 8.4: Accuracy comparison by dataset and feature reduction technique.

The Polynomial-based approach shows the highest results achieved during this set of tests for the credit card dataset. This value is very close to 1. However, the number of anomalies found is 465, with only 2 true positives. This indicates that, in this particular dataset, the Polynomial-based approach was highly effective in generating new data that the DCA classified as non-anomaly.

Despite of that, for datasets like port scan or DDoS the technique did not perform the same way, which states the importance of selecting the appropriate feature reduction technique.

The Gaussian-based technique showed some consistency across the different datasets, which can be a proper solution to tune as an automated pre-processing technique in the future. On the other hand, the Sigmoid-based technique showed high variability in the results, suggesting that it can be better suited to some datasets than others. Lastly, the Linear-based technique also provided remarkable results, particularly for the PCAP dataset. However, the diversity in the results provides insights that can be applied as an automated feature reduction technique.

8.1.2 Validation and Performance Tests

Regarding the validation tests, the results will also be analyzed through visualization to interpret the data better. The time necessary to run the DCA differs among the three validation scenarios. To better interpret the data, the results of the three validation scenarios will be presented as an average of the three scenarios.

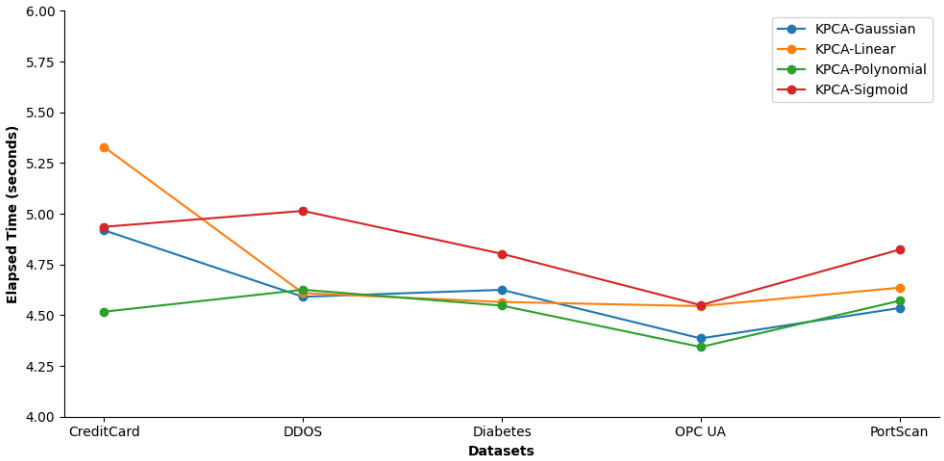

Figure 8.5: Average elapsed time by dataset and feature reduction technique across the three scenarios.

As figure 8.5 shows, there is a diversity across the three scenarios. It also illustrates the performance of each technique considering the competition time. Firstly, the KPCA-Gaussian-based approach shows the highest consistency over the selected datasets. This consistency may suggest that it can be a reliable choice for feature reduction due to its scalability.

On the other hand, the KPCA-Linear-based approach showed more variability, achieving the longest time necessary to run the DCA. The Polynomial-based approach had more ups and downs, and the Sigmoid-based approach showed the highest average time to run the DCA in all the datasets except in the OPC UA.

Despite these heterogeneous results, the differences between each technique are not great enough to provide a conclusion. Based on the results, it is possible to say that all the techniques kept the lightweight property of the DCA, which was an objective of this dissertation.

With respect to the different values of MCAV recorded during the validation tests, some insights are worth mentioning. The results provided by the KPCA-Gaussian-based approach are more consistent, although they are slightly high. The approaches were ineffective in pre-processing the port scan datasets regarding antigen presentation. On the other hand, it clearly suited the OPC UA dataset better since it recorded the two lowest values.

These values reveal that no method performs better across all datasets, meaning each technique has its strengths and weaknesses. Once again, these data indicate the importance of selecting the right feature selection technique to pre-process the data. Although neither of the techniques achieved great results, the KPCA-Linear-based might have outperformed the others. Figure 8.6 show the average MCAV values over the different datasets.

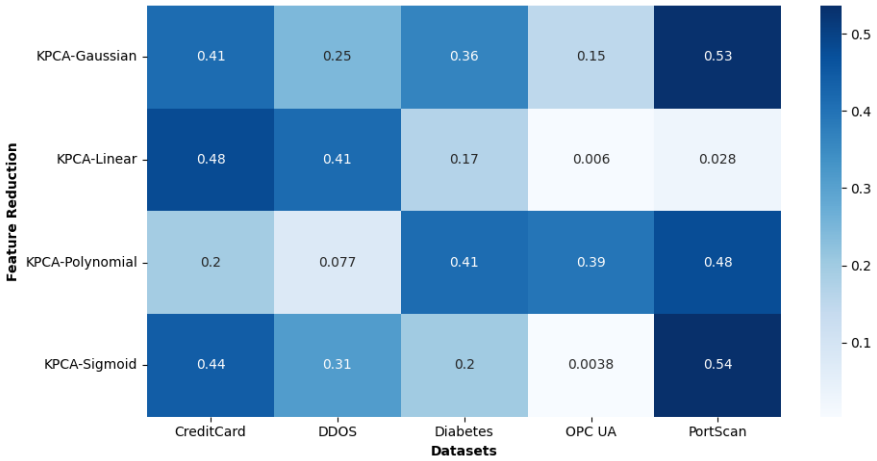

Figure 8.6: Average MCAV by dataset and feature reduction technique across the three scenarios.

As far as the value of the anomalies found by DCA is concerned, after pre-processing the data using the different feature reduction techniques, Figure 8.7 illustrates the results. These results show a wide variation among the different techniques. The Polynomial-based approach presents the largest range of values, with one outlier clearly identified.

The KPCA-Linear-based approach is the technique that shows the highest IQR, demonstrating that the middle 50% of its values are placed in a large range. This variability in the results might also suggest a capacity to deal with several datasets. Although more flexible than the Gaussian and Sigmoid, the greater amplitude means that it may not be as good at reliably capturing complex interactions. The Gaussian-based approach, once again, shows the most consistent results, with a narrower IQR. This insight suggests that it is the more reliable technique for this group of datasets since it produces data that the DCA can handle more easily.

Overall, through this box plot, it is possible to confirm that the Gaussian-based technique is a robust choice to pre-process the data before feeding the DCA since it presents an optimal balance between efficiency and consistency. Conversely, the Polynomial-based approach might need some improvements in the future. Despite its potential, its poor performance in producing data that the DCA can easily process may suggest that it is overfitted. As already mentioned, the Linear-based approach is consistent, but it probably fails to capture complex interactions.

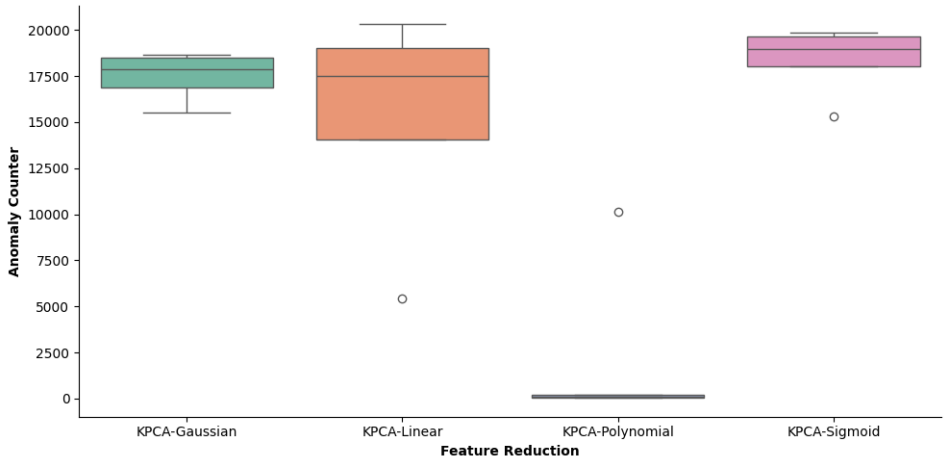

Figure 8.7: Average Anomaly Counter distribution by feature reduction technique.

Lastly, the average accuracy across the three different validation scenarios is stated as probably the most important metric to evaluate DCA's performance. From the Figure 8.8 is clear the representation of the accuracy metric. As mentioned before, these values are the average of the three scenarios, so each bar represents the average accuracy using a specific feature reduction technique with different datasets.

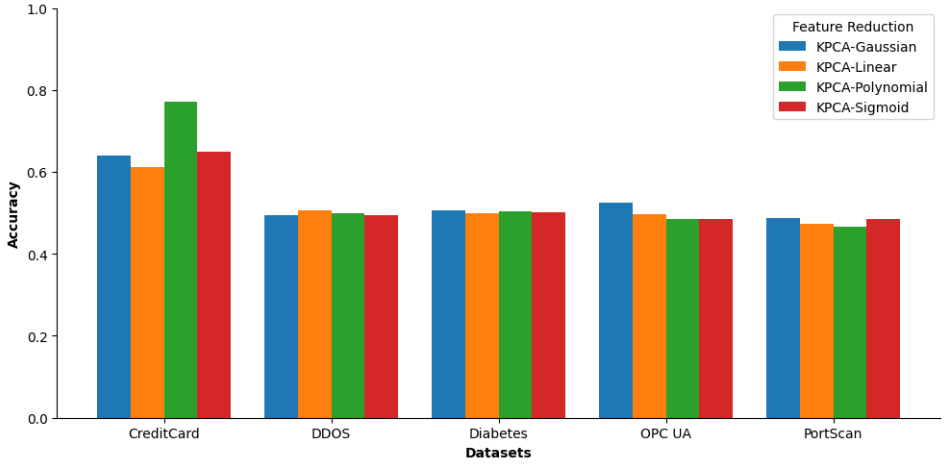

Figure 8.8: Average accuracy comparison by dataset and feature reduction technique.

From the illustration, some insights can be taken. As for the credit card dataset, it is clear that the Polynomial-based approach outperforms the others, achieving the highest accuracy value. Regarding the DDoS dataset, all four techniques showed similar results, suggesting they did not have the expected impact. The same behaviour is found when the referred techniques pre-process the other datasets. The Sigmoid-based approach presented the lowest values across all the datasets.

Overall, the performance observed by each feature reduction technique varies more in some datasets than others. This may suggest that the datasets' characteristics impact the results obtained by each approach. This also suggests that these techniques must be further improved to suit all

the domains in which the DCA is applied. In this particular test, only the credit card dataset was successfully pre-processed using the KPCA-based approaches.

According to this study, the best DCA results can be obtained by selecting a feature reduction strategy relevant to the attributes in the dataset, which needs to be improved in the future.

8.2 AE

Regarding the AE-based tests, the results will be analysed like before. Firstly, the experimental test results will be presented, and then the validation results will be shown. Comparisons between both groups of tests will also be made.

8.2.1 Empirical Tests Analysis

Concerning the experimental tests performed after the data was pre-processed through the AE-based approach, some insights arose that allowed for proper comparisons. As mentioned before, the time necessary to run the DCA was considered since one of this dissertation's objectives was to keep the algorithm's lightweight property.

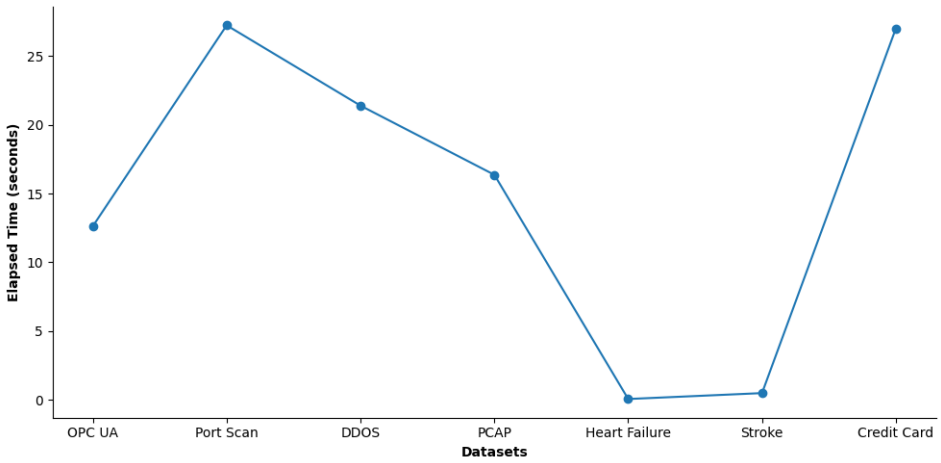

Figure 8.9: Elapsed time by dataset.

Through the analysis of Figure 8.9, it is easy to confirm the variety of time needed to run the algorithm. This variety is due to the differences in the dataset's shapes. The Port Scan dataset was the longest one, so it took more time. The running is proportional to the shape, so it might suggest that this DCA's version maintains its lightweight properties.

Comparing the results presented in Figure 8.10, the consistency found across the different datasets is clear. The best value was achieved after pre-processing the DDoS dataset. These results reveal that some work must be done to improve the AE-based performance since having values closer to 0 would be important.

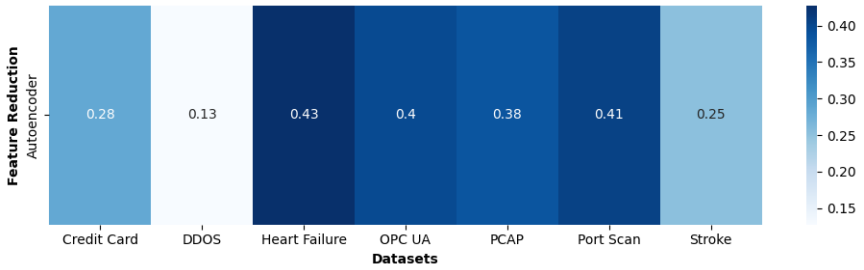

Figure 8.10: MCAV by dataset.

Regarding the anomaly counter metric, Figure 8.11 presents the distribution of the values retrieved. The median value is 10000, indicating that there are 50% of values are below and 50% above. The IQR is quite large, which indicates variability within the metric. The data is positively skewed, as the boxplot shows, and longer upper whiskers indicate data sets with much higher anomaly counts than others.

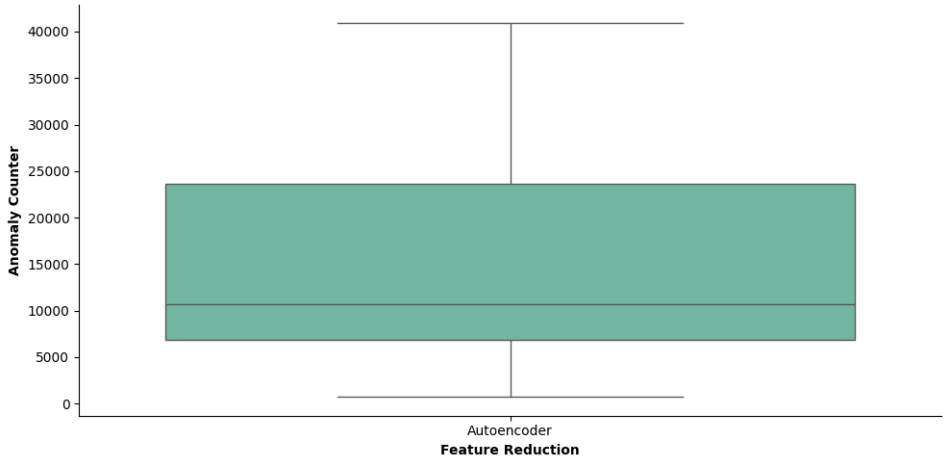

Figure 8.11: Anomaly counter distribution .

Concerning the accuracy computed by the DCA after the ingestion of the data pre-processed by this approach, the results are presented in Figure 8.12. The PCAP and the credit card datasets show very good accuracies, which suggests that the AE-based approach could be a good solution to the problem. However, some tuning might be needed to address the other metrics better. For the other datasets, the medium accuracy also suggests that this has potential that needs to be worked on.

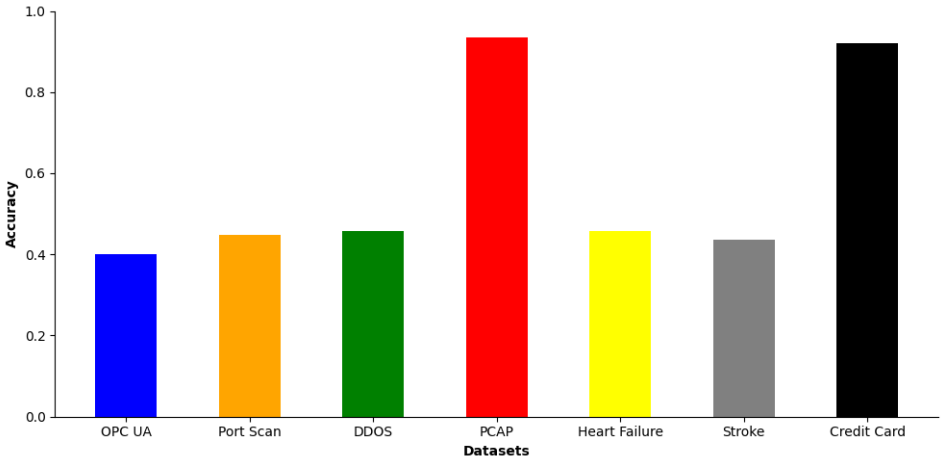

Figure 8.12: Accuracy comparison by dataset.

8.2.2 Validation and Performance Tests

Regarding the validation results, once again, the average of the three scenarios was considered to allow a better comprehension of the data. As mentioned in the previous validation tests analysed, a pattern was followed in terms of the time needed to run the model. Since these datasets had 60000, the time necessary was kept within the interval. Figure 8.13 shows the tendency during the tests, where it is possible to confirm maintenance of the lightweight property of the DCA.

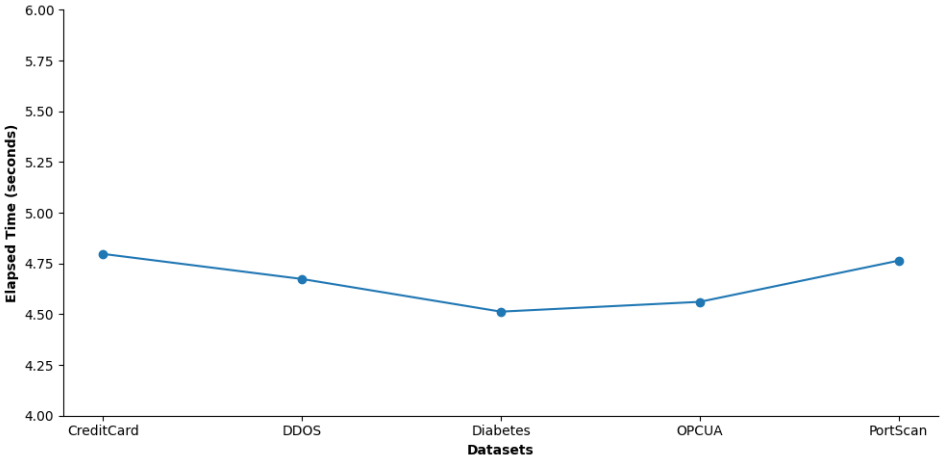

Figure 8.13: Average elapsed time by dataset.

With respect to the MCAV metric recorded in the AE-based, the results are presented in Figure 8.14. From the heatmap, it is possible to see the variations of the results across the different datasets. Since there are no values close to 0, considering this metric, it might suggest that the AE-based approach underperformed in generating data that the DCA can process. The lowest value was recorded in the Credit Card dataset, which means that, on average, more mature antigens were presented using this dataset.

With regard to the anomaly counter metric, there are few differences in comparison with the experimental tests. The distribution appears to be slightly skewed, with a concentration of values at the lower end of the range, as shown by the proximity of the median value to the lower quartile. The wide dispersion shown by the whiskers illustrates how the approach performs differently on different data sets.

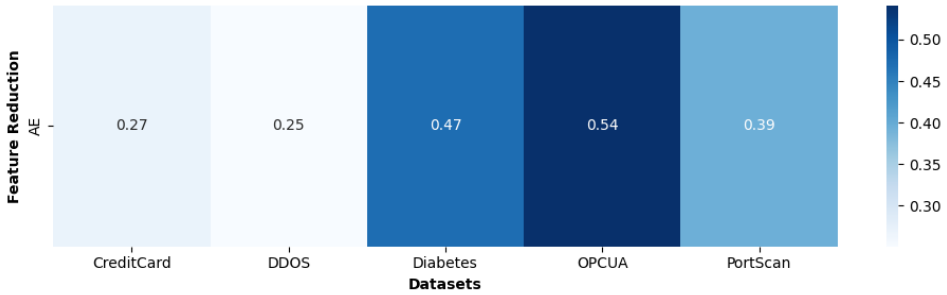

Figure 8.14: Average MCAV by dataset.

Outliers are present, especially below the lower line, which suggests that the AE approach may have found fewer anomalies in certain instances than the normal range, as it is possible to confirm in 8.15. This may demonstrate the method's sensitivity to specific data sets on which it may or may not perform extremely well.

Since the number of rows was 60000 and the IQR shows values between 12000 and 16000, the AE-based approach must be tuned to pre-process the data better before DCA's ingestion. This could lead to more anomalies found and, consequently, an improvement in the accuracy value.

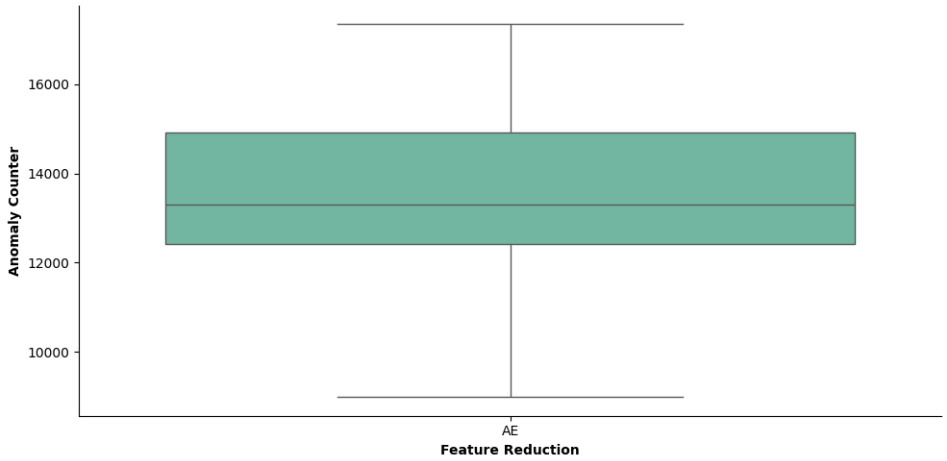

Figure 8.15: Average anomaly counter distribution .

Regarding the accuracy, the last metric to be considered, the results are presented in Figure 8.16. Depending on the datasets, the values may vary. Once again, the credit card dataset achieved the highest accuracy. This states the proficiency of this technique in dealing with this type of data. However, the results are not good enough across all the datasets. Besides the acceptable results in

the Finance domain, improving the technique to produce data capable of being correctly processed by the DCA is necessary. If this issue is overcome, better results may be found.

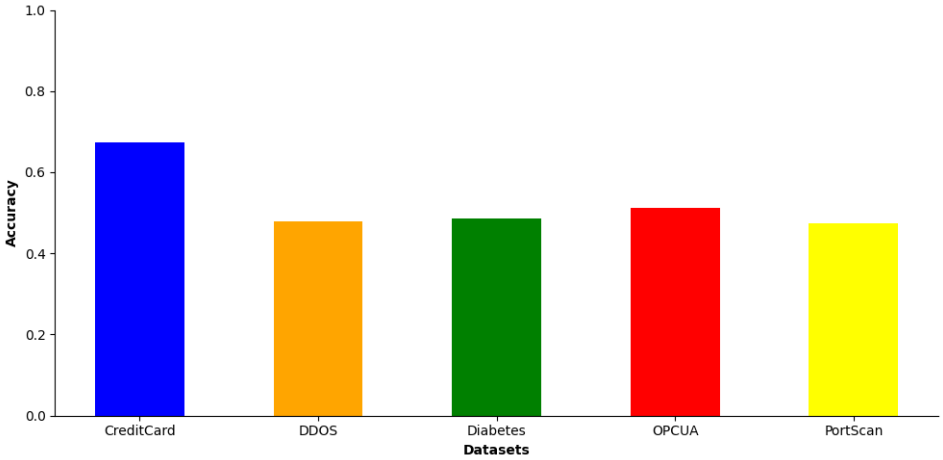

Figure 8.16: Average accuracy comparison by dataset.

8.3 AEKPCA

The last proposed solution to be analyzed is the AEKPCA. The structure will be the same as in the previous approaches. Firstly, the experimental test results will be presented, and then the validation results will be shown. Comparisons between both groups of tests will also be made.

8.3.1 Empirical Tests Analysis

Regarding the AEKPCA-based experimental tests, different insights came to attention. As it is possible to confirm from Figure 8.17, the results are quite similar to the ones recorded in the previously mentioned approaches. With datasets with a similar number of instances, the results are between 4 and 6 seconds, whereas in datasets with lower instances, the time necessary to run the algorithm also decreases. Based on that, it is possible to confirm that the lightweight property of the DCA is also maintained.

Concerning the MCAV metric, it is possible to confirm in Figure 8.18 that some improvements were found compared to the previously mentioned tests. The AEKPCA-Polynomial-based approach recorded the highest values in the set of tests, which may suggest that this technique failed to capture the context of the study with some datasets. On the other hand, it showed 0 in the OPC UA and PCAP datasets, which is clearly a positive aspect.

For the AEKPCA-Sigmoid-based approach, the results presented are generally lower when compared to the other techniques. This indicates that the Sigmoid kernel outperformed the other kernels across all the datasets.

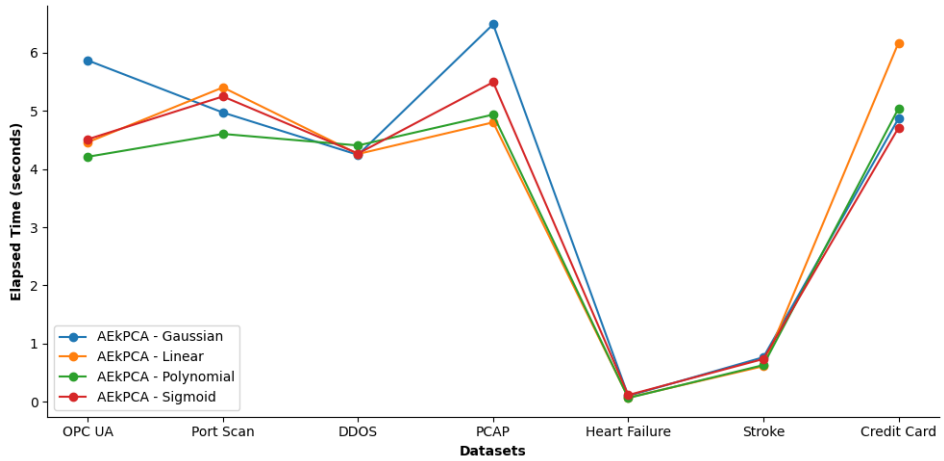

Figure 8.17: Elapsed time by dataset and feature reduction technique.

For the OPC UA dataset, the MCAV values recorded are consistently low compared to the results obtained by the other datasets. Based on this, it might suggest that it is easy for all the techniques to deal with it since they all capture the most important interactions.

Overall, Figure 8.18 presents a visualization of the results referring to the MCAV metric. The AEKPCA-Sigmoid-based approach stands out as the best approach to capturing the context within the data since it allows the most mature antigens to be presented to the DCs. On the other hand, the AEKPCA-Linear-based approach struggled to capture the context within the data and outperformed the different datasets. It is clear from the data that it is necessary to improve this technique in order to present better results.

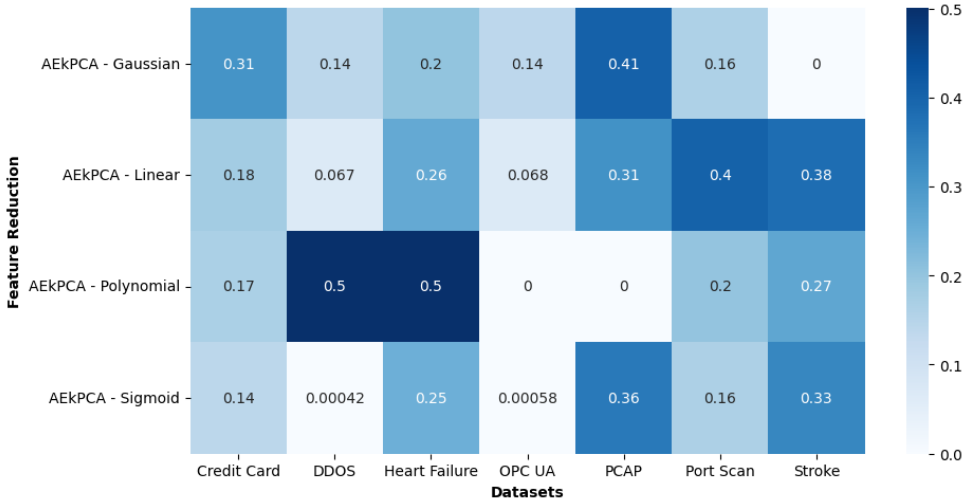

Figure 8.18: MCAV heatmap by dataset and feature reduction technique.

Regarding the anomaly counter metric, the results are quite different from the ones registered in the past. From Figure 8.19, it is evident that the AEKPCA-Linear-based and AEKPCA-Sigmoid-based approaches present the highest range of values, which suggests that variability in

both performances was found.

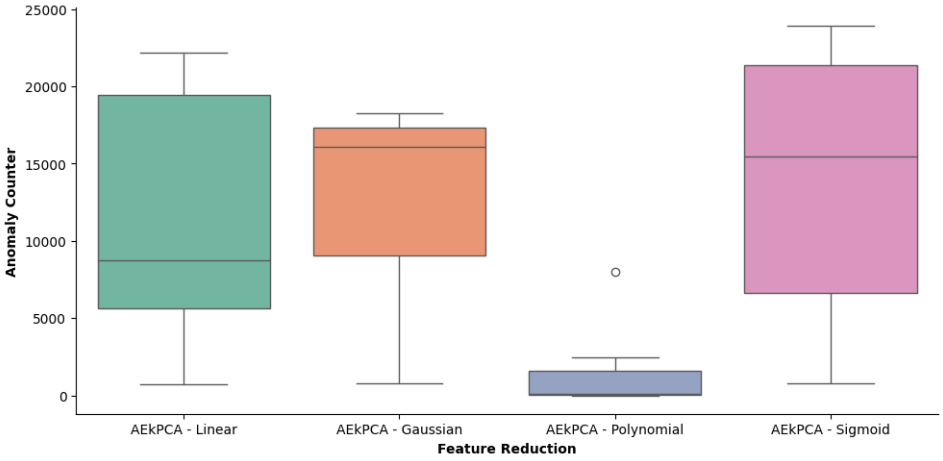

Figure 8.19: Anomaly counter distribution by feature reduction technique.

Both techniques display variability, although the AEKPCA-Linear-based approach recorded higher variability values. The median is clearly higher in the AEKPCA-Sigmoid-based approach. On the other hand, the AEKPCA-Polynomial-based approach was more consistent but underperformed since the intention was to find a higher number of anomalies.

Overall, many improvements need to be made across the different techniques, especially in the Polynomial-based kernel, since both have the potential to generate data that can better suit DCA's needs. Once it is done, we will probably see an increase in the number of anomalies found by the DCA.

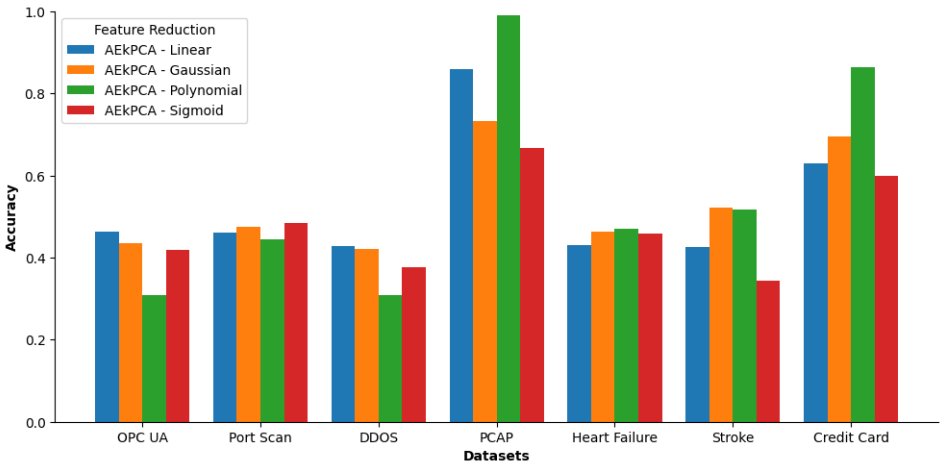

Figure 8.20: Accuracy comparison by dataset and feature reduction technique.

Regarding accuracy, it is possible to confirm from Figure 8.20 that accuracy computed by the DCA increased compared to the other datasets. In terms of datasets, it is evident that the PCAP was the easiest to pre-process using the proposed techniques. This dataset recorded the highest accuracy values with the AEKPCA-Polynomial-based and AEKPCA-Linear-based approaches.

All the techniques proposed also performed well when applied to the Credit Card dataset. Once again, the AEKPCA-Polynomial-based approach stands out as the best technique. The other datasets, namely the OPC UA, Port Scan, DDOS, Heart Failure and Stoked, had medium accuracy results. This suggests that the techniques needed to be improved in order to produce higher results.

Overall, the AEKPCA-Polynomial-based approach outperformed the others in three datasets, and it was the most impactful technique tested. Despite that, the rest of the results may suggest that neither of the techniques can be successfully applied to all the domains and produce good values, which was the objective of this dissertation.

8.3.2 Validation and Performance Tests

The last set of tests comprised the AEKPCA-based approach validation tests. To present this set of validation tests, the same criteria were applied, as the data is present on average to improve its interpretation. This criteria allows us to analyse the performance of each technique over the different tests and have a better idea of the drawn trends. Based on that, this group of tests might have provided the most inspiring results to this study, whereas all the results until here were not good enough, and no solution achieved the success that was expected in the beginning.

Regarding the time necessary to run the model, the results are very similar to those that involved time during this dissertation. As it is possible to confirm from Figure 8.21, all the tests lie within the same interval previously mentioned, which is between 4 and 6 seconds. Once again, the lightweight property of the DCA was maintained.

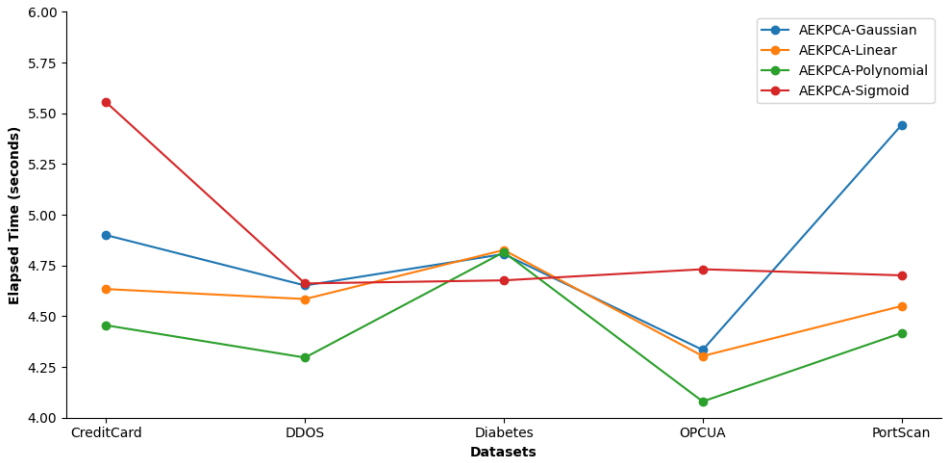

Figure 8.21: Average elapsed time by dataset and feature reduction technique across the three scenarios.

Concerning the MCAV metric, the results are not as good as expected when combining two approaches. The AE and the Kernels provided by the KPCA should be more proficient than they were in capturing the context of the data and traducing it into the newly generated data. Based on the results provided in Figure 8.22, the used technique struggled to reproduce the context inherent in both credit card and DDoS datasets since the results should have been closer to 0.

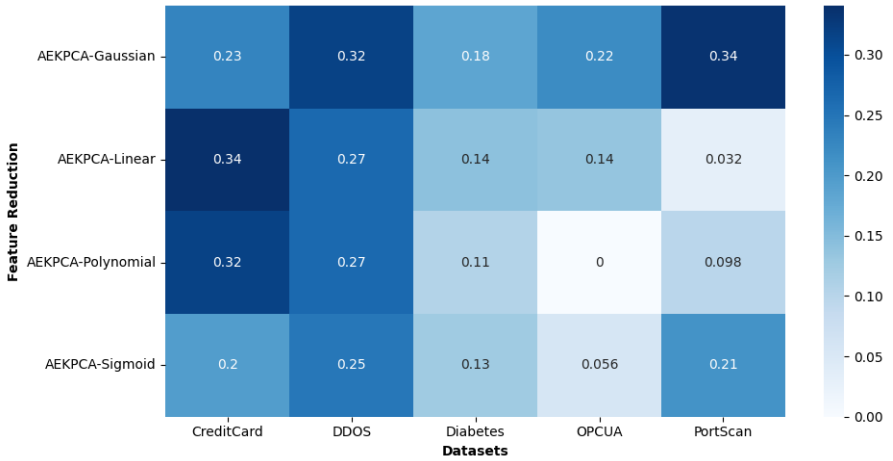

Figure 8.22: Average MCAV by dataset and feature reduction technique across the three scenarios.

On the other hand, the OPC UA, once again, was the easiest to interpret using different techniques, producing the lowest values of MCAV. These values mean that more mature antigens were present in the DCs. In general, the AEKPCA-Polynomial-based technique outperformed the others, which suggests that this approach also failed to replicate the context of data, and it needs to be better tuned in the future to be positioned as a solution to approach these datasets.

Regarding the anomaly counter metric, Figure 8.23 shows its distribution. In terms of IQR, the AEKPCA-Gaussian-based approach shows a relatively narrow range, whereas the values from the AEKPCA-Linear are wider. The AEKPCA-Polynomial has a median close to zero, suggesting that this technique failed to produce data capable of being classified as an anomaly. In general, Gaussian and Sigmoid-based techniques produced data that the DCA can process better. Even though the number of anomalies increased, it failed to achieve the goal of techniques that suit all the needs.

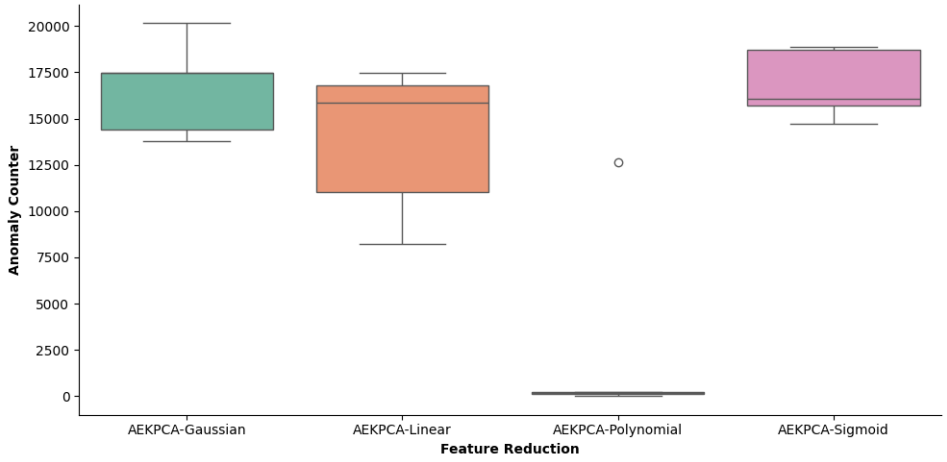

Figure 8.23: Average anomaly counter distribution by feature reduction technique.

With respect to the last metric considered, the accuracy, these values are not as good as expected. The credit card dataset was the easiest for the proposed techniques to find the complex interactions. This is stated by Figure 8.24, where the highest value achieved belongs to the AEKPCA-Polynomial-based approach. For the other datasets, the accuracy values are around 0.500 for all of them. These results showed that no automatic feature reduction technique, although some improvements were made, and this dissertation can improve either.

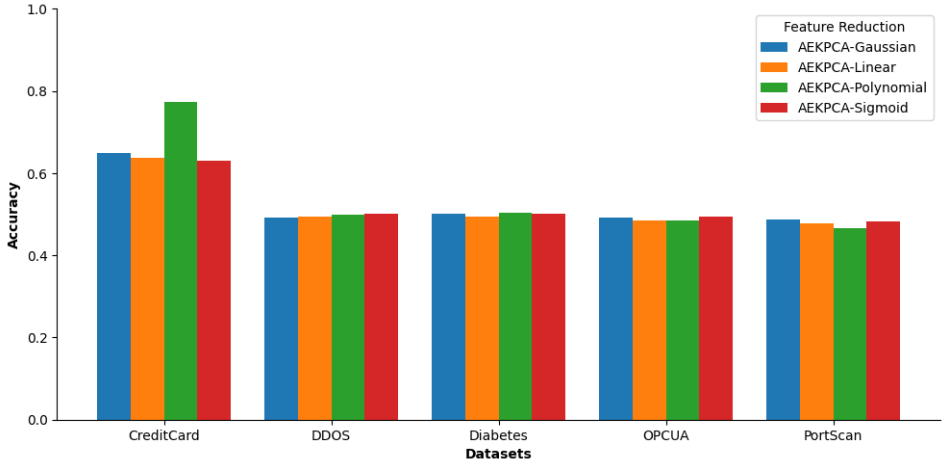

Figure 8.24: Average accuracy comparison by dataset and feature reduction technique.

8.4 Addressing the Research Questions

In this section, we address the research questions that were stated in Section 1.3.1. The aim behind this is to explore what was done to give these answers. As mentioned before, several feature reduction techniques were tested during this dissertation, including KPCA, AE and AEKPCA, in different datasets that covered a wide range of domains. After that, the resulting performance was compared. The research questions are listed as follows:

- **RQ1:** Is the DCA performance dependent on the effectiveness of data pre-processing?
- **RQ2:** Can the DCA be applicable to a larger set of problems when integrated with feature reduction techniques, such as KPCA, AE and AEKPCA?
- **RQ3:** How can KPCA, AE and AEKPCA enhance the data pre-processing phase of the DCA to make the system automated and adaptable to a given problem domain?

In order to answer the research questions, this section is divided into three subsections, each corresponding to one of the research questions. Each subsection includes an introduction, a discussion of the relevant experimental results and a conclusion.

8.4.1 Research Question 1

The first research question that this dissertation aims to answer tries to evaluate how much the pre-processing stage of the data influences the efficiency of DCA. In general, ML algorithms need detailed pre-processing to convert unprocessed data into a format that is better suited to the model in order to perform as it should be. So, this question became important to this study because optimising DCA for better anomaly identification can be achieved by understanding the influence of pre-processing.

Various pre-processing methods such as KPCA, AE and AEKPCA were used to analyse this gap in the literature. In addition, various data sets were considered to evaluate both approaches, namely OPC UA, port scan, DDoS, PCAP, heart failure, stroke and credit card or diabetes. It follows that the intention to include a wide range of domains was important for this study, allowing the applicability of DCA to be tested.

As shown in Section 6.2, the baseline test was carried out without automatic feature reduction and showed serious results in terms of performance. In turn, the result differed greatly when the KPCA was used with various kernels, including Linear, Gaussian, Polynomial and Sigmoid. Using the Polynomial kernel for the OPC UA dataset resulted in an accuracy of 0.309, which is the lowest value recorded in this approach. This may suggest that this kernel failed to recognise significant patterns in this domain. In contrast, the Sigmoid kernel came out with 0.438, the highest accuracy achieved, although this is not a good enough result.

The efficiency behind the AEs - identify non-linear relationships - varies. As an example, the AE used on the credit card dataset resulted in values similar to the ones found in the baseline test. However, it is not the same dataset. In the OPC UA dataset used in the baseline test, the accuracy retrieved by the data provided from the AE is 0.400.

The experimental results clearly show that the success of the data pre-processing stage has a significant impact on the performance of DCA. Sophisticated feature reduction methods, such as KPCA and AE, can considerably improve DCA's ability to identify anomalies, offering a more succinct and enlightening picture of the data. The key elements in maximising DCA performance are the selection of the pre-processing technique and the way in which it preserves important information from the original dataset.

8.4.2 Research Question 2

The second research question investigates DCA's ability to apply to a larger set of domains with feature reduction techniques embedded. As the last research question made clear, within the pre-processing stage, feature reduction techniques are essential to handle high-dimensional data and considerably improve the model's performance. This study aims to find if integrating the previously referred feature reduction techniques can enhance DCA's application spectrum.

In order to evaluate DCA's robustness, the tests were performed in the same datasets mentioned before, which can guarantee diversity in the chosen sample. The main objective of this research

question was to characterize DCA's performance across several domains with the feature reduction techniques embedded.

With regard to the OPC UA dataset, it is possible to confirm the difference between the results when feature reduction is applied. The baseline test, which used this dataset, found an accuracy of 1, with the pre-processing phase being done manually. Comparing these values to the ones found in the data provided by the automatic feature reduction, is clear that performance is not similar.

The other datasets followed the same behaviour since almost all their accuracy results were below 0.500. However, some important mentions should be made, namely the PCAP and the Credit Card datasets. The first one recorded an accuracy of 0.935 and 0.990 for the AE and AEkPCA - Polynomial, respectively. Despite that, the number of anomalies found is very low, which may suggest that the model is overfitting. The second recorded 0.991 and 0.865 for the same techniques, respectively. Once again, this may suggest that the model is overfitting due to the low number of anomalies found.

Integrating different and automated feature reduction techniques, such as the ones discussed in this dissertation, can effectively enhance the data quality before feeding the DCA. Although the results were not good enough, some good results were found, which means that some improvements to the architecture could become fundamental in this process. So, the experimental tests may suggest that the DCA can actually be applied to a different range of problems than the ones it was applied before. This shows the robustness and the versatility inherent in this algorithm.

8.4.3 Research Question 3

The last research question that this dissertation aims to answer focuses on looking into possible changes in the DCA's pre-processing pipeline that can effectively contribute to automating and adapting the DCA to various domains. In a real-world application, which this study aims to bring as close as possible, the data quality can vary greatly, which means that automation and adaptability are extremely necessary. The intention behind this research question is to find processes that can enhance DCA's power and turn it into a more user-friendly algorithm.

Once again, feature reduction techniques were integrated with the DCA, namely KPCA, AE and AEkPCA, to answer this research question. A pre-processing pipeline has been created that adapts to different domains of data sets without human effort. This pipeline can be optimized in the future.

The pipeline had different steps and was designed to be dynamic, which means that it is possible to change different parameters within the function without changing the whole code. By changing these parameters, this will produce a different outcome.

First of all, the pipeline begins with the possibility to choose the intended normalization technique. In the particular case of this study, Z-score normalization was selected. After the normalization process, the automated feature reduction techniques were applied. For example, KPCA with four different kernels was applied to various datasets that lately fed the algorithm. Similar to this approach, the AE and the AEkPCA were also applied to produce new representations of the data. Gathered all the data from the different sources, the DCA was computed, and different

metrics were calculated, such as the number of anomalies found, the accuracy, the MCAV and the time necessary to run the algorithm.

Drawing a pipeline to address the pre-processing phase can be very effective when it comes to addressing different problem domains. This dissertation emphasises the necessity of developing an automated pipeline to pre-process the data before DCA's ingestion. Thus, this study shows that DCA's performance in a wide range of domains can be effectively enhanced by automating the pre-processing phase. By doing this, DCA can be more robust, user-friendly and improve its adaptability. The results underline the need for flexibility in developing machine learning algorithms for practical applications, as well as the possibility of further optimization.

Chapter 9

Conclusion

This dissertation aimed to address the literature gap associated with the pre-processing phase of the DCA. Experts in the given domain usually do this pre-processing phase, leading to exhaustive dataset analysis and a long time. The first problem identified by this study is the need for techniques to reduce the number of features in large-scale datasets to enhance DCA's efficiency and accuracy in anomaly detection. To solve this, this dissertation proposed three feature-reduction techniques: KPCA, AE and a combination of both approaches called AEkPCA.

A comprehensive set of experimental and validation tests were performed regarding the methodology followed during this study. These datasets represented different domains, including biology, finance and cybersecurity. The first set of experiments, concerning the experimental tests, were made over 7 datasets. The pipeline followed during this step involved the transformation of the categorical labels, normalising the data with the Z-score technique, training the feature reduction approaches and running the generated datasets with the DCA. In the second group of tests, 3 validation scenarios were designed, including the first with the original proportion of positive and negative labels, the second with equal proportions and the last with the original proportions inverted. All the data was normalised using the same technique, and each sampled dataset had 60000 instances. The evaluation metrics included the elapsed time, MCAV, anomaly counter and accuracy.

The experimental results show different degrees of success over the different techniques and datasets. Although some techniques performed well when applied to a particular dataset, neither presented sustained results across all datasets. Based on this, the objective of this dissertation was not accomplished.

9.1 Impact on literature

As far as the impact on the literature is concerned, the proposed solution contributes significantly to the feature reduction and anomaly detection areas, especially with respect to the DCA. Integrating and testing advanced feature reduction techniques such as KPCA, AE, and AEkPCA with the DCA provided insights that improved DCA's aspiration to become an autonomous algorithm. The

stability of the KPCA in a range of application domains is emphasised by its constant performance, particularly when the Gaussian kernel is used. This result complements and expands on other studies that emphasise the importance of extracting non-linear features to improve the performance of machine learning models.

Although the study has some limitations, the AE approach contributes to the literature by illuminating the opportunities and difficulties associated with deep learning-based feature reduction in anomaly detection scenarios. Despite its inconsistent results, the AEkPCA hybrid methodology offers insightful information on the challenges of integrating various feature reduction techniques. This study offers a path for future research to improve on its findings, reinforcing its contributions to the state of the art in anomaly identification through methodological rigour and exhaustive validation on various datasets.

9.2 Limitations

Although this study has made several significant advances, it has negative aspects. The main drawback is inconsistent performance between datasets, especially with the AE and AEKPCA methods. Without further improvements, the AE approach may not be universally applicable across all data types, as seen by its difficulty in maintaining low levels of MCAV. Similarly, the uneven performance of AEKPCA suggests that these combined approaches must be properly integrated and fine-tuned.

Furthermore, the research was restricted to specific finance, biology and cybersecurity datasets. Extending the tests to larger and more varied data sets could provide a more complete assessment of the robustness and usefulness of the suggested methodologies.

9.3 Future Work

It is suggested that future research follow a series of paths to improve the robustness and applicability of the proposed solution in light of the constraints that have been observed. Firstly, the consistency of the AE technique must be improved. This may involve experimenting with various architectures, fine-tuning the hyperparameters and using regularisation strategies to avoid overfitting.

Lastly, the AEKPCA hybrid technique needs to be further developed. Optimising the interaction between AE and KPCA and exploring more complex integration techniques could improve performance on various datasets. In addition, extending validation tests to more diverse and extensive datasets will allow a more complete understanding of the generalisability and scalability of these methods.

Appendix A

Dendritic Cell Algorithm

```
def compute_DCA(self, ag_list, signals_list):
        antigen_counter = 0
        results = []
        # Variable to track the previous selected cell
        cell_track = None
        while len(signals_list) > antigen_counter:
            for ag, sig in zip(ag_list, signals_list):
                test_list = []
                # Increment the antigen counter
                antigen_counter += 1
                # Ensure there are available cells to select from
                if not self.available_cells:
                    # Reinitialize available_cells if it's empty
                    self.available_cells = list(range(len(self.cells)))
                # Determine the DC index
                cell_random_index = random.randrange(len(self.available_cells))
                # Avoid cell reptition
                if cell_random_index == cell_track:
                    cell_random_index = random.randrange(len(self.available_cells))
                else:
                    pass
                cell_index = self.available_cells.pop(cell_random_index)
                cell_track = cell_random_index
                # Select a random index from the antigens' list
                random_index = random.randrange(len(ag_list))
                # Pick the antigen
                antigen = ag_list.pop(random_index)
                # Assign the random antigen to the random dc
                self.cells[cell_index].assign_antigen(antigen)
                # Calculate csm
                csm = self.csm(antigen[1][1][0], antigen[1][1][1])
                # Calculate k
                k = self.k(antigen[1][1][0], antigen[1][1][1])
```

```
34                 for i, dc in enumerate(self.cells):
35                     # Verify the any Ag was assigned to the DC
36                     if len(dc.antigen_store) > 0:
37                         # Update the values
38                         dc.update(csm, k)
39                     # Verify if the cell is mature
40                     if dc.is_mature():
41                         results_dict = {
42                             'lifespan': dc.lifespan,
43                             'k' : dc.k,
44                             'ag_profile': dc.antigen_store,
45                         }
46                         results.append(results_dict)
47                         dc.reset()
48                         # Turn cell available again
49                         self.available_cells = list(range(len(self.cells)))
50 
51         return results
```

Listing A.1: DCA implementation

A.1 K

```
1 def k(self, danger_signal, safe_signal):
2     return danger_signal - 2 * safe_signal
```

Listing A.2: K value implementation

A.2 CSM

```
1 def csm(self, danger_signal, safe_signal):
2     return safe_signal + danger_signal
```

Listing A.3: CSM implementation

References

[1] G. Dranoff. Cytokines in cancer pathogenesis and cancer therapy. *Nature Reviews Cancer*, 4:11–22, 2004.

[2] E. Palmer. Negative selection - clearing out the bad apples from the t-cell repertoire. *Nature Reviews Immunology*, 3:383–391, 2003.

[3] Gustav J. V. Nossal. The double helix and immunology. *Nature*, 421(6921):440–444, January 2003.

[4] Leandro De Castro and Jon Timmis. Artificial immune system: A novel paradigm to pattern recognition. *ANNIS*, 8, 08 2002.

[5] Kenneth Murphy, Paul Travers, Mark Walport, and Charles Janeway. *Janeway's immunobiology*. Garland Science New York, New York, 8th ed edition, 2012. Section: xix, 868 pages : illustrations (chiefly color) ; 28 cm.

[6] Khaled A. Alaghbari, Heng-Siong Lim, Mohamad Hanif Md Saad, and Yik Seng Yong. Deep Autoencoder-Based Integrated Model for Anomaly Detection and Efficient Feature Extraction in IoT Networks. *IoT*, 4(3):345–365, August 2023.

[7] Francisco J. Pulgar, Francisco Charte, Antonio J. Rivera, and María J. del Jesus. AEkNN: An AutoEncoder kNN-based classifier with built-in dimensionality reduction, March 2018. arXiv:1802.08465 [cs].

[8] John E. Hunt and Denise E. Cooke. Learning using an artificial immune system. *Journal of Network and Computer Applications*, 19(2):189–212, April 1996.

[9] D. R. Flower and J. Timmis. *In Silico Immunology*. Springer, 2007.

[10] L. N. de Castro and J. Timmis. *Artificial Immune Systems: A New Computational Intelligent Approach*. Springer-Verlag, 2002.

[11] W. E. Paul. The Immune System – Complexity Exemplified. *Mathematical Modelling of Natural Phenomena*, 7(5):4–6, January 2012. Publisher: EDP Sciences.

[12] P. Bretscher and M. Cohn. A theory of self-nonself discrimination. *Science (New York, N.Y.)*, 169(3950):1042–1049, September 1970.

[13] CA Jr Janeway. How the immune system recognizes invaders. *Scientific American*, 269(3):72–79, September 1993.

[14] Robert Oates, Julie Greensmith, Uwe Aickelin, Jonathan M. Garibaldi, and Graham Kendall. The Application of a Dendritic Cell Algorithm to a Robotic Classifier, April 2010. arXiv:1004.5222 [cs].

[15] Julie Greensmith, Jan Feyereisl, and Uwe Aickelin. The dca: Some comparison a comparative study between two biologically-inspired algorithms. *Evolutionary Intelligence*, 1(2):85–112, 2008.

[16] Julie Greensmith. *The Dendritic Cell Algorithm*. PhD thesis, School of Computer Science, University of Nottingham, 2007.

[17] Robert Oates. *The Suitability of the Dendritic Cell Algorithm for Robotic Security Applications*. PhD thesis, School of Computer Science, University of Nottingham, 2010.

[18] Julie Greensmith, Uwe Aickelin, and Jamie Twycross. Articulation and Clarification of the Dendritic Cell Algorithm, October 2009. arXiv:0910.4903 [cs].

[19] Julie Greensmith, Uwe Aickelin, and Gianni Tedesco. Information fusion for anomaly detection with the dendritic cell algorithm. *Information Fusion*, 11:21–34, 2010.

[20] Yousof Al-Hammadi, Uwe Aickelin, and Julie Greensmith. DCA for Bot Detection. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 1807–1816, June 2008. arXiv:1001.2195 [cs].

[21] Jungwon Kim, Peter Bentley, Christian Wallenta, Mohamed Ahmed, and Stephen Hailes. Danger Is Ubiquitous: Detecting Malicious Activities in Sensor Networks Using the Dendritic Cell Algorithm. In *Artificial Immune Systems*, volume 4163, pages 390–403, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. Series Title: Lecture Notes in Computer Science.

[22] Nathan Lay and Iain Bate. Improving the reliability of real-time embedded systems using innate immune techniques. *Evolutionary Intelligence*, 1:113–132, 2008.

[23] Wei Wang, Chunxiao Zhang, and Qiang Zhang. An anomaly detection model based on cloud model and danger theory. In *Proceedings of the international standard conference on trustworthy computing and services, ISCTCS*, pages 115–122, 2013.

[24] R. Coico, G. Sunshine, and E. Benjamini. *Immunology: A Short Course*. John Wiley & Sons, Inc., 5th edition, 2003.

[25] Arpan Kumar Kar. Bio inspired computing – a review of algorithms and scope of applications. *Expert Systems with Applications*, 59:20–32, 2016.

[26] Thomas Stibor, Jonathan Timmis, and Claudia Eckert. A Comparative Study of Real-Valued Negative Selection to Statistical Anomaly Detection Techniques. In *Artificial Immune Systems*, volume 3627, pages 262–275, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. Series Title: Lecture Notes in Computer Science.

[27] N. Owens, A. Greensted, J. Timmis, and A. Tyrrell. T-cell receptor signalling inspired kernel density estimation and anomaly detection. In *Proceedings of the 8th International Conference on Artificial Immune Systems (ICARIS)*, pages 122–135, 2009.

[28] Manfred B. Lutz and Gerold Schuler. Immature, semi-mature and fully mature dendritic cells: which signals induce tolerance or immunity? *Trends in Immunology*, 23(9):445–449, September 2002. Publisher: Elsevier.

[29] Feng Gu, Julie Greensmith, and Uwe Aickelin. Further Exploration of the Dendritic Cell Algorithm: Antigen Multiplier and Time Windows, March 2010. arXiv:1003.0319 [cs].

[30] Zeineb Chelly and Zied Elouedi. A survey of the dendritic cell algorithm. *Knowledge and Information Systems*, 48(3):505–535, September 2016.

[31] Feng Gu. *Theoretical and Empirical Extensions of the Dendritic Cell Algorithm*. PhD thesis, University of Nottingham, November 2011.

[32] Feng Gu, Julie Greensmith, Robert Oates, and Uwe Aickelin. PCA 4 DCA: The Application Of Principal Component Analysis To The Dendritic Cell Algorithm, April 2010. arXiv:1004.3460 [cs].

[33] Stephanie Forrest. Genetic algorithms. *ACM Computing Surveys*, 28(1):77–80, March 1996.

[34] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, July 1983.

[35] Sun-Chong Wang. Artificial Neural Network. In *Interdisciplinary Computing in Java Programming*, pages 81–100. Springer US, Boston, MA, 2003.

[36] Tom M. Mitchell. *Machine Learning*. McGraw-Hill series in computer science. McGraw-Hill, New York, 1997.

[37] Dario Floreano and Claudio Mattiussi. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. MIT Press, August 2008. Google-Books-ID: 1S03AgAAQBAJ.

[38] Arpan Kumar Kar. Bio inspired computing – A review of algorithms and scope of applications. *Expert Systems with Applications*, 59:20–32, October 2016.

[39] Jungwon Kim, Peter J. Bentley, Uwe Aickelin, Julie Greensmith, Gianni Tedesco, and Jamie Twycross. Immune System Approaches to Intrusion Detection - A Review. *Natural Computing*, 6(4):413–466, December 2007. arXiv:0804.1266 [cs].

[40] Peter J Delves. The Immune System. *ADVANCES IN IMMUNOLOGY*, 2000.

[41] R. Coico, G. Sunshine, and E. Benjamini. *Immunology: A Short Course*. John Wiley & Sons, Inc., 5th edition, 2003.

[42] Peter Parham. *The immune system*. Garland Science, 2014.

[43] Stuart E. Turvey and David H. Broide. Innate immunity. *Journal of Allergy and Clinical Immunology*, 125(2):S24–S32, February 2010.

[44] Francisco A. Bonilla and Hans C. Oettgen. Adaptive immunity. *Journal of Allergy and Clinical Immunology*, 125(2):S33–S40, February 2010.

[45] J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune learning, adaptation, and machine learning. *Physica D*, 4:187–204, 1986.

[46] J Kuby. *Immunology*. W. H. Freeman and Co., 3rd edition, 1997.

[47] K. Rajewsky. Clonal selection and learning in the antibody system. *Nature*, 381:751–758, 1996.

[48] M. Siper. *Introduction to the Theory of Computation*. Course Technology, 2nd edition, 2005.

[49] I. R. Cohen. Immune system computation and the immunological homunculus. *Model Driven Engineering Languages and Systems*, 4199:499–512, 2006.

[50] J. Timmis, P. Andrews, N. Owens, and E. Clark. An interdisciplinary perspective on artificial immune systems. *Evolutionary Intelligence*, 1(1):5–26, March 2008.

[51] I. R. Cohen. Real and artificial immune systems: Computing the state of the body. *Immunological Reviews*, 7:569–574, 2007.

[52] S. Forrest, A.S. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 202–212, Oakland, CA, USA, 1994. IEEE Comput. Soc. Press.

[53] Jon Timmis and Jon Timmis, editors. *Artificial immune systems: second international conference ; proceedings*. Number 2787 in Lecture notes in computer science. Springer, Berlin Heidelberg, 2003. Meeting Name: ICARIS.

[54] Jon Timmis, Mark Neal, and John Hunt. An artificial immune system for data analysis. *Biosystems*, 55(1-3):143–150, February 2000.

[55] T. Knight and J. Timmis. Aine: An immunological approach to data mining. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 297–304, Los Alamitos, CA, USA, December 2001. IEEE Computer Society.

[56] Uwe Aickelin and Steve Cayzer. The Danger Theory and its Application to Artificial Immune Systems. *SSRN Electronic Journal*, 2002.

[57] Julie Greensmith, Uwe Aickelin, and Steve Cayzer. *Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection*, pages 153–167. Springer, 2005.

[58] Sahar Aldhaheri, Daniyal Alghazzawi, Li Cheng, Ahmed Barnawi, and Bandar A. Alzahrani. Artificial Immune Systems approaches to secure the internet of things: A systematic review of the literature and recommendations for future research. *Journal of Network and Computer Applications*, 157:102537, May 2020.

[59] Zhou Ji and Dipankar Dasgupta. Revisiting Negative Selection Algorithms. *Evolutionary Computation*, 15(2):223–251, June 2007.

[60] Dipankar Dasgupta, Senhua Yu, and Fernando Nino. Recent Advances in Artificial Immune Systems: Models and Applications. *Applied Soft Computing*, 11(2):1574–1587, March 2011.

[61] Steven A. Hofmeyr and Stephanie Forrest. Architecture for an Artificial Immune System. *Evolutionary Computation*, 8(4):443–473, December 2000.

[62] F. A. Gonzalez and D. Dasgupta. Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines*, 4(4):383–403, 2003.

[63] Thomas Stibor, Philipp Mohr, Jonathan Timmis, and Claudia Eckert. Is negative selection appropriate for anomaly detection? In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 321–328, Washington DC USA, June 2005. ACM.

[64] T. Stibor. *On the Appropriateness of Negative Selection for Anomaly Detection and Network Intrusion Detection*. PhD thesis, Darmstadt University of Technology, 2006.

[65] Berna Haktanirlar Ulutas and Sadan Kulturel-Konak. A review of clonal selection algorithm and its applications. *Artificial Intelligence Review*, 36(2):117–138, August 2011.

[66] L. D. Davis and M. Mitchell. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

[67] Leandro Nunes De Castro and Fernando J. Von Zuben. The clonal selection algorithm with engineering applications. In *GECCO 2002 - Workshop Proceedings*, pages 36–37. Morgan Kaufmann, 2000.

[68] L.N. De Castro and F.J. Von Zuben. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251, June 2002.

[69] J. Timmis and M. Neal. A resource limited artificial immune system for data analysis. *Knowledge-Based Systems*, 14(3-4):121–130, 2001.

[70] Andrew Watkins, Jon Timmis, and Lois Boggess. Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. *Genetic Programming and Evolvable Machines*, 5(3):291–317, September 2004.

[71] J. Kelsey and J. Timmis. Immune inspired somatic contiguous hypermutation for function optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 207–218, 2003.

[72] N. Cruz Cortes and C. A. Coello Coello. Multiobjective optimization using ideas from the clonal selection principle. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 158–170, 2003.

[73] M. Villalobos-Arias, C. A. Coello Coello, and O. Hernandez-Lerma. Convergence analysis of a multiobjective artificial immune system algorithm. In *Proceedings of the 4th International Conference on Artificial Immune Systems (ICARIS)*, volume 226-235, 2004.

[74] N. K. Jerne. Towards a network theory of the immune system. *Ann. Immunol. (Inst. Pasteur)*, 125C:373–389, 1974.

[75] Emma Hart and Jon Timmis. Application areas of AIS: The past, the present and the future. *Applied Soft Computing*, 8(1):191–201, January 2008.

[76] L. N. De Castro and F. J. Von Zuben. An evolutionary immune network for data clustering. In *Proceedings of Brazilian Symposium on Artificial Neural Networks*, pages 84–89, 2002.

[77] G. Coelho and F. J. Von Zuben. Omni-ainet: An immune-inspired approach for omni optimization. In *Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS)*, pages 294–308, 2006.

[78] Thomas Stibor and Jonathan Timmis. An Investigation on the Compression Quality of aiNet. In *2007 IEEE Symposium on Foundations of Computational Intelligence*, pages 495–502, Honolulu, HI, USA, April 2007. IEEE.

[79] Julie Greensmith, Uwe Aickelin, and Steve Cayzer. Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomoly Detection, January 2005.

[80] Jacques Banchereau, Francine Briere, Christophe Caux, Jean Davoust, Serge Lebecque, Yong-Jun Liu, Bali Pulendran, and Karolina Palucka. Immunobiology of dendritic cells. *Annual Review of Immunology*, 18(1):767–811, 2000. PMID: 10837075.

[81] R. Oates, G. Kendall, and J. Garibaldi. Frequency analysis for dendritic cell population tuning: Decimating the dendritic cell. *Evolutionary Intelligence*, 1(2), 2008.

[82] Julie Greensmith and Uwe Aickelin. The Deterministic Dendritic Cell Algorithm, June 2010. arXiv:1006.1512 [cs].

[83] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[84] Zaineb Chelly and Zied Elouedi. Rc-dca: A new feature selection and signal categorization technique for the dendritic cell algorithm based on rough set theory. In *Proceedings of the 11th International Conference of Artificial Immune Systems, ICARIS*, pages 152–165, 2012.

[85] Zaineb Chelly and Zied Elouedi. Rst-dca: A dendritic cell algorithm based on rough set theory. In *Proceedings of the 19th International Conference on Neural Information Processing, ICONIP*, pages 480–487, 2012.

[86] Zaineb Chelly and Zied Elouedi. Qr-dca: A new rough data pre-processing approach for the dendritic cell algorithm. In *Proceedings of the 11th International Conference on Adaptive and Natural Computing Algorithms, ICANNGA*, pages 140–150, 2013.

[87] Lech Polkowski. *Rough Sets: Mathematical Foundations*. Physica-Verlag, 2002.

[88] Zied Chelly and Zied Elouedi. Supporting fuzzy-rough sets in the dendritic cell algorithm data pre-processing phase. In *Proceedings of the 20th International Conference on Neural Information Processing, ICONIP*, pages 164–171, 2013.

[89] Zied Chelly and Zied Elouedi. A fuzzy-rough data pre-processing approach for the dendritic cell classifier. In *Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU*, pages 109–120, 2013.

[90] Shaul Markovitch and Dan Rosenstein. Feature Generation Using General Constructor Functions. *Kluwer Academic Publishers*, 2002.

[91] Pablo Duboue. *The art of feature engineering: essentials for machine learning*. Cambridge University Press, 2020.

[92] A.K. Jain, P.W. Duin, and Jianchang Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

[93] C. O. S. Sorzano, J. Vargas, and A. Pascual Montano. A survey of dimensionality reduction techniques, March 2014. arXiv:1403.2877 [cs, q-bio, stat].

[94] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[95] Laurens van der Maaten, Eric Postma, and H. Herik. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research - JMLR*, 10, January 2007.

[96] Saul Lawrence K., Weinberger Kilian Q., Sha Fei, Ham Jihun, and Lee Daniel D. Spectral Methods for Dimensionality Reduction. In Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 292–308. The MIT Press, September 2006.

[97] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Machine learning proceedings 1992*, pages 249–256. Elsevier, 1992.

[98] Konstantin Hopf and Sascha Reifenrath. Filter Methods for Feature Selection in Supervised Machine Learning Applications – Review and Benchmark, November 2021. arXiv:2111.12140 [cs, stat].

[99] A. Jovic, K. Brkic, and N. Bogunovic. A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205, Opatija, Croatia, May 2015. IEEE.

[100] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita. Mifs-nd: A mutual information-based feature selection method. *Expert Systems with Applications*, 41(14):6371–6385, 2014.

[101] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proc. 20th International Conference on Machine Learning (ICML-2003)*, pages 856–863, Washington DC, USA, 2003. AAAI Press.

[102] H. Liu and R. Setiono. A probabilistic approach to feature selection-a filter solution. In *Proc. 13th International Conference on Machine Learning (ICML-1996)*, pages 319–327, Bary, Italy, 1996. Morgan Kaufmann.

[103] Vipin Kumar. Feature Selection: A literature Review. *The Smart Computing Review*, 4(3), June 2014.

[104] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proc. 15th International Conference on Machine Learning (ICML-1998)*, pages 82–90, Madison, Wisconsin, USA, 1998. Morgan Kaufmann.

[105] S. Maldonado, R. Weber, and F. Famili. Feature selection for high-dimensional class-imbalanced data sets using support vector machines. *Information Sciences*, 286:228–246, 2014.

[106] Y. S. Kim, W. N. Street, and F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6(6):531–556, 2002.

[107] J. C. Cortizo and I. Giraldez. Multi criteria wrapper improvements to naive bayes learning. In *LNCS*, volume 4224, pages 419–427, 2006.

[108] C. Liu, D. Jiang, and W. Yang. Global geometric similarity scheme for feature selection in fault diagnosis. *Expert Systems with Applications*, 41(8):3585–3595, 2014.

[109] S. Ma and J. Huang. Penalized feature selection and classification in bioinformatics. *Briefings in Bioinformatics*, 9(5):392–403, 2008.

[110] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[111] N. Abd-Alsabour. On the role of dimensionality reduction. *JCP*, 13(5):571–579, 2018.

[112] R. Aziz, C. Verma, and N. Srivastava. Dimension reduction methods for microarray data: a review. *AIMS. Bioengineering*, 4(1):179–197, 2017.

[113] A. S. Eesa, A. M. Abdulazeez, and Z. Orman. A dids based on the combination of cuttlefish algorithm and decision tree. *Science Journal of University of Zakho*, 5(4):313–318, 2017.

[114] Ali Ghodsi. Dimensionality Reduction A Short Tutorial. *Department of Statistics and Actuarial Science, University of Waterloo*, 2006.

[115] Mathieu Fauvel, Jocelyn Chanussot, and Jón Atli Benediktsson. Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas. *Journal of Advanced Signal Processing*, pages 1–14, 2009.

[116] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[117] D. Widjaja, C. Varon, A. Dorado, J. A. K. Suykens, and S. Van Huffel. Application of Kernel Principal Component Analysis for Single-Lead-ECG-Derived Respiration. *IEEE Transactions on Biomedical Engineering*, 59(4):1169–1176, April 2012.

[118] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, July 1998.

[119] David Charte, Francisco Charte, Salvador García, María J. del Jesus, and Francisco Herrera. A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78–96, November 2018. arXiv:1801.01586 [cs].

[120] W. Wang, Y. Huang, Y. Wang, and L. Wang. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 496–503, 2014.

[121] Yoshua Bengio. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[122] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[123] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative review. *J Mach Learn Res*, 10:66–71, 2009.

[124] Python Software Foundation. Python programming language – official website, 2024.

[125] Pandas Development Team. Pandas: Python data analysis library, 2024.

[126] NumPy Community. Numpy: The fundamental package for scientific computing with python, 2024.

[127] Matplotlib Development Team. Matplotlib: Visualization with python, 2024.

[128] Project Jupyter. Project jupyter: Open source tools for interactive computing, 2024.

[129] Digi2 Project Team. Digi2: Digital and embedded systems, 2024.

[130] Rui Pinto. M2m using opc ua, 2020.

[131] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, January 2018.

[132] Université Libre de Bruxelles (ULB) Machine Learning Group. Credit card fraud detection, 2016. Accessed: 2024-07-05.

[133] Fedesoriano. Heart failure prediction, 2021. Accessed: 2024-07-05.

[134] Fedesoriano. Stroke prediction dataset, 2021. Accessed: 2024-07-05.

[135] Prosper Chuks. Diabetes, hypertension and stroke prediction dataset, 2021. Accessed: 2024-07-20.

[136] Dhanush Narayanan R. Credit card fraud detection, 2021. Accessed: 2024-07-20.