U.PORTO
FACULDADE DE ECONOMIA
UNIVERSIDADE DO PORTO
FEP

Anomaly Detection in Pet Behavioural Data

Inês Pinto e Silva

Dissertation

Master in Modelling, Data Analysis and Decision Support System

Supervised by
Prof. João Manuel Portela da Gama
Prof. Rita Paula Ribeiro

2023

Acknowledgments

This dissertation represents the final step towards the completion of my studies in the Master's program in Modelling, Data Analysis and Decision Support Systems. It marks the culmination of my journey through the challenging yet immensely demanding path at Faculdade de Economia do Universidade do Porto.

I would first like to thank Maven for the challenging project, which I adored from day one, and for allowing me to explore new techniques in such an innovative context. Additionally, I would like to extend my sincere appreciation to Rui Correia, who played a crucial role as my mentor and guide during my time at the company.

I would like to express my deepest gratitude to João Gama and Rita Ribeiro, my supervisors, for their unwavering belief in my abilities, continuous guidance, and support throughout my journey. Their expertise, mentorship, and encouragement have been invaluable in shaping my research, refining my ideas, and overcoming challenges. I am truly grateful for their dedication and willingness to go above and beyond to help me succeed.

On a more personal note, I want to thank my close friends, especially the ones that have been a part of my path at Faculdade de Economia do Porto from day one, for all the moments shared and all the support.

I am also grateful to my family, especially my mother and sister, for being so supportive of my decisions and believing in me. All the love you gave me propelled me to bigger things. I will always be thankful for your continuous I am also grateful to my family for being so supportive of my decisions and believing in me. All the love you gave me propelled me to bigger things. I will always be thankful for your continuous encouragement.

Furthermore, I want to thank my number one supporter, Ricardo, for all the love, patience, kind words, assistance and presence in my academic, professional and personal life. Your companionship during moments of celebration as well as during the more challenging times has been invaluable in shaping my journey and bringing me to where I am today.

Last, but certainly not least a big acknowledgement is due to myself, for my determination, for consistently setting my sights higher and challenging myself. I am proud of the decisions I have made, even when they came at a cost.

Abstract

Pet owners are increasingly becoming conscious of their pet's necessities and are paying more attention to their overall wellness. The well-being of their pets is intricately linked to their own emotional and physical well-being. In this sense, different solutions for the veterinary system emerge to grant proactive healthcare alternatives to pets.

In this dissertation, pet behavioural data collected from an accelerometer sensor positioned on the neck of the pet was used. Furthermore, three unsupervised machine-learning techniques were exploited for anomaly detection, the best way to represent pet behaviour data for the specific task was researched and a solution for explaining the classification obtained by the model was proposed.

Specifically, aggregations by an average of Isolation Forest, Local Outlier Factor and K-Nearest Neighbour for different values of the most sensitive parameters were performed using different thresholds to separate abnormal from normal events. This approach was performed for different ways to represent data, concluding that the best way to detect anomalies is using daily data divided into periods.

Regarding model explainability, Shapley Values were suggested and a demonstration of the global (by an average of local analysis) to the best-performing model and best data representation was presented.

The performance varied across datasets and animals, with notable findings in anomaly detection. The Local Outlier Factor algorithm aggregation exhibited promise when analyzing daily data segmented into periods and aggregating results across all datasets, especially when prioritizing the identification of true anomalies over false positives. Moreover, the Isolation Forest aggregation demonstrated a remarkable balance between precision and recall trade-off, being the optimal choice for minimizing false alarms and exclusively detecting genuine anomalies.

Keywords: Anomaly Detection, Unsupervised Machine Learning, Local Outlier Factor.

Resumo

Os donos de animais de estimação estão cada vez mais conscientes das necessidades dos seus animais e dão cada vez mais atenção ao seu bem-estar. A qualidade de vida dos seus animais está intrinsecamente ligada ao seu próprio bem-estar emocional e físico. Nesse sentido, diferentes soluções para o sistema veterinário têm vindo a surgir, de forma a oferecer alternativas proativas de cuidados de saúde para os animais de estimação.

Nesta dissertação, foram utilizados dados comportamentais de animais de estimação recolhidos a partir de um sensor de acelerómetro posicionado no pescoço do animal. Além disso, foram exploradas três técnicas de aprendizagem automática não supervisionada para deteção de anomalias, no sentido de investigar a melhor forma de representar os dados de comportamento animal para o desafio em específico. Foi também apresentado uma solução para explicar a classificação obtida pelo modelo.

Especificamente, foram realizadas agregações pela média do *Isolation Forest*, *Local Outlier Factor* e *K-Nearest Neighbour* para diferentes valores dos parâmetros mais sensíveis, utilizando diferentes limiares para separar eventos anómalos de eventos normais. Esta abordagem foi realizada de diferentes formas para representar os dados, concluindo que a melhor forma de detetar anomalias é utilizar dados diários divididos em períodos temporais.

No que diz respeito à explicabilidade do modelo, foi sugerida a utilização de *Shapley Values*, e foi apresentada uma demonstração da análise global (através da média da análise local) para o modelo com melhor desempenho e melhor representação dos dados.

O desempenho variou entre os conjuntos de dados e os animais, com resultados relevantes na deteção de anomalias. A agregação do algoritmo *Local Outlier Factor* revelou-se promissora ao analisar dados diários segmentados em períodos de tempo e ao agregar resultados de todos os conjuntos de representação temporal dos dados, especialmente quando se prioriza a identificação de verdadeiras anomalias em detrimento de falsos positivos.

Além disso, a agregação do *Isolation Forest* demonstrou um notável equilíbrio entre a precisão e a taxa de verdadeiros positivos, sendo a escolha ideal para minimizar falsos positivos e detetar exclusivamente anomalias genuínas.

Palavras-chave: Deteção de Anomalia; Aprendizagem automática não supervisionada ; Local Outlier Factor.

Acronyms

AD - Anomaly Detection
ADL - Activities of daily living
AI - Artificial Intelligence
CBLOF - Cluster-Based Local Outlier Factor
DBSCAN - Density-Based Spatial Clustering of Applications with Noise
EAL - Evaluable Activity Level
FBAT - Fourier Based Approximation with Thresholding
FPR - False Positive Ratio
IF - Isolation Forest
IQR - Interquartile Range
K-NN - K-Nearest Neighbour
LDCOF - Local Density Cluster-based Outlier Factor
LIME - Local Interpretable Model-Agnostic Explanations
LOF - Local Outlier Factor
ML - Machine Learning
RB - Reference Behaviour
ROC - Receiving Operator Characteristic Curve
SD - Standard Deviation
SHAP - Shapley Additive Explanations
SVM - Support Vector Machines
TPR - True Positive Ratio
XAI - Explainable Artificial Intelligence

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the dissertation and presents the problem description. It provides an overview of the motivation behind the development of this project.

The problem at hand, which involves developing a methodology for detecting abnormal behaviour in pet data using unsupervised machine learning (ML) techniques, is presented.

The chapter concludes with an outline of the research questions and the structure of the dissertation.

1.1 Motivation

The Industrial Revolution was a period of significant economic development characterized by the introduction of new forms of manufacturing and transportation. The first Industrial Revolution, which began in the late 18th century in Great Britain, was marked by advancements in the textile industry and the use of steam power. The second Industrial Revolution, which occurred in the late 19th century, saw the rise of new forms of energy such as electricity and the widespread use of machines in factories. The third Industrial Revolution also called the Digital Revolution, began in the late 20th century and is characterized by the widespread use of computers and the Internet. These revolutions have led to significant changes in society, including urbanization and increased productivity, but also brought about social and environmental challenges.

As we enter the fourth industrial revolution, technological development has the potential of creating new opportunities to increase well-being in different aspects of our day-to-day life. Xu, David, and Kim (2018) mentions that there are five main opportunities from this revolution that stand out: "lower barriers between inventors and markets, more active role for the artificial intelligence, integration of different techniques and domains (fusion), improved quality of our lives (through robotics) and the connected life".

The development of technologies such as Artificial Intelligence (AI) and robotics is especially relevant in healthcare. Various publications enhance the importance of computational intelligence in diagnosis, biological image analysis, computer-aided surgery/ therapy and smart houses for senior citizens, a group of major concern. However, affordable devices available to the general population like wearables, such as smartphones and smartwatches, can provide relevant data that allows monitoring of an individual's physiological functions and can be used to predict, prevent, and treat public and individual health (Tana, Forss, and Hellstén (2017)).

These techniques focusing on human healthcare are now expanding to pet healthcare as most people's perception of animals has shifted drastically. One main component of this change was the Covid-19 pandemic. The need for companionship spiked the adoption rates as human-animal interaction helped with dealing with stress, anxiety, and depression due to isolation (Ho, Hussain, and Sparagano (2021)). According to Ho et al. (2021), a study based on Google Trends found that searches regarding pet adoption increased by 250% in the same period (April and May 2020), compared to the previous five years.

A new market for improving the health of animals and preventing possible complications has surfaced, with different companies already working in that sense, such as Whistle[1] and Fi [2], which create GPS tracking devices (the second one through collars) and software for pet owners, allowing them to track their pet's location, monitor their activity levels, and set up virtual boundaries, among other features. The second company also includes a two-way audio feature, allowing the owner to communicate with their dog remotely. Another example is Pawp[3] , which offers a solution of a digital health clinic and telehealth platform for pet owners and their pets that connects them with veterinary professionals.

The insurgence of companies that focus more and more on animal care and preventive pet healthcare with data obtained from wearables gains a new relevance as veterinary services, specifically in the US face issues as the number of pets has been increasing but the staff in the veterinary system has not followed that trend (AAVMC (2021)). At all levels of the system, this issue has been noticeable and has led to negative consequences such as "lack of vacancies for regular check-ups, emergency hospitals closing overnight, owners being referred hundreds of miles away for an elusive open spot, and dogs with true emergencies having to wait hours and hours to be seen", according to Moses, Malowney, and Boyd (2018).

The main focus of companies like this is proactive healthcare, i.e., preventing illness and promoting overall well-being, rather than treating illness after it has occurred. It involves maintaining good health and preventing disease through regular screenings, check-ups, healthy lifestyle choices and risk assessments.

Anomaly detection (AD) is an important step regarding this approach to health as it looks through data finding patterns that can be clinically relevant but invisible to the human eye.

One such company is Maven which provides the context and data for the present dissertation. Maven, launched in April of 2021 with the goal of giving a voice to pets and increasing their well-being and safety, emerged as a proactive healthcare-for-pets solution. It allows pet owners to track and monitor their pet's activity through a property collar, which is also analyzed by vets that can be contacted 24/7 to give advice or consultation.

Maven offers two devices, the "Maven Collar", which collects accelerometer data such as pet's activity and sleep, and the "Maven Home", which stores through Wi-Fi the data collected and enables the company to access it. A clinical team receives this data, which together with medical reports, allows them to have deep knowledge about the pet's health and needs and give

[1]Whistle, "Home," Whistle Website, www.whistle.com, accessed June 29, 2023.
[2]Fi, "Home," Fi Website, https://tryfi.com/, accessed June 29, 2023.
[3]Pawp, "Home," Pawp Website, https://pawp.com/, accessed June 29, 2023.

notifications to pet owners.

Maven's solution makes use of Artificial Intelligence techniques to prevent possible problems so that the vets can act on them.

Pet owners also have access to a mobile app where they can track the pet's activity, contact vets 24/7, receive updates or notifications when their pet's behaviour is out of the ordinary, personalized recommendations and weekly check-up meetings.

1.2 Problem Description

This dissertation was done in collaboration with the company Maven, with the goal of providing a solution that aligns with the company's objectives.

Maven seeks to offer innovative solutions to its clients, specifically targeting the prevention of health complications in pets by detecting anomalies in their behaviour. Such a solution would not only enhance the well-being of pets but also reduce associated expenses.

The problem at hand involves deriving a methodology from accelerometer data (collected through a proprietary collar), which provides a feature representing the level of activity of the pet.

This methodology must meet the requirements and limitations set by Maven and should enable the detection of abnormal behaviour in pet data through the utilization of unsupervised machine-learning techniques.

Additionally, an important aspect of this dissertation is to develop an explainable AI solution. It is crucial to find an approach that can provide insights into the model's decision-making process, allowing for a clear explanation of how anomalies are detected. To achieve this, explainable AI techniques will be employed.

By addressing these challenges, the dissertation aims to contribute to Maven's mission of providing proactive healthcare alternatives for pets and delivering a solution that can be both effective and interpretable in detecting abnormal behaviour in pets' data.

In this dissertation, the main goal is to address and find responses to the following research questions:

- **Question 1:** Is the data suitable for anomaly detection?
- **Question 2:** What is the best way to present the data in order to detect anomalies?
- **Question 3:** Are the techniques chosen adequate in detecting anomalies, for the specific study case?
- **Question 4:** How can the results obtained from the different algorithms be explained?

1.3 Dissertation Structure

This dissertation starts with the present Introduction and is then followed by Chapter 2, Related Work, which explores the existing literature and research related to the subject matter. It delves into various topics, such as animal behaviour and abnormal behaviour in animals, anomaly detection techniques, challenges of anomaly detection, the application of anomaly detection in animal behaviour and unsupervised machine learning techniques for anomaly detection. This chapter also discusses the concept of explainable AI and its importance in machine learning models, as well as presents the most common methods for the interpretability of models.

Chapter 3, Case Study, focuses on the specific business objectives and context of the research. It explains the relevance of the data sources used and outlines the requirements and limitations of the study. It also presents a detailed analysis of the data utilized in the study and highlights the key conclusions from the analysis. Moreover, it provides a comprehensive overview of the data preparation techniques applied, to ensure the data is appropriately formatted for subsequent analysis.

Chapter 4, Methodology, outlines the specific methodologies employed for anomaly detection. It introduces the Isolation Forest (IF), K-Nearest Neighbour (K-NN), and Local Outlier Factor (LOF) algorithms as the primary techniques used in the research. Each algorithm is described in detail, including the modelling process and implementation considerations. Additionally, this chapter addresses the concept of model explainability, particularly focusing on the application of Shapley values in the case study. Furthermore, it presents the experimental setup and describes the evaluation metrics used to evaluate the performance of the techniques applied.

Chapter 5, Obtained Results, presents the results obtained from applying the anomaly detection algorithms to the prepared datasets. It discusses the evaluation metrics used to assess the performance of the models and provides a detailed analysis of the data representation and algorithms' performance. Furthermore, this chapter examines the explainability of the models and their interpretability through the application of Shapley values.

Finally, Chapter 6, Conclusion, offers final remarks summarizing the main findings and contributions of the research. It also acknowledges the limitations of the study and suggests potential avenues for future research and development in the field.

Chapter 2

Related Work

In this chapter, a comprehensive review of the literature of the main topics to address in this dissertation is presented. First, the concept of abnormal behaviour in animals is explored (Section 2.1), followed by an explanation of anomaly detection (Sections 2.2.1 and 2.2.2). Secondly, there's a detailed analysis of anomaly detection techniques in both humans, with emphasis on accelerometer data (Section 2.2.3), and animal behaviour (Section 2.2.4). Thenceforth, an overview of the available techniques for unsupervised anomaly detection is examined (Section 2.2.5). Finally, the topic of Explainable AI for ML models is delved into (Section 2.3) emphasizing the importance of explainability in AI, the taxonomy for ML interpretation and the techniques available for ML explainability.

2.1 Animal Behaviour and Abnormal Behaviour in Animals

Understanding animal behaviour has long been a topic of interest for the general population and as so the field of Ethology, has gained popularity being now a well-established scientific discipline. The importance of animal behaviour derives from the knowledge that how an animal interacts and responds to a certain environment or situation is crucial to gaining information on animal needs, requirements, preferences, and dislikes.

This information is of main importance to focus on three main components of animal well-being: "maintaining physical health or physiological normality; preventing or reducing illness, fear, stress, pain, or tension and providing pleasure, comfort, or satisfaction.", Mench (1998). In this sense, detecting abnormal behaviours is key to understanding pet behaviour and potentially identifying early warning signs of health issues or other problems.

2.2 Anomaly Detection

Anomaly Detection, according to Chandola, Banerjee, and Kumar (2009), "refers to the problem of finding patterns in data that do not conform to the expected behaviour". The nonconforming patterns are usually denominated anomalies and outliers. The importance of the detection of anomalies, according to Nassif, Talib, Nasir, and Dakalbab (2021), "concerns the risk that abnormal data may represent significant, critical, and actionable information".

Understanding and detecting anomalies can have applications in various domains: fraud detection, loan application processing, monitoring of medical conditions, cyber security intrusion detection, fault detection for aviation safety study, streaming, hyperspectral imagery, and more (Chandola et al. (2009); Koren, Koren, and Peretz (2023); Nassif et al. (2021)).

To select the best AD technique some factors must be taken into consideration: the nature of the data (quantitative, qualitative or mixed), if labels to classify the data are already defined and the type of anomaly that may be encountered.

2.2.1 What is an anomaly?

Types of Anomalies

Anomalies are patterns that are outside of the range of the ordinary and can be divided into various different types Chandola et al. (2009); Fahim and Sillitti (2019); Habeeb et al. (2019); Nassif et al. (2021); Sgueglia, Sorbo, Visaggio, and Canfora (2022):

- Point Anomaly: This type of anomaly occurs when single instances of the data are far from the rest of the data, i.e., that instance is anomalous considering the remainder of the data. That instance is the point anomaly or also known as an outlier.

- Contextual Anomaly: This type of anomaly occurs when an instance is normal in a certain context but abnormal in another. Usually common in time-series data.

- Collective Anomalies: This type of anomaly occurs when a sequence of related (collective) observations is anomalous concerning an entire dataset.

- Global Anomalies: Global anomaly is an anomaly that spans the whole dataset or a significant section of it. It affects a sizeable section of the data and constitutes a major departure from the predicted trends.

- Local Anomalies: This type of anomaly occur within a specific region or neighbourhood of the dataset. They are instances of anomalous behaviour within localized subsets of the data, considering the behaviour of neighbouring instances.

Data Labels

As mentioned, one of the crucial points in applying appropriate Anomaly Detection methods is the existence or not of labels to classify an observation as an anomaly or not. According to the existence or not of these labels, Boukerche, Zheng, and Alfandi (2020); Chandola et al. (2009); Koren et al. (2023) highlight three types of Anomaly Detection techniques available:

- Supervised Anomaly Detection: When training data to find patterns out of common in a supervised way, the existence of labeled normal and abnormal data is necessary. The training data, already classified, is used to predict the label for new instances.

- Unsupervised Anomaly Detection: This type of anomaly detection does not require training data, as it works with the assumption that anomalies are not nearly as frequent as normal instances.

- Semisupervised Anomaly Detection: In semisupervised learning, the only classified data are the normal instances and so the techniques used focus on understanding the patterns of normal data and identifying as anomalies the observations that do match with those patterns.

2.2.2 Approaches for Anomaly Detection

Techniques for Anomaly Detection

Multiple literature review papers and surveys written by the community already exist that discuss and present various techniques to detect anomalous data (Chandola et al. (2009); Fahim and Sillitti (2019); Nassif et al. (2021); Niu, Shi, Sun, and He (2011); Sgueglia et al. (2022)). From these, it is possible to distinguish three main fields that offer techniques for anomaly detection: Statistical Methods, Machine Learning algorithms, and domain-specific heuristics. The choice of method will depend on the nature of the data and the type of anomalies being sought.

Statistical Methods

Statistical methods for anomaly detection typically involve identifying data points significantly different from most of the data. The main goal is to build a statistical model for the regular instances and then a statistical inference is carried out to test if a new instance is likely to be a part of the model constructed. If the probability of the new instance being part of the learned model is low, then that observation will be classified as an anomaly. The methods used to conduct this type of anomaly detection are proximity-based, parametric, non-parametric, and semi-parametric techniques.

- Parametric: Gaussian Model-Based, Regression Model-Based, Mixture of
  Distributions-Based.

- Non-Parametric: Histogram-Based, Kernel Function-Based.

Machine Learning Algorithms

Machine learning algorithms, on the other hand, can be trained to recognize patterns in the data and identify unusual observations. These methods can be effective at detecting complex or previously unseen anomalies, but they require a large amount of labeled training data and may not be suitable for all types of data. Regarding the usage of ML algorithms, the most used methodologies can be grouped into the following types of techniques:

Classification-based techniques: In classification, models are trained using input/labelled data and then, with the classes learned during the training, the model will classify new instances (as normal or anomaly). To apply classification the assumption that it is possible to distinguish between normal and abnormal observations and that it can be learned by a classifier has to hold. Among classifiers, there are two ways to learn and detect an outlier: Multi-class techniques, in which the model learns and is able to identify different classes (the anomalies) and distinguish them from all normal classes, and One-class classification in which all training data is considered as only one class and from that, a discriminative boundary where new instances will have to be a part of in order to be considered normal is learned. The different types of classification algorithms most used are Neural-Networks, Bayesian- Networks, Support Vector Machines(SVM) and Rule-Based Classifiers.

Nearest neighbours Techniques: These types of techniques focus on the assumption that normal observations will occur in the same neighbourhoods, while anomalies happen far from those neighbourhoods. The distance or similarity between observations is computed using different metrics to measure the closeness to neighbourhoods. The methods used to detect anomaly when using nearest neighbours can be of two types:

- using the distance to the kth nearest neighbour as the anomaly score;
- using the relative density of each data instance to compute its anomaly score.

Clustering-based Techniques: These techniques consist in grouping similar data instances. By clustering seemingly alike observations, different types of methods to detect anomalous instances exist: The ones based on the assumption that similar data instances can be grouped together, and anomalies will belong to a different group/cluster. (ex: Density-based spatial clustering of applications with noise (DBSCAN), ROCK, SNN Clustering). The ones that for each instance the distance to its cluster centroid is computed as its anomaly score. (ex: K-means Algorithm and Expectation Maximization). The ones in which a threshold is defined and if the instances in a certain cluster are of a size or density below that threshold then the cluster contains only anomalous instances. (Ex: Cluster-Based Local Outlier Factor(CBLOF)).

Other approaches to anomaly detection include **domain-specific heuristics**, which are rules or guidelines based on expert knowledge of the data and the types of anomalies that may occur. Information Theoretic techniques and Spectral Anomaly techniques can also be used, however, these are not so common due to their disadvantages.

Challenges of Anomaly Detection

Although the concept of anomaly detection may initially appear deceptively straightforward, a comprehensive examination reveals the presence of various factors that significantly augment the complexity of this procedure. As mentioned before, the type of data available, the nature of the data, the type of anomalies and the availability of labels for the data are all aspects that influence the approach to anomaly detection for a specific problem formulation. Each combination of factors comes with its own challenges.

- Definition of an anomaly: Defining what is considered normal or abnormal and the line that distinguishes them is complicated since this differentiation may not be strictly precise.

- Susceptibility to Fraud: Anomalies can result from negative intentions (for example, fraud). In that case, the aim is to make the anomaly seem as normal as possible, thereby creating difficulty in defining what is considered normal.
- Defining anomaly in evolving domains: Different domains characterize anomalies differently, making it harder to create a general approach to detect anomalies. Besides that, in some domains, what may be seen as an outlier is constantly evolving and changing, so a constant update is needed.
- Lack of data and difficulty on understanding the data: The availability of data for training the models or validating them is also one of the major difficulties. Also, regarding data, one issue faced is the difficulty of distinguishing noise from anomalies.

2.2.3 Anomaly Detection for Accelerometer Retrieved Data

In this dissertation, the data to detect anomaly detection from pet behaviour is collected from a property collar and retrieved from an accelerometer sensor. In that sense, it is important to explore the existing literature on anomaly detection from acceleration data. Accelerometers are used in a variety of domains with different purposes and so, when trying to identify anomalies in this type of data, different authors use different strategies. The domains in which accelerometers are most used, according to Krichen (2021), are Healthcare Monitoring, and Traffic and Roads Monitoring.

Most studies in detecting anomalies from accelerometer data, mainly in the healthcare domain, get the tri-axial data from mobile sensors and usually consider more than just acceleration data. Focusing on healthcare monitoring, one of the main concerns regards the increase of life expectancy and the need to prevent health issues of the elderly. Khan and Hoey (2017) reviewed some fall detection methods and concluded that falling should be considered as an abnormal activity and also suggested the usage of an auto-encoder or Recurrent Neural Network to detect this type of anomaly. Medrano, Igual, Plaza, and Castro (2014) presented a system to detect falls, and to prevent injuries. The falls are in this case seen as anomalies to detect and use accelerometer data obtained via smartphone to do so. The authors experimented with a machine learning approach on a one-class classifier only trained on regular activities of daily living(ADL) to detect falls. The methods used were the k-Nearest Neighbour and a two-classes Support Vector Machine, having the second option presented the best results. Albert, Kording, Herrmann, and Jayaraman (2012); D.-S. Huang et al. (2006)) also use Support Vector Machines to detect falls, considering them as anomalies.

Authors such as Lee and Carlisle (2011) take a different approach to preventing falls, instead of using ML techniques, the chosen method is based on a threshold. Chehade, Ozisik, Gomez, Ramos, and Pottie (2012) suggested a semi-supervised approach in which the model is only trained for normal walking patterns and the anomaly will occur in instances that deviate from the pattern defined by the model. The density by Gaussian distribution is computed from the training data for each user and when the value estimated is lower than a defined threshold then there's a presence of an anomaly.

Micucci, Mobilio, Napoletano, and Tisato (2017) reviewed different methods for detecting falls and experimented with one-class and two-class K-Nearest Neighbour as well as one-class and two-class Support Vector Machine and compared the results for the four techniques. The classifiers were trained with only regular activities of daily life and tested with both activities of daily living and fall instances. For all of them, a test instance was considered an anomaly (fall) when its score was higher than a defined threshold (different thresholds were used for better analysis). The metrics used to evaluate the different algorithms were receiving operator characteristic curve (ROC), the area under the ROC curve (AUC), Sensitivity, and Specificity. Regarding the K-NN, the Euclidean distance was used, to choose the ideal number of neighbours of k a 10 cross-validation was used and the values of k that were experimented with were 1-10. For the SVM, in each fold of the 10-cross validation, an inner 10-cross validation was performed for choosing the best regularization and kernel parameters. The results obtained indicate that the two-classifier SVM is the best model, however, the difference among the classifiers was very small.

Mahfuz, Zulkernine, and Nicholls (2018) use data from the accelerometer and gyroscope, after feature extraction, to detect anomalies using a Deep Neural Network of four layers (two of them hidden). The proposed method presented 98.75% accuracy for binary classification (fall and not fall).

Mental Health is another specific area in Healthcare Monitoring in which accelerometer data is used, as stated by Tron, Resheff, Bazhmin, Weinshall, and Peled (2018).

D'Mello, Melcher, and Torous (2022) aimed to "explore a method of aligning time series data captured from personal smartphones to detect abnormalities in behaviour related to mental illness". Different types of sensors were used, however, to detect anomalies from the accelerometer data a specific method was developed. From the raw data, the jerk is computed, and the "Temporally-aligned Similarity" technique developed in the article is applied. This approach consists in considering a temporal rule R in which partitions of the time series will be created according to mutual time characteristics. After applying this rule, an L subseries group has been created and for each group (G) there are subseries with a user-specified time resolution under 24 hours. For each G a matrix is created to measure pairwise similarity (the similarity measure used is Dynamic Time Warping) and the columns with the highest dissimilarity indicate the occurrence of an anomaly.

Another mental health usage of an accelerometer is watching that tracks patients' walking patterns with Schizophrenia. In this case, Tron et al. (2018) suggested the usage of an AutoRegressive Integrated Moving Average Model to detect anomalies in the behaviour of the patient. An "ARIMAX(1,1,1) Seasonal (1,1,2)" model based on the previous 7 days was used to predict the pattern of the following day. Abnormal behaviour is considered to occur when the predicted value is not in the Confidence Interval of the model when the residuals between model prediction and observed values are higher than the threshold and the certainty (computed using the confidence interval and standard deviation (SD) of the data) of the model is lower than the threshold. The results suggest that when clients were given a certain medication or had some type of episode, this seemed to be shown by the forecasting model as an anomaly.

When considering actigraphy ("objective measurement method that assesses limb movement activity via a small recording device typically worn on the wrist" (Edinger, Means, Carney, and

Manber (2011))) data monitoring of humans,Fuster-Garcia, Juan-Albarracín, Bresó, and García-Gómez (2013) suggested using a Daily Activity Monitoring System (DAMS) based on functional data analysis algorithms for signal alignment and non-linear dimensionality reduction techniques based on manifolds, that allows for detecting unusual behaviour for each daily activity signal an anomaly measure based on the nearest neighbour analysis. The anomaly score of observation is defined as its distance to kth nearest neighbour as anomalies occur far from their closest neighbours. This study considered the number of neighbours (k) to be fours to avoid activity patterns that happen once a week being considered abnormal.

Focusing on the actigraphy data monitoring of humans and grasping one already mentioned concept, Georgakopoulos et al. (2022) investigated the application of an autoencoder to extract features from time series data, after being characterised as a specific activity. Using the non-linear transformations performed by the autoencoder, the extracted features are then applied to a distance-based statistical technique (Mahalanobis-Minimum Covariance Determinant Distance) selected to detect anomalies. When the distance is larger than the criterion (extension of the median plus or minus a coefficient times the Median Absolute Deviation), the instance is considered an anomaly.

Using an unsupervised technique based on physical activity measurements, according to Candás et al. (2014) for abnormal behaviour detection is a real-time solution for anomaly detection without human intervention. The authors suggest a data mining statistical-based technique that does not assume anything about the data distribution to overcome the limitations of other techniques. The algorithm used is divided into three steps: activity level difference detection, statistical features extraction and abnormal behaviour detection. In the first step, the difference between the reference behaviour (RB) and the raw evaluable activity level (EAL) is computed (the variation between the evaluable activity level and the expected level according to the calculated reference behaviour). A median filtering is used to eliminate the noise of the difference. In the second step, the moving standard deviation of the RB and EAL are computed. A low moving standard deviation indicates a neighbourhood with a stable activity level and a high moving SD indicates an unstable activity level. Therefore, the moving SD represents the stability of the activity level for each sample. The abnormal human behaviour detection step compares the activity difference and the moving standard deviations of the reference and the evaluable activity level and uses them to build a fuzzy valuation function. The fuzzy valuation function, which gives an abnormal behaviour value, is based on a trapezoidal membership function with values from -1 to 1. If the absolute value of the first step is higher than the maximum SD calculated, the sample is considered abnormal; If the absolute value is lower than the maximum SD calculated but higher than the minimum, the abnormality of the sample value depends on the relation to the reference and evaluable neighbourhood. If the absolute value of AD is lower than the minimum SD calculated the sample is considered as normal.

2.2.4 Anomaly Detection in Animal Behaviour

Anomaly detection in animal behaviours remains an area that warrants further comprehensive investigation. Nevertheless, select researchers have already contributed to the existing literature, shedding light on this subject.

When considering the usage of the accelerometer to detect anomalies,Waele et al. (2021) introduced the application of an unsupervised machine-learning algorithm to detect changes in mares' behaviour that can be important to identify the onset parturition. This method presented a 100 % accuracy in detecting foalings in the dataset, however, a high number of false alarms occurred. The accuracy when the threshold was altered decreased to 80%, but also the number of false alarms was significantly reduced. The algorithm proposed is an unsupervised autoencoder-based anomaly detection algorithm trained on regular horse behaviour, followed by a dynamic threshold to make the final decision when classifying an instance. Two one-dimensional convolutional layers, which perform feature extraction automatically are used in both the encoder and decoder. When the autoencoder faces an abnormal behaviour, it will not be able to reconstruct the behaviour, presenting then a higher reconstruction error (mean squared error). If the reconstruction error is above the defined threshold (defined specifically for each mare), then the window of time will be classified as anomalous.

Wagner et al. (2020) use an indoor tracking system that classifies dairy cows' behaviour into three activity levels (eating, resting, and in alleys). The data was labeled and in the division of the data for training and testing half of the abnormal instances were considered in each division. The algorithm proposed was the Fourier Based Approximation with Thresholding (FBAT), however, Dynamic Time Warping, Bag of SFA Symbols, Hive-Cote and ResNet were also performed for comparison. The FBAT classifies time series considering that an anomalous series will include a break in its cycle, and as so if the variations in cyclic components are high the series is seen as an anomaly. The algorithm extracts two sub-series and their harmonic decomposition creating a new model with these harmonics. During the extraction, the models are delayed, so a shift (computed as a distance) to the model must be applied. The distance reflects the cyclic component, and the higher it is, the higher the variation. If the distance is bigger than a defined threshold then the time series is concluded to be abnormal.

Wagner et al. (2020) conducted a study to detect changes in the behaviour of cows submitted to "Sub-Acute Ruminal Acidosis", using other cows not submitted to the disease as control cows. The same type of activity level distinction as the one applied to the previously mentioned study was used. The algorithms used were K Nearest neighbours for Regression (KNNR); Decision Tree for Regression (DTR); MultiLayer Perceptron (MLP); Long Short-Term Memory (LSTM); and an algorithm where activity is assumed to be similar to one day to the next. The algorithms aimed to predict the following day's activity, considering the prior 24h. The error between the predicted and observed values was then compared to a threshold to distinguish abnormal from normal values. KNNR performed best, detecting 83% of SARA cases (true-positives), but it also produced 66% of false positives.

2.2.5 Unsupervised Anomaly Detection Techniques

Unsupervised Machine Learning for anomaly detection methods can be divided in: Nearest-neighbour-based techniques, Clustering based methods and Statistical algorithms (Dataman (2023); Goldstein and Uchida (2016)).

Regarding the Nearest-neighbour techniques, it is possible to distinguish global anomaly detection and local anomaly detection.

In what concerns finding global anomalies the most common algorithm used is the k-nearest-neighbour global unsupervised anomaly detection in which the k-nearest neighbours are found and then an anomaly score is computed using the neighbours either through the distance to kth-nearest neighbour or through the average of distances to all k-nearest neighbours.

Goldstein and Uchida (2016) performed a comprehensive evaluation of 19 different unsupervised anomaly detection algorithms in 10 different datasets and was able to conclude that nearest neighbour-based algorithms tend to perform better than clustering techniques when the goal is outlier detection. Furthermore, it highlights the global K-NN algorithm as the best candidate for anomaly detection in its category of techniques, specially when the type of anomalies to be found are global.

In a recent study, 52 real-world multivariate tabular datasets were used to assess 32 unsupervised anomaly detection algorithms, the most comprehensive comparison of unsupervised anomaly detection algorithms to date. From this study, the authors were able to conclude that the k-thNN (distance to the k-nearest neighbour) algorithm was found to outperform most other algorithms in this collection of datasets (Bouman, Bukhsh, and Heskes (2023)).

By utilizing statistical distributions, in Steinbuss and Böhm (2021), the authors suggested a novel method for combining anomalies. They evaluated the effectiveness of four well-known anomaly detection methods using various datasets created from 19 basic datasets. The k-Nearest Neighbour, Isolation Forest, and Local Outlier Factor algorithms were among those being studied. The results of the study showed that k-NN and Isolation Forest algorithms performed better than others at spotting global anomalies. These algorithms successfully located anomalies that considerably differed from the trends found across the dataset. On the other hand, the Local Outlier Factor method showed extraordinary effectiveness in detecting local anomalies and dependence abnormalities, i.e., anomalies that are distinguishable by their closeness to other data points or their connections to other variables.

When focusing on finding local anomalies, different options are available such as Local Outlier Factor (LOF), Connectivity-Based Outlier Factor, Influenced Outlierness and Local Outlier Probability. However, the most well-known is LOF in which an anomaly is found by computing the k-nearest neighbours, computing the local reachability density and creating a LOF anomaly score that considers the local reachability density of a record and its k-nearest neighbours. Records with lower local density will have a higher anomaly score and will be considered anomalies.

Campos et al. (2016) applied multiple nearest neighbours-based techniques on 11 different datasets to evaluate their performance, concluding that Local Outlier Factor significantly outperforms most available techniques.

A common use of LOF is credit card fraud detection, John and Naaz (2019) applied both Local Outlier Factor and Isolation Forest to a data correspondent to credit card transactions during a specific month in Europe. The authors highlighted Local Outlier Factor as a better fit than the Isolation Forest for the application of the study.

Besides credit card fraud, LOF has also been applied to medical fraud. Bauder, da Rosa, and Khoshgoftaar (2018) focused on detecting Medicare fraud, which contributes to rising healthcare costs, using the Medicare Part B dataset. The paper aims to use unsupervised machine learning for the specific context. Isolation Forest, Unsupervised Random Forest, Local Outlier Factor, autoencoders, and k-Nearest Neighbours are exploited in this sense. The List of Excluded Individuals/Entities (LEIE) database is used to validate the performance of each method. Results showed that Local Outlier Factor performs best, while k-Nearest Neighbours, autoencoders, and Isolation Forest perform poorly.

Regarding the clustering-based methodologies, it is possible to highlight global anomaly detector models, for example, Clustering Based Outlier Factor, in which the anomaly score for each data point is determined based on its distance from the cluster's centroid after the data have been initially clustered into subgroups.

Ester, Kriegel, Sander, and Xiaowei (1996) introduced, a density-based clustering algorithm named DBSCAN that deals with large datasets with noise and that can identify clusters having arbitrary shapes and sizes. Clusters with a high density of points are treated as normal clusters, whereas regions with a low density indicate noisy clusters. However, the parameter setting is the main issue with DBSCAN and it does not work well for varying-density clusters.

He, Xu, and Deng (2003), introduced a new Cluster-Based Local Outlier Factor approach named CBLOF. A novel cluster-based local outlier factor approach that overcomes the limitations of traditional clustering algorithms in outlier detection. By considering cluster size and distance to the nearest cluster. CBLOF aims for a balance between optimizing clusters and accurately identifying outliers.

There are also local anomaly detectors cluster-based, such as Local Density Cluster-based Outlier Factor (LDCOF) that estimates the clusters' densities assuming a spherical distribution of the cluster members. Similar to CBLOF, the proposed approach starts with k-means clustering. Clusters are then separated into small and large clusters. The LDCOF score is calculated for each cluster by comparing the distance of an instance to its cluster centre with the average distance of all cluster members. This score provides a local measure considering varying cluster densities.

Statistical methods, as already mentioned, can also be used with some of the most common ones being Histogram Based Outlier Detection, Gaussian Mixed Models, Empirical Cumulative Outlier Detection, and dimensionality reduction using Principal Component Analysis.

Besides these three main groups of techniques, it is also possible to identify the usage of classification-based techniques, as already mentioned.

Another relevant technique, specifically in unsupervised learning is density-based anomaly detection. These methods lack the need of specifying the number of clusters or labels, rather, they use a density threshold or a neighbourhood size to define the density level. These techniques have the benefits of being resilient to noise and outliers and handling arbitrary cluster shapes and sizes. However, a decision on the neighbourhood size or the density threshold, which may depend on the size and distribution of the data, is needed and directly impacts the models' performance. Some of the models in other groups of unsupervised ML techniques, already mentioned, such as LOF and DBSCAN can also be included in the density-based techniques for anomaly detection.

One very common density-based approach is Isolation Forest, which stands out for its numerous real-life applications and for providing good results when compared to other density-based approaches.

Ahmed, Lee, Hyun, and Koo (2019) proposed the usage of Isolation Forest to detect Cyber-Physical Data Integrity Attacks (CDIAs) in smart grid communications networks. They also used a Principal Component Analysis(PCA)-based feature extraction mechanism to handle the complexity of power systems. The approach outperformed other machine learning schemes in terms of detection accuracy.

Zhong et al. (2019a), proposed an unsupervised anomaly detection method for gas turbine gas path using Isolation Forest. It makes use of two versions of Isolation Forest. The proposed method achieved excellent anomaly detection performance on real OEM data from CFM56-7B aero-engines. According to the authors, it has the advantage of not requiring labelled or abnormal data, as well as it also being able to handle small datasets and continuous data effectively, making it suitable for gas turbine anomaly detection.

2.3 Explainable AI and Machine Learning Explainability

A significant rise in the popularity and progress of Artificial Intelligence has been noticed in the past decade. This occurrence originates from the increase in the adoption of machine and deep learning algorithms to solve complex problems. Furthermore, it also leads to an even bigger increase in the usage of AI algorithms as the results obtained show promising results that can have a direct impact on various levels of our daily life.

However, these achievements are also accompanied by an increase in model compatibility and less transparency, originating the need for a solution to understand the results obtained by using AI. In that sense, Explainable Artificial Intelligence (XAI) emerged as a way to bring more clarity into the field (Saeed and Omlin (2023)).

According to Gunning and Aha (2019), XAI is a solution for the lack of transparency in the AI field that mainly focuses on developing and creating techniques that allow the interpretation of AI models and "that empower end-users in comprehending, trusting, and efficiently managing the new age of AI system".

Explainable AI encompasses considerations of confidence, safety, security, privacy, ethics, fairness, and trust (Kieseberg, Weippl, and Holzinger (2016)). Usability is also an important aspect that needs to be addressed in XAI research. All these factors are crucial for the applicability of AI in healthcare, particularly in the context of personalized healthcare.

2.3.1 Importance of Explainable AI

According to Bhattacharya (2022), the need and relevance of being able to explain Machine Learning models focus on four main topics.

The first topic concerns the verification and debugging of machine learning systems. In this first topic, various benefits can come from explainable AI.

- Transparency and Interpretability: In the sense that explainability techniques allow transparency in the decision-making of ML systems, which translates to a better understanding, for all members evolved, (from the stakeholders to end-users) of the reason and the why of a certain prediction.

- Error Diagnosis: ML systems may occasionally produce unexpected or incorrect results. In such cases, explainable AI techniques serve as tools for diagnosing and troubleshooting errors.

- Model Validation and Performance Assessment: Explainable AI makes it easier to validate and evaluate ML models. By gaining insight into the inner workings of the system, developers can evaluate the performance of the model, identify any deviations or limitations, and ensure that it behaves as expected under various circumstances. This validation process ensures that ML systems meet the required standards of accuracy, fairness, and ethical considerations.

- Fairness and Bias Detection: Explainable AI can help identify and mitigate bias in ML systems and ensure fairness and impartiality by analyzing decision-making patterns.

- Trust and Accountability: Explainable AI builds trust by enabling stakeholders to understand system decisions and increases accountability through a transparent decision-making process, which helps identify and resolve ethical or legal issues.

- Continuous Improvement: With explainable AI, developers can identify areas of improvement for ML models, leading to iterative development and improving the robustness and reliability of the system over time.

The second topic of interest relates to the capacity of including human experience, intuition and user-centred strategies to enhance ML solutions. By providing insights into model decisions, it encourages transparency, trust, and acceptance on the part of users. It facilitates user feedback, aids in the correction of biases, and encourages iterative development, all of which contribute to more dependable and user-friendly ML systems.

Thirdly, the capacity of XAI to give new insights of knowledge by uncovering the reasoning behind the experiences and examples got by the model is one of the drivers of XAI's importance. Users and developers gain valuable knowledge about the model's operation, the reasons behind certain decisions, and the underlying factors that drive those decisions thanks to this deeper comprehension.

Lastly, and gaining more importance every day, Explainable AI is crucial for compliance with legislation. It ensures transparency, accountability, and fairness in AI systems, helping organizations meet legal requirements, mitigate biases, and build trust.

All those topics can be included in the FAT model of explainable AI, as suggested by Masís (2021).

2.3.2 Machine Learning Explainability Taxonomy

To methodically sort and comprehend the different scopes of strategies and procedures utilized in machine learning explainability, some authors suggest the usage of a taxonomy (Gopinath (2021)).

The various methods for interpreting and explaining machine learning models are organized and categorized within this taxonomy, which serves as a structured framework. The taxonomy helps researchers and practitioners navigate the field and select suitable methods based on their specific requirements by defining various dimensions and categories.

According to Gopinath (2021), the taxonomy for ML explainability should focus on four main dimensions.

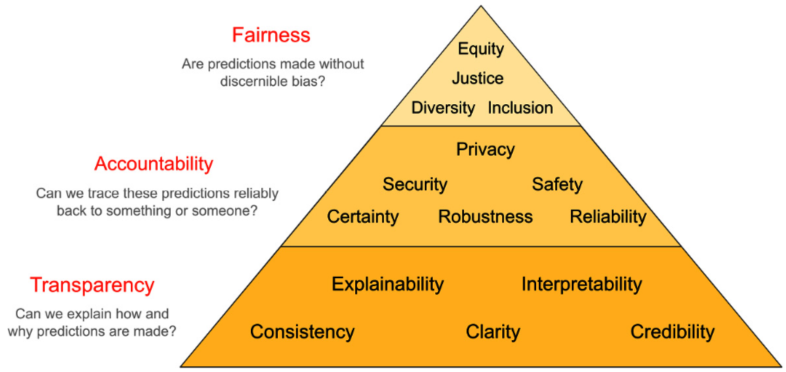

Figure 2.1: FAT Model of Explainable ML (from Interpretable Machine Learning with Pyhton by Serg Masís)

The first dimension corresponds to the **explainability scope**, that is what are the outputs that we aim to explain. The explanation can be global, in which the goal is to explain the model by analyzing the underlying factors that contribute to its predictions across multiple data points,i.e., understanding the important features that drive the model predictions. On the other hand, the explanation to achieve can also be local, that is, elucidating the factors that drive a prediction for an individual data point. There is also the chance that the scope of explainability is a middle point between global and local, in order to elucidate the factors driving a model's predictions for a specific population segment, such as women, for example.

Another point of consideration in the explanation scope is the type of output we aim to explain, in the sense that the explanation is determined not only by the number of outputs but also by their type. For example, when explaining a loan eligibility model, we must consider whether we are explaining probability scores or classification decisions. The choice between these options has significant implications for the explanation itself, as it affects the focus and interpretation of the explanations provided.

Secondly, we need to consider the **explanation inputs**, this is the type of inputs that the explanation is being computed from. Interpretability/explainability methods can focus on different aspects of the model and data. The most common approach is to interpret the input features and analyze the impact of each feature on the model's decision. For example, in healthcare, certain characteristics such as "blood pressure" and "family history" can be identified as influential. Another approach is to interpret intermediate features, which is useful for models such as convolutional neural networks. Instead of interpreting individual pixels in an image, interpretations are generated for intermediate levels (layers) that capture patterns that indicate, for example, the presence of a dog. Additionally, some techniques aim to justify model behaviour by attributing it to the training data itself. These methods quantify the data points that contribute most to some aspect of the model's learned behaviour. Overall, the techniques used can focus on explaining input features and intermediate features, or trace back to training data to provide insight into the model's decision-making process.

Thirdly, the **explanation access**, which regards the information that the explanation knows about the model in use is one of the four crucial points in this taxonomy.

In this sense, it is possible to have Model-agnostic techniques like Local Interpretable Model-Agnostic Explanation (LIME) and Shapley Values that assume limited access to the model's inputs and outputs, without knowledge of its internal architecture. These techniques focus on the effect of inputs on model outputs, making them useful for comparing explanations across different models trained on the same dataset.

On the other hand, there are interpretation techniques for specific model classes. These techniques use the structure of the model to improve performance.

In extreme cases, there are model-specific techniques which may require full access to model objects. Although these techniques are not easily transferable to different model classes, they can provide deep insights and performance improvements for specific model types. Gradient-based strategies, such as Integrated Gradients, SmoothGrad, and Grad-CAM developed for neural networks, are examples of such model-specific techniques.

The last point in Gopinath (2021) taxonomy is the **explanation stage**.

Methods of explanation can be used before, during, or after model training. Self-interpretable models, such as linear regression and decision trees, offer logical justifications based on the relative relevance of the features and predetermined criteria during training. But if these models are used exclusively, machine learning applications' versatility is constrained.

Alternately, after training both self-interpretable and algorithmically interpretable models, post-hoc explainability approaches can be used. Frequently referred to as "black box" models, algorithmically interpretable models generate rules during training that may not be immediately clear. Post-hoc explainability is relevant to a wide variety of machine learning models since it enables the explanation of these models.

Some authors, besides these four main points, highlight also the speed versus accuracy trade-off, computational costs and artificial intelligence governance, risk management and compliance needs as other points of consideration when opting for an explainability technique.

There is also literature that opts on focusing only on dimensions such as if the technique should be model-specific or model-agnostic techniques. The interpretability should be intrinsic or post-hoc, and the explanations should be provided at a local or global level (Barredo Arrieta et al. (2020)).

2.3.3 Explainable AI Techniques for Machine Learning

Transparent Models

In order to present which techniques are available for model explainability, it is important that a first distinguishing is done. Some models, known as transparent models or white box models, are understandable by themselves, either through simulatability, decomposability or algorithmic transparency (Arrieta et al. (2020)).

- Simulatability refers to a model's capacity to be simulated or comprehended solely by a human, emphasizing the role of complexity.

- Decomposability refers to the ability to explain the individual parts of a model (input, parameter, and calculation), which enables the understanding, interpretation, and explanation of the model's behaviour. Every component must be understandable without additional tools for an algorithmically transparent model to be decomposable.

- Algorithmic transparency can be perceived in various ways, involving the user's comprehension of the model's process to generate outputs from input data. Algorithmically transparent models are constrained by their mathematical exploitability and analytical methods.

Some examples of transparent models are Linear/Logistic Regression, Decision Trees, K-Nearest Neighbours, Rule Based Learners, General Additive Models and Bayesian Models.

However, according to Arrieta et al. (2020), most times the models can get too complex, due to various facts related to specific model parameters or data that post-hoc techniques end up having to be utilized.

Post-hoc Explainaiblity

For models that are not readily interpretable (black box models), there is a need to resort to post-hoc techniques, such as text explanations, visual explanations, local explanations, explanations by example, explanations by simplification and feature relevance explanations techniques. In this case, the methods used aim to explain the predictions of the model. Among the post-hoc techniques, there are model-agnostic and model-specific methods.

Model-Specific Techniques: Model-specific techniques, i.e., tools that are used to interpret models with specific features and potential, for ML models usually are performed by using (Gohel, Singh, and Mohanty (2021)):

- Feature Relevance: Understanding the most impactful features in the decision-making process. One very common method to interpret the model based on feature relevance is Feature Importance - which shows the impact factor of each feature in derived decisions (Rajani, McCann, Xiong, and Socher (2019)).

- Condition-based Explanation: Condition-based explanations involve justifying predictions or outcomes based on specific conditions and observed inputs. By asking "Why?" oriented questions, the model generates all possible explanations with corresponding conditions. Completeness phenomena ensure a comprehensive set of conditions is considered. "What if" scenarios allow for hypothetical reasoning and counterfactual justifications. A logical model converts user inputs into constraints and determines if they are satisfied, providing justification. This approach aims to systematically analyze conditions and constraints for a comprehensive explanation.

- Rule-based learning: By translating insights into rules, full transparency can be achieved for explainable AI (XAI). Framing rules for all predictions makes even complex neural network models transparent, allowing customers and naive users to comprehend the results easily.

For ML algorithms, two models that require additional explanation stand out: Tree ensembles (random forests and multiple classifier systems) and Support Vector Machine (SVM).

Tree ensembles are models that aggregate multiple decision trees to improve generalization and reduce overfitting. However, the combination of models in tree ensembles makes their interpretation more complex than individual decision trees.

Some techniques for explaining tree ensembles include explanation by simplification and feature relevance techniques.

Explanation by simplification involves creating a simplified model from a set of random samples labelled by the ensemble model or using two models (one for interpretation and one for prediction). An example of this is the Simplified Tree Ensemble Learner (STEL), which turns a complex tree ensemble into a rule-based learner. The ensemble method averages over the variance of multiple models, which in turn deprives the interpretations of individual models (Dwivedi et al. (2023)).

Feature relevance techniques, analyze variable importance within random forests by measuring the impact on accuracy when a variable is randomly permuted. An example of such a method is measuring the Mean Decrease Accuracy (MDA) or Mean Increase Error (MIE) of the forest when a certain variable is randomly permuted in the out-of-bag samples (Auret and Aldrich (2012); Breiman (1984)). Another example is the *tree interpreter*, which decomposes the prediction results to a sum of feature contributions and bias.

Support Vector Machines (SVMs) build hyperplanes in high-dimensional space to maximize the margin between the hyperplane and the nearest training data points. SVMs are well-known for their capacity to predict and generalize, but they require explanatory approaches to be interpretable. Explanation by simplification, local explanations, visualizations, and explanations by example are post-hoc explainability strategies used with SVMs.

The most common techniques used to explain SVM are (Barredo Arrieta et al. (2020)):

- Rule extraction, including modified sequential covering algorithms, fuzzy rule extraction, and the creation of hyper-rectangles from the intersections between support vectors and the hyperplane.

- Model Simplification, including adding the SVM's hyperplane and support vectors to rule creation, clustering methods to group prototype vectors, and creating non-overlapping rules as a multi-constrained optimization problem.

- Visualization Techniques, such as visualizing the kernel matrix to extract information content or using heatmaps.

- Bayesian Interpretation: SVM models can be interpreted as Maximum A Posteriori (MAP) solutions to inference problems with Gaussian Process priors. This approach allows for tunable hyper-parameters and the prediction of class probabilities instead of binary classification.

Model- Agnostic Techniques Model-Agnostic Techniques are techniques that can be applied to any model, and as so are usually preferred over Model-Specific methods, since it allows for the development of different models for the same problem.

In mode-agnostic explainability, two different ways of dividing the available techniques exist Global Model-Agnostic Methods and Local-Agnostic Methods.

Global Model-Agnostic Methods, describe the average behaviour of a machine-learning model and are often expressed as expected values based on the distribution of the data (**?**.

Most of these methods focus on the features of the models, the most relevant being:

- Partial Dependence Plot (PDP): A visualization technique used to analyze the marginal effect of one or two features on the predicted outcome of a machine learning model, that can be used for both regression and classification problems. It aims to show what the model predicts on average when each instance has a different value for the specific feature. A flat PDP indicates an unimportant feature and greater variation in the PDP indicates more relevance of the feature. The feature importance measure calculates the deviation of unique feature values from the average curve.

- Accumulated local effects: explain how features influence the prediction of a machine learning model on average. These plots are usually a faster and more unbiased alternative to partial dependence plots (PDPs). It aims to show how the model predictions change in a small "window" of the feature around a certain value for data instances in that window.

- Permutation Feature Importance: Based on the approach suggested by Breiman (1984), it consists in measuring the importance of a feature by computing the leverage in the model's prediction error after permuting the feature. A feature is "important" if shuffling its values increases the model error and is less relevant when doing the same does not affect the error.

Another common technique for Global Model-Agnostic explanations is the global surrogate a simplified, interpretable model that surrogates the predictions of a complex black box model. The surrogate model is trained using the same input parameters as the original model, however with a simpler architecture such as a linear model or decision tree. It assists us in understanding the way the original model makes predictions and the connections between input features and predictions.

Local Model-Agnostic Methods aim to explain individual model predictions, as opposed to Global Model-Agnostic Methods.

In regards to this type of technique, it is possible to highlight the following as the most relevant (Dhinakaran (2021); Dwivedi et al. (2023); Gopinath (2021); Y. Huang (2021); Molnar (2023)):

Local Interpretable Model- Agnostic Explanations (LIME): A technique for local interpretability that aims to explain the decisions taken by a model for a single input by perturbing the features of that data instance and observing how these perturbations affect the model's predictions. LIME approximates the local behaviour by training a simpler and more interpretable model, the "local surrogate." The local surrogate model aims to capture the decision boundary of the black-box classifier specifically for the local instance of interest.

According to, Dhinakaran (2021) LIME can contribute to understanding the importance of different features in explaining the model's local predictions, even in cases where the model's overall accuracy is not perfect.

It should be noted that LIME does not ensure a full understanding of the black box model as a whole, although it can provide valuable insight into local behaviour. It provides only explanations in respect of the particular situations and characteristics which are considered as a result of the perturbing and approximation process.

Counterfactual Explanations: This technique is usually applied to binary datasets, however, it can also be used for, classification problems with multiple target values. A counterfactual explanation simulates small changes to the input features of certain instances to analyze how it can result in a different prediction or outcome.

Identifying the minimal changes required to affect the prediction, such as reaching a certain threshold or flipping the predicted class, are what is aimed to uncover these scenarios.

To create counterfactual explanations, we must first choose an instance to be explained and establish the intended outcome. From there, a tolerance parameter must be set, and the loss function must be optimized to locate the appropriate counterfactual explanation. This optimization procedure could require raising a parameter's value until a satisfactory solution within the targeted tolerance is reached.

Generating counterfactual explanations can be an arduous task, particularly for models that are complex and have high-dimensional features.

While counterfactual explanations may offer valuable insights, it is crucial to acknowledge that they are not without their faults and may not entirely grasp the model's functionality. Also, the explanatory effectiveness and understandability of counterfactuals pivot on the data's accuracy, how the model is structured, and the underlying assumptions used throughout the explanation phase.

Scoped Rules (Anchors): This method allows explanations for individual predictions by trying to find decision rules (known as anchors) so precise that changes in other feature values do not affect the target value.

Anchors explore the neighbourhood of the instance being explained by creating and evaluating perturbations, allowing it to be model-agnostic and applicable to any type of model. Matching the predicates of an anchor's neighbours and instance, the precision of the anchor is measured through model predictions. The aim is to uncover a high-precision rule with significant coverage across input space by surpassing a specific threshold.

It takes advantage of reinforcement learning techniques, graph search algorithms, and probabilistic definitions to generate scorable IF-THEN rules that capture the model's decision-making process, which allows for lower model calls and a better ability to bounce back from local optima.

This approach includes four main components that generate candidates and identify the best ones based on precision and coverage. These components are candidate validation, candidate generation, and best candidate identification.

As with any explanation based on perturbations, using the anchor algorithm entails a range of obstacles. Every domain requires a thorough correction, such as the configuration of hyperparameters and development adjustment functions that are individually tailored to it. Discretization is often necessary to maintain the boundaries of the decision, but if it is not done correctly, it can lead to confusion. Since this algorithm relies on the model's effectiveness, its performance is erratic. It is also difficult to define the coverage of a given field, e.g. image analysis, since this isn't always apparent. These drawbacks are highlighted by the complexity and domain-specific nature of perturbation theory explanations, reminding us that careful implementation is needed.

Shapley Values - based techniques: The Shapley Values concept originated in cooperative coalitional game theory as a solution whose focal point was understanding how cooperation between groups of players (coalitions), contributed to the overall success of these alliances.

Shapley Value-based explanation techniques, such as Shapley Additive exPlanations (SHAP) make use of this concept to provide exact mappings for the model's output scores or classification decisions for each input.

The Shapley values, which correspond to the average of all the marginal contributions to all possible coalitions, are the basis for SHAP explanations in the context of explainable artificial intelligence. In the context of explaining predictions, Shapley values can help determine how much each feature value has contributed to a particular outcome of an ML model compared to the average prediction. It is used to explain local predictions of complex ML models, however, what makes this technique stand out is that it also allows global application. This means that it can be used to explain not only specific predictions but also overall model behaviour.

Shapley values are a valuable tool for ML models, answering several critical questions. Firstly, they provide explanations for the specific outputs generated by the model for given inputs. Secondly, they allow comparisons between an individual's case and others, highlighting the factors that contribute to the prediction differences. Furthermore, they attribute the model's decisions to particular features, enabling a deeper understanding of their contribution. Lastly, Shapley values assign numerical values to each feature, quantifying their relative importance in the model's predictions.

These techniques offer a rigorous approach to understanding the behaviour and decision-making process of ML models with clear interpretation.

The Shapley value is computed as the average marginal contribution of a feature value across all possible coalitions. These values not only reveal the relevance of each feature but also indicate whether a feature has a positive or negative influence on the predictions. For example, in a fraud model, the features "purpose" and "loan amount" might have a significant impact on determining whether the model predicts fraud. In order to determine the Shapley value for one feature, we have to assess its contribution in various coalitions using simulated combinations of feature values. This means that, in order to make predictions, it is necessary to randomly select another subset of the dataset and use its feature values in combination with the feature value of interest. We can estimate its marginal contribution by comparing the predictions with and without a feature value of interest. The end goal is to explain the difference between the actual prediction and the average prediction by determining the contribution of each feature value.

Opposed to LIME, which shows an overall model prediction, SHAP elaborates on individual features that contribute to the prediction. SHAP measures the variation between the model's predictions with or without a specific feature's value. Although it emphasizes the contribution of each feature to the end result, it doesn't directly address the impact of input changes on predictions.

2.4 Discussion

In this chapter, a detailed explanation of what anomaly detection is was presented and a description of the techniques exploited for anomaly detection in different scenarios with emphasis on unsupervised machine learning techniques.

The literature on anomaly detection from acceleration data , collected from accelerometer sensors, shows a variety of approaches and domains. In the healthcare monitoring domain, the authors mainly focus on fall detection using machine learning techniques such as k-Nearest Neighbour, Support Vector Machine, and Deep Neural Networks. The two-class SVM was found to be the best model. Another application of accelerometer data in healthcare monitoring is for mental health, where the authors explore methods to detect abnormalities in behaviour related to mental illness, using techniques such as Temporally-aligned Similarity and ARIMAX models.

Anomaly detection in animal behaviour is a developing field. Unsupervised autoencoder-based machine learning, the FBAT algorithm, and KNNR are the most commonly used in this sense. KNNR and unsupervised autoencoders performed best, however, presented a high percentage of false positives.

Unsupervised anomaly detection techniques offer effective approaches for anomaly detection without labelled data. Nearest-neighbour-based methods, such as k-nearest neighbour and Local Outlier Factor excel in detecting global and local anomalies, respectively. Clustering-based methods, like Cluster-Based Local Outlier Factor, strike a balance between optimizing clusters and identifying outliers. Statistical algorithms provide robust anomaly detection using techniques like Histogram Based Outlier Detection and Gaussian Mixed Models. Density-based methods, including Isolation Forest, handle arbitrary cluster shapes and sizes and have shown promising results. The choice of the best-performing model depends on the dataset and the nature of the anomalies. Evaluating multiple techniques is recommended for selecting the most suitable approach for a specific anomaly detection task.

Besides the main focus of anomaly detection, an extensive review of state-of-the-art methods regarding the explainability of ML models was explored. Explainability in machine learning can be achieved through various techniques. Transparent models, which have algorithmic transparency, are by themselves understandable. On the other hand, post-hoc techniques aim to explain predictions of black box models. These methods include text explanations, visual explanations, and feature relevance, among others.

Additionally, interpretability techniques can be model-specific or model-agnostic. Model-specific techniques are tailored to interpret models with specific characteristics, such as tree ensembles or support vector machines (SVMs).

On the other hand, model-agnostic techniques can be applied to any model. They include global methods that provide insights into the average behaviour of a model, as well as local methods that explain individual predictions. Some popular techniques in the field of interpretability include LIME, counterfactual explanations, scoped rules, and SHAP. These methods offer different approaches to understanding machine learning models, providing valuable insights into predictions, feature importance, and the decision-making processes of these models.

In the context of this dissertation, the contextual anomalies can be global, when the pet health is affected for a long period of time, or local, specific circumstances that affect the pet's well-being (such as an injury or a stomach bug). For these reasons, the unsupervised techniques were used to tackle both types of anomalies: K-Nearest neighbour and Isolation Forest to detect global anomalies and Local Outlier Factor to study local anomalies.

To provide insights into the significance of each feature, aiding in understanding their impact on detecting anomalies in pet data, Shapley values were utilized. The option for this technique was due to its capacity of determining feature importance as a model-agnostic approach.

Chapter 3

The Pet Behaviour Case Study

In this chapter of the dissertation, the focus is on discussing several key aspects. Firstly, the chapter provides an overview of Maven's goals, data sources, and the requirements and limitations of the project (section 3.1).

Following that, the chapter delves into the data description section (3.2), which provides a detailed description of the collected data, including its processing algorithm, and presents an exploratory analysis of the activity levels for different animals. In this subsection, the answer to the first research question is provided.

Lastly, the data preparation and tools section (3.4) outlines the steps taken to prepare the data for analysis and organizing datasets based on time ranges and derived features. It also highlights the tools used in the project.

3.1 Business Understanding

This section provides an overview of the company's functioning goals and context, followed by a brief discussion of the data sources and relevance. Finally, the requirements imposed by the company in the development of this project are presented.

3.1.1 Business Objectives and Context

As covered in the introduction of this dissertation, it is clear that for multiple reasons, pet owners prize their pets in a way never seen before. In that sense, there is a rising need in the pet care market for creative solutions that promote pet well-being and proactive health care.

Maven understands this opportunity and intends to position itself as a proactive provider of pet healthcare solutions. Maven aspires to differentiate itself in the industry and provide a full solution to pet owners by employing innovative technologies such as Artificial Intelligence.

The company's emphasis on combining data from the Maven Collar and Maven Home devices with medical reports is connected to being able to provide intimate knowledge of pets' past and present to gain insights and a full picture of a pet's health, which a standard in-person veterinary is not able to give.

Maven's goals focus on enhancing pet well-being by providing pet owners access to real-time activity tracking, behaviour analysis, personalized recommendations, and preventing and addressing pet health issues.

However, there is also the aim to establish Maven as a trusted and innovative healthcare solution, to foster strong relationships with pet owners by offering a mobile app and regular check-up meetings to increase customer satisfaction and loyalty and to differentiate itself in the competitive pet care market by offering a comprehensive, data-driven healthcare solution.

The anomaly detection in pet behavioural data project aligns with Maven's vision of being a proactive provider of healthcare solutions for pets and with pet owners' needs and current reality.

Maven can enhance its business solution by identifying abnormal behaviours or patterns by implementing anomaly detection techniques. This helps pet owners, and veterinarians take timely action to prevent potential health issues.

The project contributes to Maven's overall goal of providing comprehensive, data-driven healthcare by leveraging advanced technology to proactively improve pet well-being and address health issues.

3.1.2 Data Sources and Relevance

Maven's solution originates from collecting data through a property collar, a device that collects accelerometer and gyroscope data. The data obtained through the collar is then transmitted via Wi-Fi to *Maven Home*, a device that stores the data collected by the *Maven Collar*. It is a central repository for the pet's behaviour data, allowing easy access and analysis.

For this project, the data from the accelerometer, stored in *Maven Home*, was processed through a company's specific algorithm that transforms the information of each 15-second window in a feature corresponding to the activity level. The activity level can be: excited, active or inactive.

Besides that, medical records created by the company's veterinaries, regarding the animals approached in the study were provided to evaluate the proposed solution.

3.1.3 Requirements and Limitations

For the specific project development of this dissertation, some requirements and limitations were imposed by the company.

Regarding the requirements, it is possible to highlight:

- Exclusion of Time-Series or Statistical Solutions: The company expressed a preference for machine-learning approaches rather than time-series or statistical-focused solutions. This decision likely stems from the desire to leverage the capabilities of Artificial Intelligence techniques in the solution it provides to its customers.

- Emphasis on Machine Learning Approaches: Maven specifically requested machine learning techniques for the anomaly detection project. This requirement highlights the company's interest in leveraging the power of machine learning algorithms to analyze and identify abnormal patterns in the data.

- Exploration of Different Data Organization Techniques: The company desired the experiments to incorporate different ways of organizing the data in time.

- Explanation of Anomalies: The company also required that the solution include a way to understand why an anomaly occurred. This requirement is because after an anomaly is flagged, there will be a veterinary that will try to understand if action is required.

Focusing on the limitations of the project, the following can be mentioned:

- Consideration of Computational Limitations: The company preferred to avoid deep learning or complex machine learning alternatives due to limited computational resources.

- Unsupervised Machine Learning Approach: The absence of labels for the project led to the adoption of an unsupervised machine learning approach.

- Limited Data Availability: The available data only consisted of a single categorical feature for each animal, with values ranging from 0 to 2. This limited feature set represented a challenge in the anomaly detection task, as there was the need to create new features to provide a robust solution.

- Unstructured and Incomplete Medical Records: Another project limitation was the unstructured nature of the provided medical records. These records sometimes lacked updates specifically related to when the animal exhibited deviant activity. Furthermore, the dates provided in the records were not precise enough.

It is important to note that despite these limitations, the anomaly detection project aimed to make the best use of the available data and unstructured medical records, to develop a robust unsupervised machine learning approach for detecting anomalies.

3.2 Data Description

This section provides a detailed description and an exploratory analysis of the data used in the experimental study are provided.

3.2.1 Data Understanding

The data was collected from an accelerometer, that is inserted in a property collar, of a company for different animals and different periods. The data retrieved from the accelerometer is measured in the 3-axis with a sample rate of 25 Hz and a full scale of 8G. It is then processed with a specific algorithm created by the company. This algorithm is composed of two groups of operations: sample operations, operations at the sample level which manipulate the data using physics knowledge by doing a norm computation, and window operations, operations that transform a stream of samples into a 15-second window. Lastly, the algorithm divides the data into three activity levels: inactive as 0, active as 1 and excited as 2. After the algorithm processing, for each of the pets, only the days with information for all windows of 15 seconds were considered.

3.2.2 Exploratory Data Analysis

First pet - Dog: The dog considered is 2 years and 9 months old, female, weighs 20.9 kg (ideal body condition) and is an Australian cattle dog. It has no chronic conditions, spends most time indoors, has high usual activity and walks daily.
The plot below 3.1 shows the total dispersion of 15-second windows for each activity level.

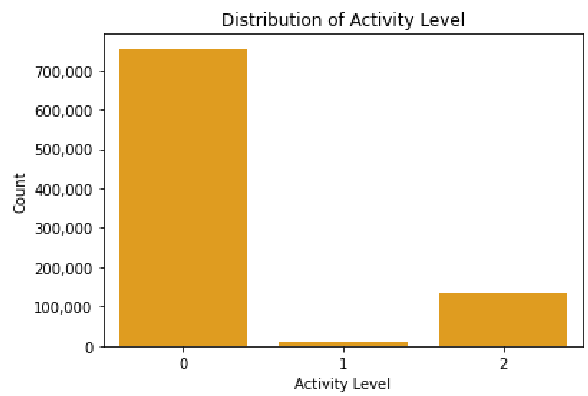

Figure 3.1: Bar plot of activity level for all 15-second windows of the 156 days.

Most time of the animal is spent inactive (83.99% of the 156-day period) with an average of 20 hours, 9 minutes and 23 seconds a day with an activity level equal to 0.
After time inactive, time excited follows with an average of 4 hours,47 minutes and 31 seconds per day of activity level equal to 2, corresponding to 14.87% of the 156-day period.
Finally, the pet spends an average of 16 minutes and 28 seconds of the day active, with only 1.14% of the 156-day period having an activity level equal to 1.

It is possible to analyze per day of the week the average time for each activity level. The image 3.2, obtained from the data, suggests that Mondays, Thursdays, and Saturdays are less active (more time inactive and less time active), while Tuesdays and Fridays are more exciting (with less time active and more excited time, more significantly Tuesdays).

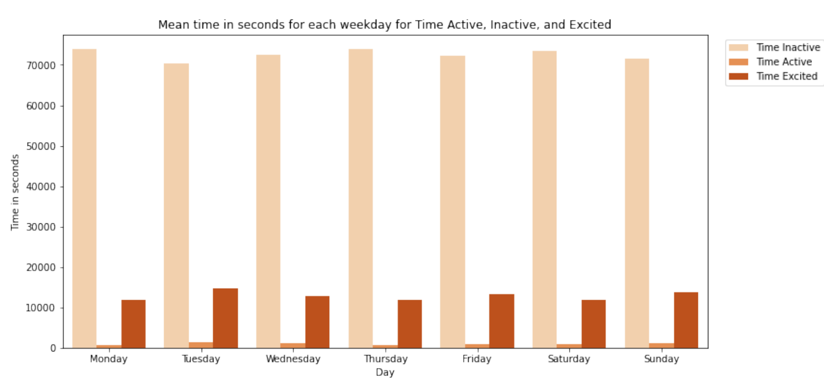

Figure 3.2: Mean time in seconds for each weekday for Time Active, Inactive, and Excited.

After this first analysis, using the interquartile range and the box plot, some days can be considered outliers. Regarding time excited and inactive, the same 32 days could be distinguished as outliers (all of them extreme). For time active, 33 days are extreme outliers (one more than the ones detected for the other variables).

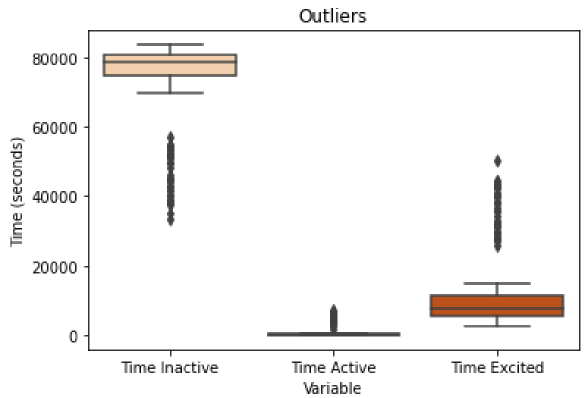

Figure 3.3: Box plot of Time Active, Time Inactive and Time Excited per day.

Second pet - Dog2: The dog considered is 4 years and 5 months old, female, weighs 9.1 kg (ideal body condition) and is a Shiba Inu dog. The animal has chronic pancreatitis, spends most time indoors, has low usual activity and does not walk daily.
For this animal there are a total of 151 days of complete data.
The plot below 3.4 shows the total dispersion of 15-second windows for each activity level.

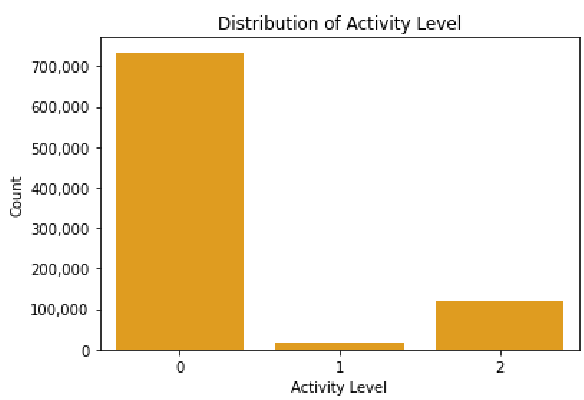

Figure 3.4: Bar plot of activity level for all 15-second windows of the 151 days.

Most time of the animal is spent inactive (84.4% of the period) with an average of 20 hours, 15 minutes and 58 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 03 hours, 18 minutes and 32 seconds per day of activity level equal to 2, corresponding 13.8% of the period. Finally, the pet spends an average of 25 minutes and 28 seconds of the day activities, with only 1.77% of the period considered having an activity level equal to 1.

The data, presented in 3.5, suggests that on Wednesdays, Thursdays, and Fridays, there is generally more time spent inactive. Wednesdays and Thursdays tend to have less active time overall, while Fridays see an increase in activity compared to other weekdays. Additionally, Wednesdays and Thursdays are days that typically the animal is less time excited, while Saturdays and Mondays tend to show more time spent excited.

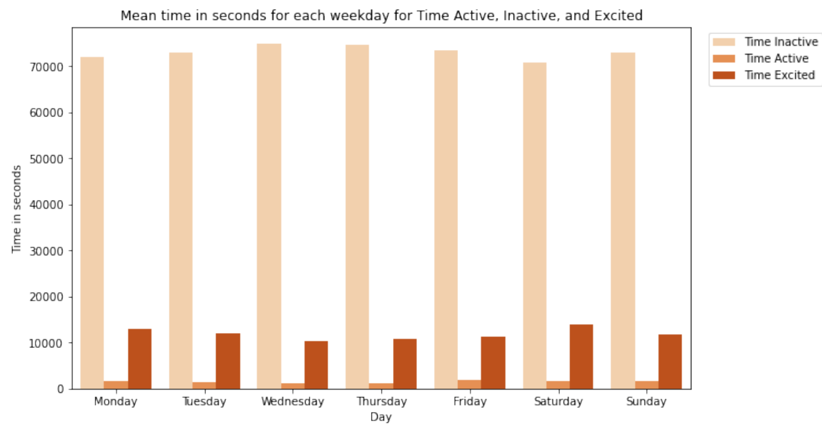

Figure 3.5: Mean time in seconds for each weekday for Time Active, Inactive and Excited

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered outliers. Figure 3.6 shows the Box plot of Time Active, Inactive, and Excited considering the daily data from the 151-day period which allows to the conclusion that there were no days considered as abnormal when using this technique.

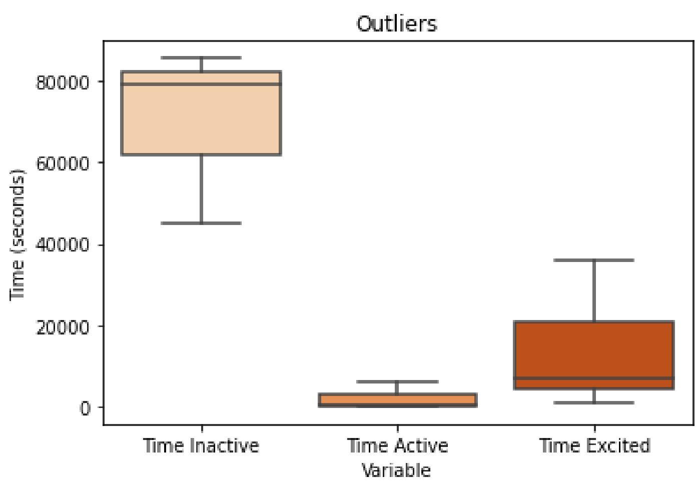

Figure 3.6: Box plot of Time Active, Inactive and Excited considering the daily data from the 151-day period.

Third pet - Dog3: The dog considered is 2 years and 4 months old, female, weighs 16.1 kg and is a Mixed Breed - Medium sized. It has chronic skin allergies, spends most time outdoors,

has low usual activity and walks daily.
For this animal there are a total of 108 days of complete data.
The plot below 3.7 shows the total dispersion of 15-second windows for each activity level.

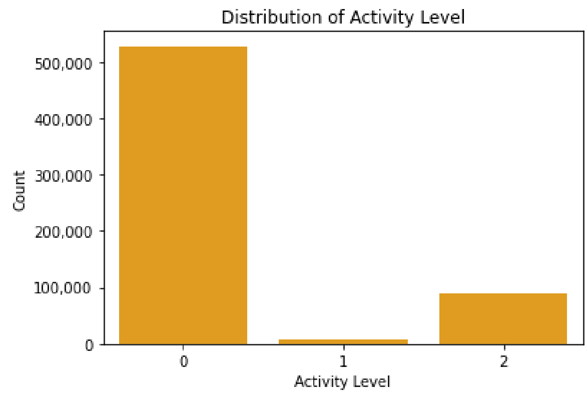

Figure 3.7: Bar plot of activity level for all 15-second windows of the 108 days.

Most time of the animal is spent inactive (84.9% of the period) with an average of 20 hours, 22 minutes and 38 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 03 hours, 23 minutes and 3 seconds per day of activity level equal to 2, corresponding to 14.1% of the period. Finally, the pet spends an average of 14 minutes and 17 seconds of the day active, with only 0.99% of the period considered having an activity level equal to 1.

The data, presented in 3.8, suggests that Wednesdays and Thursdays are characterized by a higher amount of inactive time, meaning there is less overall activity. Moreover, Mondays and Sundays tend to have relatively less active time compared to other days. However, Tuesdays stand out as days with more active time than usual. In terms of excitement, Wednesdays and Thursdays are generally less thrilling. Conversely, Mondays and Saturdays are known for having a higher level of excitement compared to the rest of the week, with Saturday being also the day that the pet spends less time inactive.

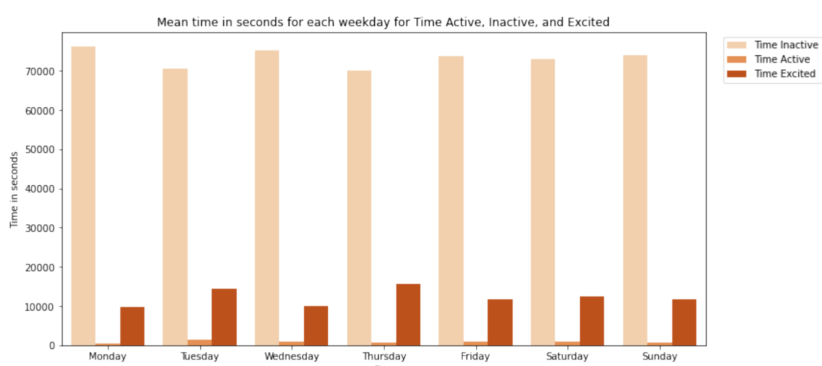

Figure 3.8: Mean time in seconds for each weekday for Time Active, Inactive and Excited.

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered outliers. Figure 3.9 shows the Box plot of Time Active, Inactive, and Excited considering the daily data from the 108-day period. Regarding time inactive and time excited the same 23 days were identified as outliers (60.86% of them occurred from Tuesdays to Thursdays). For time active two more days were considered outliers, one of them being on a Wednesday and the other on a Monday, which leads to the conclusion that 70% of the outliers occurred in the period of Tuesdays to Thursdays.

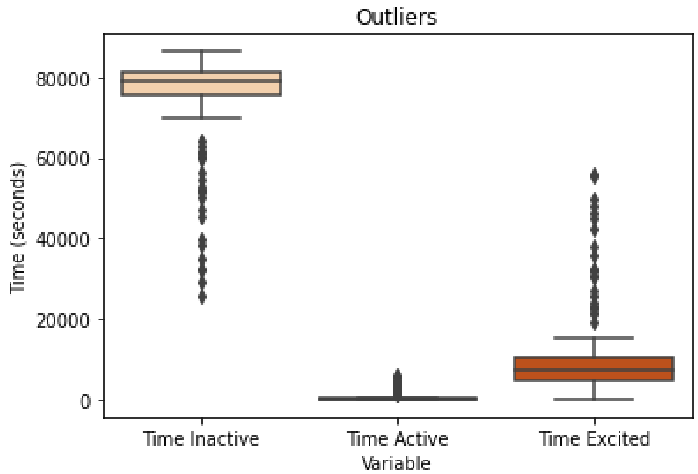

Figure 3.9: Bar plot of Time Active, Inactive and Excited considering the daily data from the 108-days period.

Fourth pet - Dog4: The dog considered is 3 years and 10 months old, female, weighs 13.7 kg (ideal body condition) and is a Mixed Breed - Medium sized. It has no chronic conditions, is always indoors, has low usual activity and walks daily. For this animal, there are a total of 192 days of complete data.

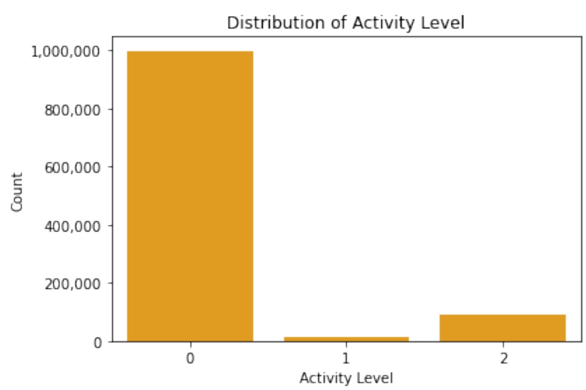

Figure 3.10: Bar plot of activity level for all 15-second windows of the 192 days.

The plot above 3.10 shows the total dispersion of 15-second windows for each activity level.

Most time of the animal is spent inactive (90.25% of the period) with an average of 21 hours, 39 minutes and 35 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 02 hours and 48 seconds per day of activity level equal to 2, corresponding 8.39% of the period. Finally, the pet spends an average of 19 minutes and 36 seconds of the day active, with only 1.3% of the period considered having an activity level equal to 1.

The data, presented in 3.11, suggests that Tuesdays and Thursdays are typically characterized by a higher amount of inactive time, indicating less overall activity. On the other hand, Fridays tend to have less active time compared to usual, while Mondays and Sundays stand out as days with more active time. In terms of excitement, Tuesdays and Fridays are generally less thrilling. In contrast, Mondays and Saturdays are known for being more exciting than other days of the week.

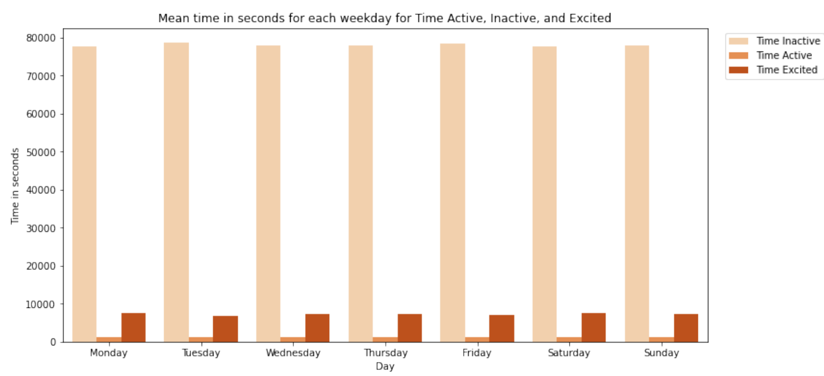

Figure 3.11: Mean time in seconds for each weekday for Time Active, Inactive and Excited.

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered outliers. Figure 3.12 shows the Box plot of Time Active, Inactive, and Excited considering the daily data from the 192-day period. For time inactive four days were identified as outliers, two of them appearing in the outliers of time inactive and the other two in the ones of time excited. Regarding time inactive and excited 8 days for each were found to be outliers. However, the six for each that were not connected to time inactive were unique to the corresponding variable. In time active the most unusual days were Sundays corresponding to 42.86% of the outliers. In time excited the most unusual days were Fridays and Saturdays corresponding together to 50% of the outliers.

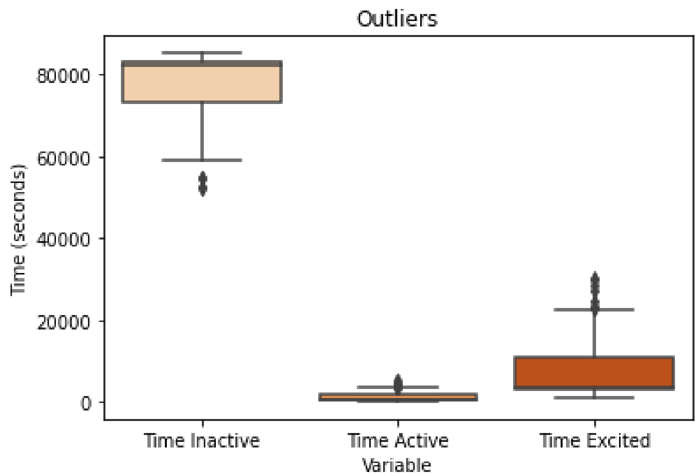

Figure 3.12: Box plot of Time Active, Inactive and Excited considering the daily data from the 192-days period.

Fifth pet - Dog5: The dog considered is 2 years and 5 months old, female, weighs 37.3 kg (ideal body condition) and is a German Shepherd Dog. It has chronic skin allergies and has to take a monthly injection, is always indoors, has low usual activity and walks daily.
For this animal there are a total of 86 days of complete data.
The plot below 3.13 shows the total dispersion of 15-second windows for each activity level.

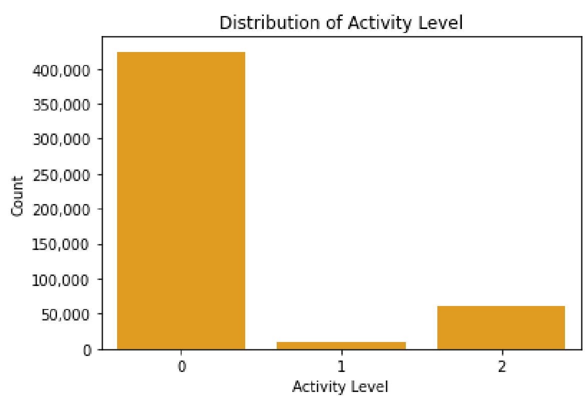

Figure 3.13: Bar plot of activity level for all 15-second windows of the 86 days.

Most time of the animal is spent inactive (95.72% of the period) with an average of 20 hours, 34 minutes and 23 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 02 hours, 57 minutes and 44 seconds per day of activity level equal to 2, corresponding 12.34% of the period. Finally, the pet spends an average of 27 minutes and 52 seconds of the day activities, with only 1.94% of the period considered having an activity level equal to 1.

The data, presented in 3.14, suggests that Mondays, Thursdays, and Saturdays are characterized by a higher amount of inactive time, meaning there is generally less activity during these days.

Specifically, Thursdays and Saturdays tend to have even less active time compared to other weekdays. Moreover, Mondays, Thursdays, and Saturdays are typically associated with lower levels of excitement compared to the rest of the week.

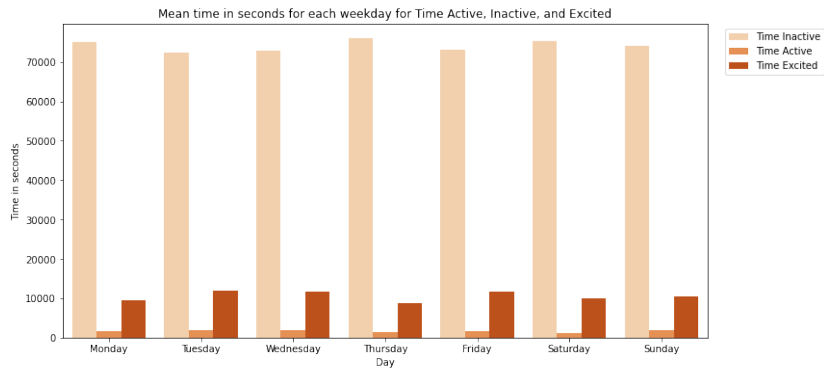

Figure 3.14: Mean time in seconds for each weekday for Time Active, Inactive and Excited.

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered as outliers. Figure 3.15 shows the Box plot of Time Active, Inactive, and Excited considering the daily data from the 86-day period. From the IQR and boxplot visualization it is clear, that for the specific period, no outliers were detected.

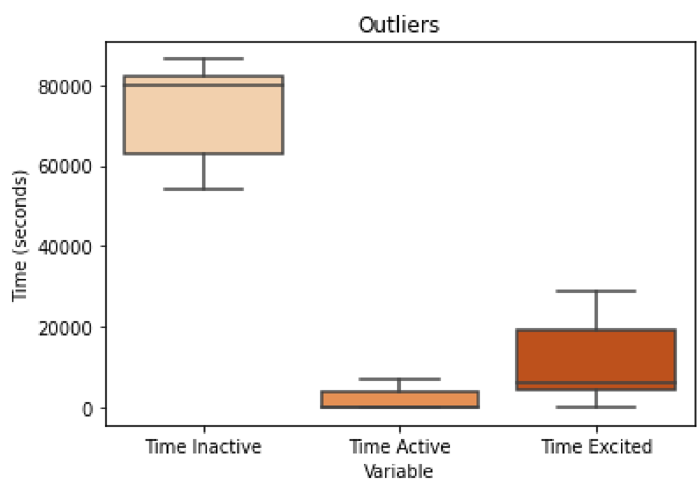

Figure 3.15: Box plot of Time Active, Inactive and Excited considering the daily data from the 86-day period.

Sixth pet - Cat1: The cat considered is 2 years and 8 months old, male, weighs 5.9 kg (ideal body condition) and is an American Shorthair cat. It spends most time indoors, has usually medium activity and has no chronic conditions.
For this animal there are a total of 156 days of complete data.
The plot below 3.16 shows the total dispersion of 15-second windows for each activity level.

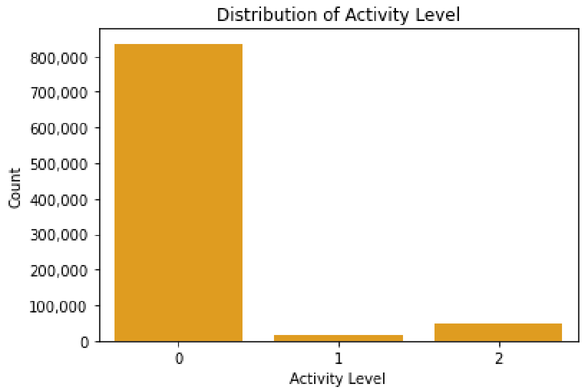

Figure 3.16: Bar plot of activity level for all 15-second windows of the 156 days.

Most time of the animal is spent inactive (93.1% of the period) with an average of 22 hours, 20 minutes and 29 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 01 hours and 59 minutes per day of activity level equal to 2, corresponding 5.23% of the period. Finally, the pet spends an average of 24 minutes and 5 seconds of the day activities, with only 1.67% of the period considered having an activity level equal to 1.

The data, presented in 3.17, suggests that Fridays and Sundays are days of more activity and excitement (and less time inactive).

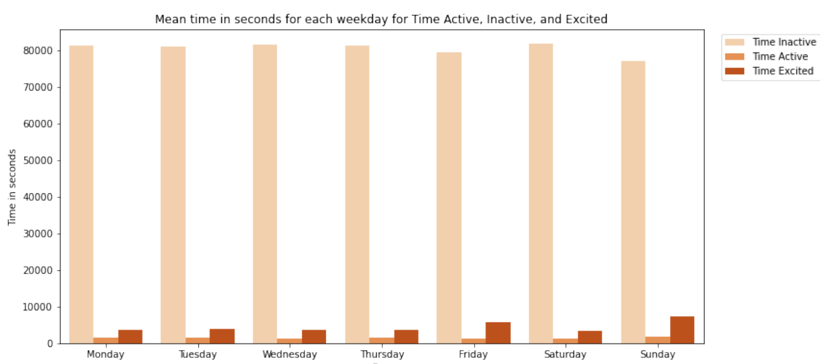

Figure 3.17: Mean time in seconds for each weekday for Time Active, Inactive and Excited.

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered as outliers.

Regarding time inactive and time excited 34 outliers were found and for time inactive 33 outliers were found.

Figure 3.18 shows the box plot of Time Active, Inactive, and Excited considering the daily data from the 156-day period.

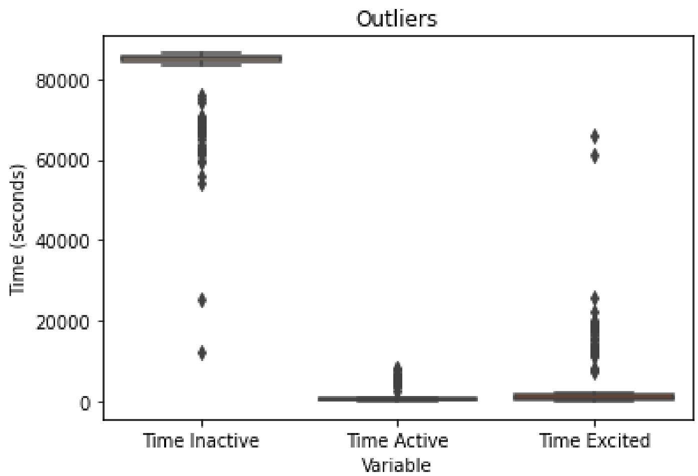

Figure 3.18: Box plot of Time Active, Inactive and Excited considering the daily data from the 156-day period.

Seventh pet - Cat2: The cat considered is 2 years and 9 months old, female, weighs 3.2 kg (ideal body condition) and is a mixed breed cat. It spends all time indoors and has usually low activity. It suffers from cardio-respiratory diseases, skin allergies and generalized anxiety disorder. For this animal, there are a total of 155 days of complete data.
The plot below (3.19) shows the total dispersion of 15-second windows for each activity level.

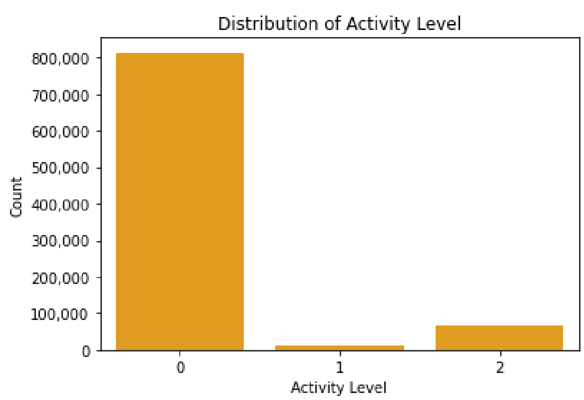

Figure 3.19: Bar plot of activity level for all 15-second windows of the 155 days.

Most time of the animal is spent inactive (91.2% of the period) with an average of 21 hours, 53 minutes and 19 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 01 hour, 48 minutes and 7 seconds per day of activity level equal to 2, corresponding 7.51% of the period. Finally, the pet spends an average of 18 minutes and 32 seconds of the day active, with only 1.29% of the period considered having an activity level equal to 1.

The data, presented in 3.20 , suggests that Mondays, Wednesdays, and Saturdays are marked by a greater amount of inactive time, indicating less overall activity during these days. In particular, Tuesdays and Saturdays tend to have even less active time compared to other days of the week. Additionally, Mondays, Wednesdays, and Saturdays are generally associated with lower levels of excitement.

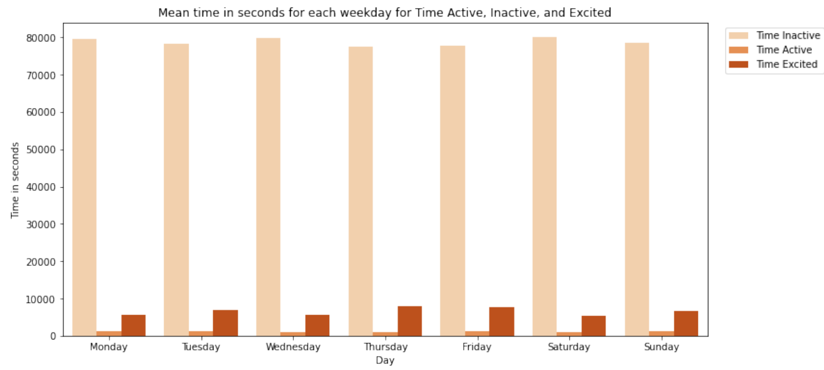

Figure 3.20: Mean time in seconds for each weekday for Time Active, Inactive and Excited.

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered outliers. Regarding time inactive, one outlier was found, which also corresponds to one of the five outliers found for time excited. In what concerns time active, no outlier was found.
Figure 3.21 shows the box plot of Time Active, Inactive, and Excited considering the daily data from the 156-day period.

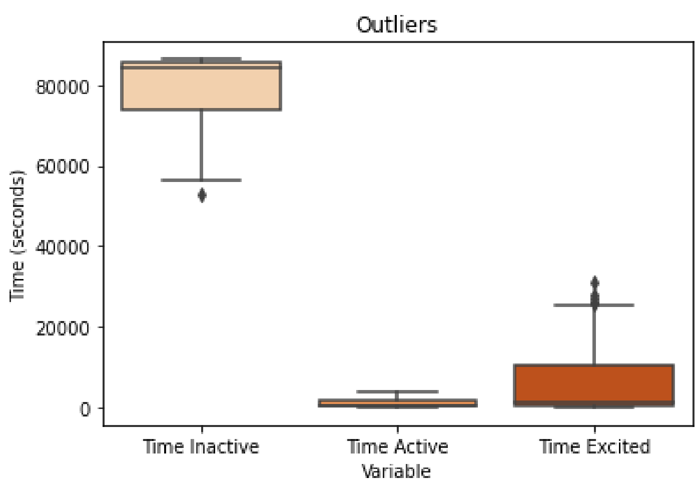

Figure 3.21: Box plot of Time Active, Inactive and Excited considering the daily data from the 155-day period.

Eight pet - Cat3: The cat considered is 10 years and 10 months old, male, weighs 5.5 kg (ideal body condition) and is a Russian Blue Cat. It spends all its time indoors and has usually medium activity. It has chronic skin allergies and anxiety.

For this animal, there are a total of 162 days of complete data.
The plot below 3.22 shows the total dispersion of 15-second windows for each activity level.

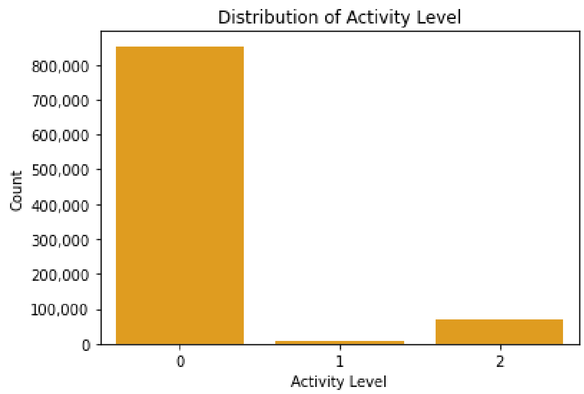

Figure 3.22: Bar plot of activity level for all 15-second windows of the 162 days.

Most time of the animal is spent inactive (91.47% of the period) with an average of 21 hours, 57 minutes and 9 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 01 hour, 48 minutes and 49 seconds per day of activity level equal to 2, corresponding 7.56% of the period. Finally, the pet spends an average of 14 minutes and 1 second of the day activities, with only 0.97% of the period considered having an activity level equal to 1.

The data, presented in 3.23, suggests that from Monday to Thursday, there is a higher amount of inactive time compared to Friday and Saturday. During the weekend, the pet tends to be more active, showing an increase in activity compared to the rest of the week, with Saturday being the day of highest activity. In terms of excitement, there is an overall increase during the weekends, with Saturday being the day that stands out as the day in which the animal is excited for longer.

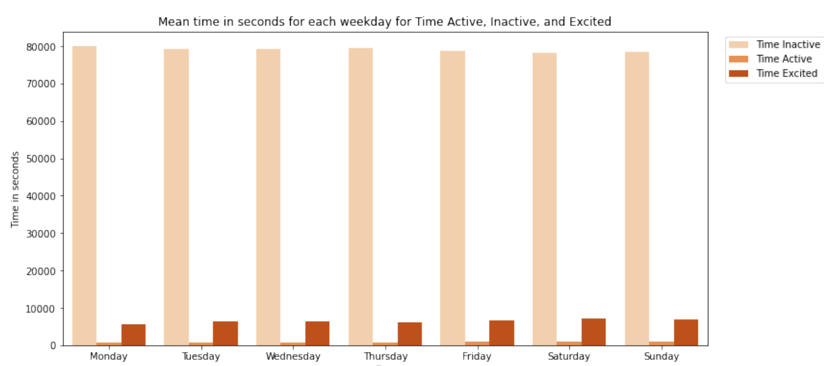

Figure 3.23: Mean time in seconds for each weekday for Time Active, Inactive and Excited.

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered as outliers. Regarding time inactive and time excited 25 days were identified as outliers all in common for both variables. In time active 26 days were identified as outliers, one more than for the other two. Sundays are the days in which more outliers occur (20%), however, for the rest of the days, the abnormalities are equally spread. Note that only 0.08% of outliers occur on Tuesdays. Figure 3.24 shows the Box plot of Time Active, Inactive, and Excited considering the daily data from the 162-day period.

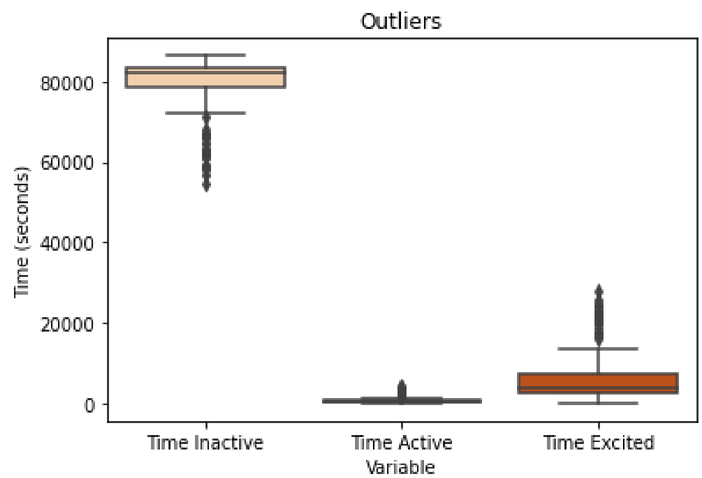

Figure 3.24: Box plot of Time Active, Inactive and Excited considering the daily data from the 162-days period.

Ninth pet - Cat4: The cat considered is 2 years and 10 months old, female, weighs 4.7 kg (ideal body condition), has no chronic conditions and is a Siberian Cat. It spends most time indoors and has usually low activity. For this animal, there are a total of 179 days of complete data.
The plot below 3.25 shows the total dispersion of 15-second windows for each activity level.

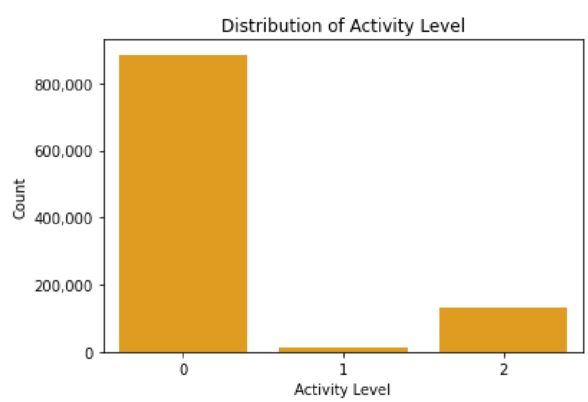

Figure 3.25: Bar plot of activity level for all 15-second windows of the 179 days.

Most time of the animal is spent inactive (86% of the period) with an average of 20 hours, 38 minutes and 13 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 03 hours, 7 minutes and 8 seconds per day of activity level equal to 2, corresponding 13% of the period. Finally, the pet spends an average of 14 minutes and 38 seconds of the day active, with only 1% of the period considered having an activity level equal to 1.

The data, presented in 3.26, suggests that from Friday to Sunday, there is typically a higher amount of inactive time compared to the rest of the week. On Fridays, the pet tends to be more active than during other weekdays, while Wednesdays are generally characterized by low activity. In terms of excitement, there is a noticeable difference from Friday to Sunday compared to the rest of the week. During these days, the pet tends to spend less time in an excited state.

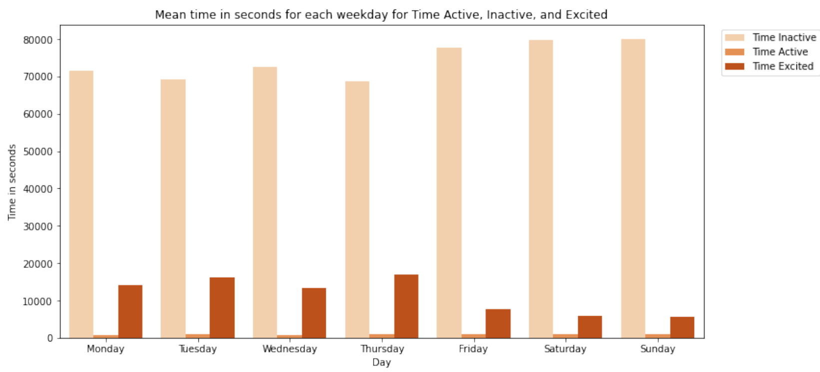

Figure 3.26: Mean time in seconds for each weekday for Time Active, Inactive and Excited.

Figure 3.27 shows the box plot of Time Active, Inactive, and Excited considering the daily data from the 162 days.

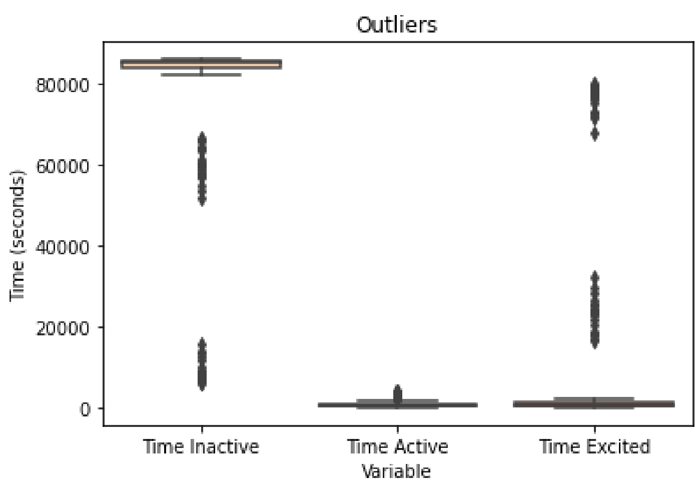

Figure 3.27: Box plot of Time Active, Inactive and Excited considering the daily data from the 179-day period.

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered as outliers. For both time excited and inactive the same 40 days were identified as outliers. Regarding time active only 23 days were identified as outliers, all being outliers also in what concerns the other two variables. It is possible to note that most of these outliers are extreme.

Tenth pet - Cat5: The cat considered is 2 years and 8 months old, female, weighs 3.6 kg (ideal body condition), has no chronic conditions and is a Siamese Cat. It spends most time indoors and has usually high activity. For this animal, there are a total of 36 days of complete data.
The plot below 3.28 shows the total dispersion of 15-second windows for each activity level.

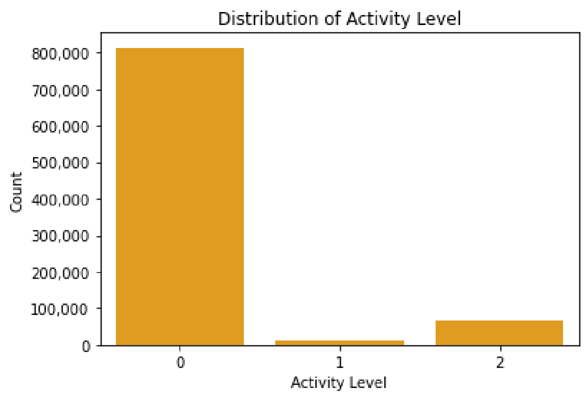

Figure 3.28: Bar plot of activity level for all 15-second windows of the 36 days.

Most time of the animal is spent inactive (91.20% of the period) with an average of 22 hours, 55 minutes and 56 seconds a day with an activity level equal to 0. After time inactive, time excited follows with an average of 03 hours, 7 minutes and 8 seconds per day of activity level equal to 2, corresponding 7.51% of the period. Finally, the pet spends an average of 6 minutes and 59 seconds of the day activities, with only 1.29% of the period considered having an activity level equal to 1.

The data, presented in 3.29 suggests that from Saturday to Monday, there is generally a higher amount of inactive time compared to the rest of the week. However, Tuesdays and Wednesdays stand out as days with less inactive time. Similarly, from Saturday to Monday, there is typically less active time than the rest of the week. Wednesdays and Fridays, on the other hand, are the more active days. In terms of excitement, from Saturday to Monday, there is generally less time spent in an excited state than the rest of the week. However, Tuesdays and Wednesdays stand out as days with more excitement.

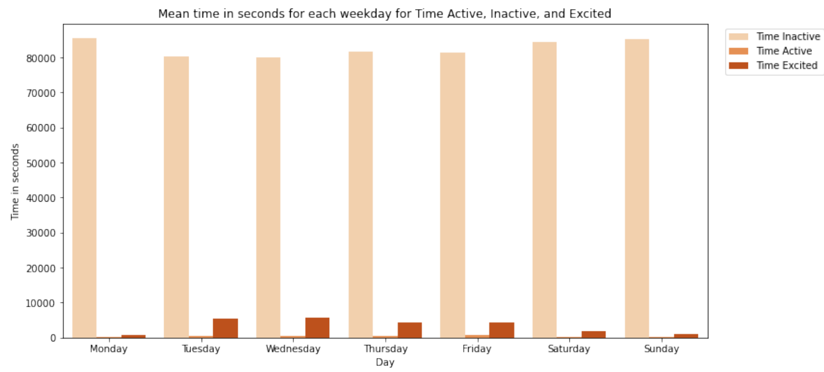

Figure 3.29: Mean time in seconds for each weekday for Time Active, Inactive and Excited.

Using interquartile range and box plot visualization, it is possible to analyze which days can be considered outliers. For all variables, the same four days were considered outliers.

Figure 3.30 shows the Box plot of Time Active, Inactive, and Excited considering the daily data from the 162-day period.

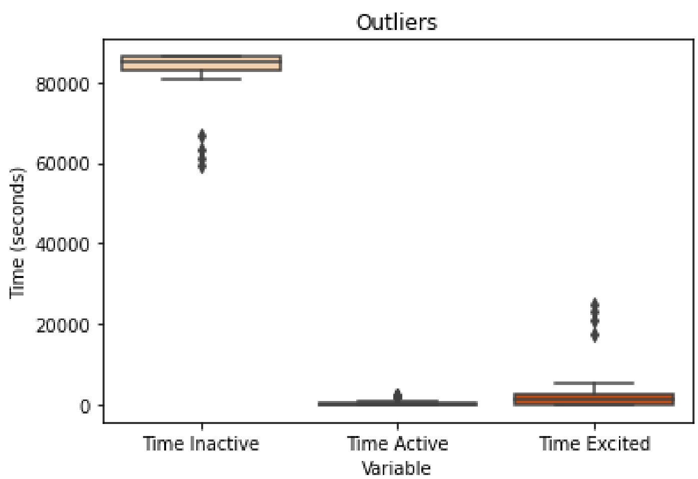

Figure 3.30: Box plot of Time Active, Inactive and Excited considering the daily data from the 36-day period.

3.3 Exploratory Data Analysis - Main conclusions

Based on the analysis of the data for the 10 animals, several key insights can be drawn.

Firstly, it was observed that the animals, on average, spend a significant amount of time inactive, ranging from 83.99% to 95.72% of the observed period. This finding suggests that the threshold used in the processing algorithm used by the company may not be ideal for classifying the inactivity, activity and excitement of the animals. This conclusion answers the first research question.

Additionally, the pets generally exhibit lower excitement levels than their inactive time. The average time spent in an excited state ranges from 1.14% to 8.39% of the observed period. This indicates that the animals experience relatively short bursts of high activity or excitement, interspersed with more extended periods of inactivity.

Furthermore, the animals spend a relatively small amount of time being actively engaged, with their active time ranging from 0.99% to 25 minutes and 28 seconds of the observed period. This finding suggests that the animals have limited intervals of moderate activity.

Analyzing the data every week revealed some observable patterns. Certain weekdays tend to have more inactive time compared to others, while some weekdays exhibit higher levels of activity or excitement. However, these patterns vary among the animals, suggesting individual preferences or routines that influence their behaviour throughout the week.

At last, the analysis identified outliers, representing days with extreme values for activity levels. These point outliers were determined using interquartile range and box plots. The occurrence of outliers varied among the animals, with some individuals exhibiting more outliers than others.

Overall, the analysis suggests that the animals in the study tend to have relatively low levels of activity and excitement, spending the majority of their time in an inactive state. By understanding these patterns, valuable insights into the behaviour of the animals can be gained, enabling the identification of any abnormal or unusual activity levels.

3.4 Data Preparation and Tools

The data preparation phase involves collecting, cleaning, and transforming the data to make it suitable for analysis. This section describes the datasets used for the experiments and outlines the steps to prepare them.

3.4.1 Datasets

The data for this study was collected from an accelerometer device attached to the pets. A private algorithm developed by the collaborating company was applied to preprocess the raw data. The algorithm cleaned and transformed the data, resulting in a variable indicating the activity level for each 15-second window block. The activity levels were 0 for inactive, 1 for active, and 2 for excited.

Only days with complete data were considered for subsequent analysis to ensure the reliability and comprehensiveness of the data.

After the preprocessing step, datasets were constructed for the modelling tasks. The datasets were organized based on different time ranges and derived features from the activity levels.

Daily Datasets

The first dataset includes the duration in seconds that the pets spent on each activity level in a day, i.e., for each of the days available for each animal three new variables were created: total time spent inactive, total time spent active and total time spent excited. Six other variables were also created from the preprocessed data, representing the count of activity level changes within consecutive 15-second blocks (from level 0 to 1, from level 0 to 2, from level 1 to 0, from level 1 to 2, from level 2 to 0 and from level 2 to 1). Additionally, two variables related to the day of the week were included. One variable represents the specific day of the week (e.g., Monday, Tuesday), while the second variable classifies each day as a weekday, weekend day, or national holiday (the national United States of America (USA) holidays were the ones considered since the clients of the company are USA-based).

	time_inactive	time_active	time_excited	day_of_the_week	inactive_to_active	inactive_to_excited	active_to_inactive	active_to_excited	excited_to_active	excited_to_inactive	Week_weekend_holiday
05/10/2022	49860	3690	32850	2	97	169	97	74	74	169	0
06/10/2022	38925	3765	43710	3	77	110	75	85	83	112	0
08/10/2022	33345	2865	50190	5	65	98	72	51	58	91	1
09/10/2022	34920	7095	44385	6	138	176	145	173	180	168	1
10/10/2022	37695	5700	43005	0	80	135	87	66	73	129	0

Figure 3.31: Example of Daily Dataset

In summary, the features developed for daily datasets were the following:

- Index: Day
- **Time Inactive:** Total time, in seconds, that the animal spent inactive (activity level equal to 0) for the day.
- **Time Active:** Total time, in seconds, that the animal spent active (activity level equal to 1) for the day.
- **Time Excited:** Total time, in seconds, that the animal spent excited (activity level equal to 2) for the day.
- **Inactive to Active:** Number of times in the day that the activity level changed from Inactive (0) to Active (1).
- **Inactive to Excited:** Number of times in the day that the activity level changed from Inactive (0) to Excited (2).
- **Active to Inactive:** Number of times in the day that the activity level changed from Active (1) to Inactive (0).
- **Active to Excited:** Number of times in the day that the activity level changed from Active (1) to Excited (2).
- **Excited to Active:** Number of times in the day that the activity level changed from Excited (2) to Active (1).
- **Excited to Inactive:** Number of times in the day that the activity level changed from Excited (2) to Inactive (0).

- **Day of the week:** Corresponds to the day of the week - from Monday to Sunday, in which Monday equals to 0 and Sunday equals 6.
- **Week, weekend or holiday:** The following variable is equal to 0 if the day in question is a working day, 1 when it is a weekend day and 2 when the day is a United States National Holiday.

Daily Dataset - Days of the Week

This dataset contains daily information for days from Monday to Friday. It includes the same variables as the daily dataset, excluding the variable representing whether each day is a weekday, weekend day, or holiday.

	time_inactive	time_active	time_excited	day_of_the_week	inactive_to_active	inactive_to_excited	active_to_inactive	active_to_excited	excited_to_active	excited_to_inactive
27/12/2022	61170	1980	23250	1	41	110	49	64	72	102
28/12/2022	59460	2205	24735	2	49	108	54	73	78	102
29/12/2022	63315	2115	20970	3	58	112	63	57	62	108
30/12/2022	66735	2280	17385	4	41	86	59	52	70	68
04/01/2023	86100	120	180	2	8	9	6	2	0	11

Figure 3.32: Example of Daily Dataset - Days of the Week

Daily Dataset - Days of the Weekend

This dataset contains daily information for days from Sunday to Saturday. It includes the same variables as the daily dataset, excluding the variable representing whether each day is a weekday, weekend day, or holiday.

	time_inactive	time_active	time_excited	day_of_the_week	inactive_to_active	inactive_to_excited	active_to_inactive	active_to_excited	excited_to_active	excited_to_inactive
31/12/2022	83310	330	2760	5	8	9	8	11	11	9
08/01/2023	84885	210	1305	6	10	47	9	5	4	48
14/01/2023	86400	0	0	5	0	0	0	0	0	0
15/01/2023	86400	0	0	6	0	0	0	0	0	0
25/03/2023	86400	0	0	5	0	0	0	0	0	0

Figure 3.33: Example of Daily Dataset - Days of the Weekend

Daily Dataset with Time Periods

For this set of datasets, the same variables as in the daily dataset were aggregated based on different time periods within a day.

The time periods considered were: early morning (00:00:00 - 06:00:00), morning (06:00:01 - 09:00:00), early afternoon (09:00:01 - 12:00:00), afternoon (12:00:01 - 16:00:00), early night (16:00:01 - 20:00:00), and night (20:00:01 - 23:59:59).

In summary, the features developed for daily datasets were the following:

- Index: Day per period of time.
- **Time Inactive:** Total time, in seconds, that the animal spent inactive (activity level equal to 0) for the period of time.
- **Time Active:** Total time, in seconds, that the animal spent active (activity level equal to 1) for the period of time.
- **Time Excited:** Total time, in seconds, that the animal spent excited (activity level equal to 2) for the period of time.
- **Inactive to Active:** Number of times in the period of the day that the activity level changed from Inactive (0) to Active (1).
- **Inactive to Excited:** Number of times in the period of the day that the activity level changed from Inactive (0) to Excited (2).
- **Active to Inactive:** Number of times in the period of the day that the activity level changed from Active (1) to Inactive (0).
- **Active to Excited:** Number of times in the period of the day that the activity level changed from Active (1) to Excited (2).
- **Excited to Active:** Number of times in the period of the day that the activity level changed from Excited (2) to Active (1).
- **Excited to Inactive:** Number of times in the period of the day that the activity level changed from Excited (2) to Inactive (0).
- **Day of the week:** Corresponds to the day of the week - from Monday to Sunday, in which Monday equals to 0 and Sunday equals 6.
- **Week, weekend or holiday:** The following variable is equal to 0 if the day in question is a working day, 1 when it is a weekend day and 2 when the day is a United States National Holiday.

	Time_Period	day_of_the_week	inactive_to_active	inactive_to_excited	active_to_inactive	active_to_excited	excited_to_inactive	excited_to_active	time_inactive	time_active	time_excited	Week_weekend_holiday
05/10/2022	night	2	20	20	20	8	21	8	17400	555	3660	0
05/10/2022	early morning	2	7	9	6	1	10	0	4710	105	5985	0
05/10/2022	morning	2	18	13	19	11	11	12	8535	840	1425	0
05/10/2022	early afternoon	2	14	32	13	13	34	12	6990	465	6945	0
05/10/2022	afternoon	2	25	69	28	25	65	28	6840	990	6570	0
05/10/2022	early night	2	13	26	11	16	28	14	5385	735	8265	0

Figure 3.34: Example of Daily Dataset with Time Periods

Daily Dataset - Days of the Week per Time Period

This dataset includes the same variables as the "Daily Dataset - Days of the Week" but for each time period within a day. The time periods considered are the same as mentioned in the previous paragraph.

Daily Dataset - Days of the Weekend per Time Period

Similar to the previous dataset, this dataset includes the same variables as the "Daily Dataset - Days of the Weekend" but for each time period within a day. The time periods considered are the same as mentioned in the previous paragraph.

3.4.2 Data Preparation - Main Conclusions

In this subsection, the data were prepared for analysis by undergoing preprocessing using a private algorithm. The collected data from the accelerometer device was transformed into a suitable format for modeling, categorizing activity levels as inactive, active, or excited.

Datasets were created based on different time ranges and derived features, serving as the foundation for the upcoming modeling phase. These datasets enable the exploration of anomaly detection algorithms and different methods for aggregating time-based anomalies while preserving their relevance.

With the data now prepared and organized, the focus can shift to the modeling phase, where various algorithms will be applied to build models capable of identifying anomalies and evaluating different aggregation approaches.

The data preparation phase ensures the data is in the right format for accurate analysis, paving the way for valuable insights and informed decision-making in the subsequent stages of the study.

3.4.3 Tools

In the development of this dissertation, Python was the software used. For the data analysis tasks the packages used were "pandas", "matplotlib", "time", "numpy" and "seaborn". Regarding the dataset creation and modelling, the following packages were used: "pandas", "matplotlib", "seaborn" and " pyod" - specifically the "KNN", "LOF", "IForest", "standardizer" and "models combination (average)". In what concerns the application of Shapley Values for model explainability the package used was "shap".

Chapter 4

Methodology

In this chapter, the modelling techniques used for anomaly detection are described, specifically Isolation Forest (IF), K-nearest neighbours (KNN), and Local Outlier Factor (LOF).

The parameter selection and the aggregation approach applied to these algorithms are also discussed. Furthermore, the ways of defining a threshold for differentiating anomalies from normal instances are mentioned.

Additionally, the experimental setup using the data, as mentioned in section 3.4, and the methodology described in this section are explained.

Finally, the evaluation metrics used to assess the performance of the proposed approach are presented.

4.1 Model Selection

The K-Nearest neighbour (K-NN), Local Outlier Factor (LOF), and Isolation Forest (IF) algorithms have been selected approaches for the anomaly detection task based on their proven effectiveness and suitability for different types of anomalies, mentioned in 2.

K-NN is a widely used method for global anomaly detection and has consistently outperformed clustering techniques in outlier detection tasks.

Isolation Forest is effective in detecting global anomalies that significantly deviate from the overall data trends. It does not require labelled or abnormal data, making it suitable for the specific application.

LOF is designed for detecting local anomalies and anomalies influenced by proximity or interdependencies. It has demonstrated superior performance in domains like credit card fraud and medical fraud detection, outperforming other available techniques.

By choosing K-NN, LOF, and IF, the anomaly detection task can effectively address both global and local anomalies. K-NN and IF are suitable for capturing global anomalies, while LOF excels in identifying local anomalies and anomalies influenced by proximity. These algorithms have been validated in studies and proven effective across various domains, making them reliable choices for the anomaly detection task.

4.1.1 Isolation Forest for Anomaly Detection

Isolation Forest is a density-based tree-ensemble outlier detection method that differs from most methods since it explicitly finds anomalous data points. Other methods usually start by recognizing the normal data points and then classifying anomalies as data points that do not fit into the profile of normal data.

It does not use distance measures, allowing the algorithm to be faster and perform well for large and high-dimensional datasets (Dataman (2023); Iivari (2022); Zhong et al. (2019b)). It is also an unsupervised technique in the sense that it does not need labels.

This algorithm detects anomalies by creating an ensemble of decision trees. Then the data is recursively randomly partitioned/sub-sampled in a tree structure based on randomly selected features until all data at the node has the same values, the node has only one sample, or the tree reaches the restricted height. Given a sample of data $X = x1, ..., xn$ of m instances from a d-variate distribution, to build an isolation tree (iTree), we recursively divide X by randomly selecting an attribute q and a split value p, until either: the tree reaches a height limit, $|X| = 1$ or all data in X have the same values.

An iTree is a binary tree in which each node has precisely zero or two daughters. Assuming that all instances are distinct when an iTree is fully grown, each instance is isolated to an external node, in which case the number of external nodes is n and the number of internal nodes is n -1; the total number of nodes of an iTree is $2n$- 1; and thus the memory requirement is bounded and only grows linearly with n.

The concept of sub-sampling serves as the foundation for the model's mechanism. Isolation Forest is able to reduce the variance of the anomaly score estimates by creating a less correlated forest of trees by randomly selecting a subset of the data. The utilization of sub-testing empowers the algorithm to take advantage of sub-examining to a degree that isn't practical in existing techniques, creating an algorithm which has a linear time complexity with a low constant and a low memory requirement (Liu, Ting, and Zhou (2008)).

The algorithm for constructing an iTree is as follows:

1. If the tree depth limit is achieved or there is only one data point left, return a leaf node containing the data point.

2. Randomly select a feature and a split value between the maximum and minimum values of the selected feature.

3. Partition the data into two subsets based on the selected feature and split value.

4. Recursively apply steps 1-3 to each subset until all data points are isolated in their own leaf node.

The algorithm for constructing an iForest is as follows:

1. For each sub-sampling size, randomly select a subset of the data.

2. Construct an iTree using the selected subset of data.

3. Repeat steps 1-2 for the desired number of trees.

4. Calculate the anomaly score for each data point based on the number of splits required to isolate it in the forest.

To sum up, the algorithm chooses a feature at random and then chooses a split value at random from the range of the feature. Using this split value as a basis, the data is then divided into two partitions, and the procedure is continued recursively until every data point is isolated in its partition. The anomaly score of a data point is calculated using the number of splits necessary to isolate it (Zhong et al. (2019b)).

The following properties of anomalous samples, also known as outliers, are then utilized by the Isolation Forest Algorithm to define anomalies:

- Fewness: There will be a small number of anomalous samples in any dataset because they are rare.

- Different: Anomaly samples differ greatly from normal samples in terms of their values and attributes.

Samples deeper in the tree are less likely to be anomalous since more cuts are required to isolate them. Anomalies will be the observations with short average path lengths on the trees,i.e., shorter branches for samples suggest anomalies since the tree found it simpler to distinguish them from other data. The average path length measures the degree of susceptibility to isolation.

The anomaly score of a data point is calculated using the number of splits necessary to isolate it (Zhong et al. (2019b)), as can be seen with the following figure that represents in a simple way how the model works.

The bright orange dot corresponds to an anomaly, the light orange dot is a point that depending on the threshold used could be classified as an anomaly. The yellow point corresponds to a normal instance.

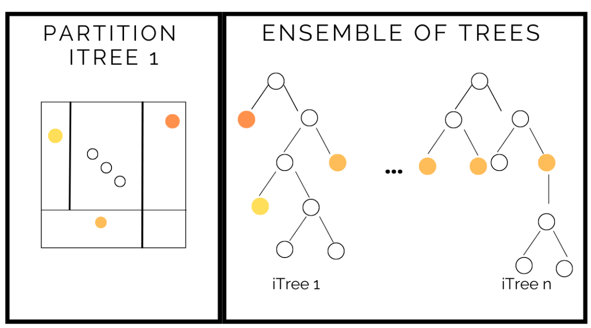

Figure 4.1: Isolation Forest Functioning.

After the Isolation Forest is created (a collection of Isolation Trees) the algorithm uses the following equation to compute the anomaly score given a data point x and a sample size of m (Mavuduru (2022); Mougan (2020)):

$$s(x, m) = 2^{-\frac{E(h(x))}{c(m)}} \tag{4.1}$$

In the equation 4.1:

- *$s(x,m)$:* represents the anomaly score of instance x in a given tree with m instances.
- *$h(x)$:* represents the path length of x in an isolation tree (the number of edges of x transverses on an iTree from the root node until the transversal is terminated at an external node).
- *$E(h(x))$:* represents the expected values of the path length across all the Isolation Trees, i.e., the average of *h(x)* from a collection of isolation trees.
- *$c(m)$:* represents the average values of *h(x)* given a sample size of m and is used for normalizing *h(x)*. It is defined using the following equation.

$$c(m) = 2H(n-1) - \frac{2(n-1)}{n} \tag{4.2}$$

  Where $H(i)$ is the harmonic number and it can be estimated by *ln(i)+ 0.5772156649* (Euler's constant).

Once the anomaly score $s(x,m)$ is computed for a given point, we can detect anomalies using the following criteria:

- If $s(x,m)$ is close to 1 then x is very likely to be an anomaly.
- If $s(x,m)$ is less than 0.5 then x is likely a normal point.
- If $s(x,m)$ is close to 0.5 for all of the points in the dataset then the data likely does not contain any anomalies.

It is important to consider that the task of anomaly detection is to provide a ranking that reflects the degree of anomaly. In that sense, the usual analysis of $s(x,m)$ may not be ideal for all circumstances. One way to detect anomalies is to sort data points according to their anomaly scores and anomalies will be points that are ranked at the top of the list (Liu et al. (2008)).

When applying this algorithm the following parameters, more relevant to the performance of the algorithm, should be defined:

- **Maximum number of samples**: This refers to the number of samples randomly selected as candidates for splitting at each node.
- **Number of estimators**: Which refers to the number of trees in the ensemble.
- **Contamination Rate**: This refers to the percentage of anomalies present in the data. For most cases in real-world applications, this parameter is very hard to find and so usually a value equal to five or ten per cent is defined and then specific threshold techniques are used to identify anomalies.

Isolation Forest Modelling

An aggregation by an average of multiple models with different parameters was performed for the Isolation Forest algorithm, as according to Dataman (2023), in the case of unsupervised learning in which hyper-parameter tuning is not possible, this is a feasible approach for a more stable model.

Specifically, the values of 20, 40, 60, 80, and 100 were combined for the number of estimators and the maximum number of samples (the most relevant parameters), enabling an evaluation of the algorithm's performance across various parameter combinations.

The contamination rate was set to 10 per cent, a widely used value that then loses its importance when defining an adequate threshold.

To differentiate anomalies from normal instances, two techniques were employed: boxplot and interquartile range visualization, as well as the 95th percentile. The anomaly scores were plotted in a boxplot, and instances lying outside the whiskers were identified as potential outliers. Additionally, a threshold was determined using the 95th percentile of the anomaly scores, classifying instances above this threshold as anomalies.

In Dataman (2023), the author suggests the visualization of the data to define a reasonable threshold and in that sense, the boxplot was used, however, as mentioned during this section, anomalies should be the instances with the highest ranking in anomaly score thus the use of the 95th percentile.

4.1.2 K-NN for Anomaly Detection

The k-nearest neighbour is a technique that is mostly used in supervised learning. However, K-NN can be used as an unsupervised algorithm in detecting global anomalies by computing the k-nearest neighbours. The model computes the distance to other data points for each data point, then sorts the data points from smallest to largest by distance and selects the first k entries. The anomaly score can be computed in two different ways: the distance to the nearest neighbour or the average distance to its k-nearest-neighbours. The distance can be computed through different measurements, the most popular being Euclidean distance (Dataman (2023); Zhong et al. (2019b)).

The unsupervised version of the K-Nearest neighbour can be described by the following steps:

1. For each instance, the distance to other data points is computed.
2. Then the data points are sorted from smallest to largest by the distance.
3. Lastly, the first K entries are picked.

The distance can be measured using different measurements, however, in this case, the Euclidean distance, given by the following formula was applied.

$$EuclideanDistance(x, y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2} \tag{4.3}$$

Where:

- x, y: represent the two data points.
- x_i, y_i: are the respective coordinates of the two points in each dimension.
- d: represents the number of dimensions or features in the dataset.

After the application of the K-NN model, the anomaly score is defined as the largest of distance to k neighbours. Besides the largest distance, some also commonly used measures to define the anomaly score can be the median or mean distance to the k neighbours.

When applying this algorithm the following parameters, more relevant to the performance of the algorithm, should be defined:

- **Number of neighbours**: This refers to the number of neighbours to consider when detecting the anomalies.
- **Contamination Rate**: This refers to the percentage of anomalies present in the data. For most cases in real-world applications, this parameter is very hard to find and so usually a value equal to five or ten per cent is defined and then specific threshold techniques are used to identify anomalies.

K-NN Modelling

Two approaches were considered to determine the anomaly scores: the distance to the nearest neighbour and the average distance to the k-nearest neighbours. The Euclidean distance, a commonly adopted metric for KNN-based anomaly detection, was used to measure the distance.

Similar to the Isolation Forest, an aggregation by an average of multiple parameters was conducted for the K-NN algorithm, as according to Dataman (2023) in the case of unsupervised learning in which hyper-parameter tuning is not possible, this is a feasible approach for a more stable model.

For the datasets that included a period of time, the number of neighbours started with 1, 2, 3,4,5 and 10 and was incremented by 10 until reaching the total number of rows. For the datasets with daily combined data, the number of neighbours started with 1, 2, 3, 4, and 5, and then increased by five until reaching the total number of rows. The contamination rate for the reasons mentioned in the previous model was set to ten.

To define a threshold for distinguishing anomalies from normal instances, the distribution of anomaly scores was visualized using a boxplot and interquartile range. Additionally, the 95th percentile of the anomaly scores was used as a threshold, classifying instances above this threshold as anomalies.

4.1.3 Local Outlier Factor for Anomaly Detection

Local outlier factor (LOF) is a density and nearest-neighbour- based outlier detection algorithm that aims to find local anomalies through the comparison of the density of each data point.

The method determines a data point's local density in relation to its neighbours and classifies anomalous data points as those that have a significantly lower density than their neighbours. The use of distance ratios in LOF ensures that varying local densities can be accounted for (Dataman (2023)). Each data point receives a score from the LOF algorithm based on the density of its neighbours. The score for a data point is calculated as the ratio of its own local density to the average local density of its k-nearest neighbours. The inverse of a point's average distance to its k-nearest neighbours is known as its local density. Usually, values with LOF scores above one are considered outliers, however, visualising the scores is essential to define a reasonable threshold to separate abnormal points from normal ones (Budiarto, Permanasari, and Fauziati (2019); Dataman (2023)).

To achieve the Local Outlier Factor scores the following steps are produced by the algorithm (Ma, Ngan, and Liu (2016)) :

1. For each (using the Euclidean measurement, as defined in equation 4.3)

2. Definition of all K-Nearest neighbours for each data point.

3. Computation of the K-distance, i.e., for each data point *p*, calculate the distance to its *k-th* nearest neighbour - the k-distance(*p*).

4. Computing the reachability density For each pair of points *p* and *q*: corresponds to the maximum of either the K-distance(*q*) or the distance between *q* and *p*. The longer the reachability distance, the more likely point *p* is an outlier. The reachability density is given by:
$$RD_k(p, m) = \max(\text{k-distance(q)}, dist(q, p)) \tag{4.4}$$
   Where:

   - q is a target point and p is the current data point
   - *k-distance(q)* corresponds to the *kth* smallest distance to a data point m.
   - *dist(q,p)* corresponds to the Euclidean distance between the target point *m* and the current point *p*

5. Computing the Local Reachibility distance: The reciprocal of the mean of the reachable distance of the data point *p* and its nearest neighbour *m*. Low values imply the closest data body is far from point *p*. The LRD of a point is used to compare with the average LRD of its K neighbours and is given by the following formula:
$$\text{LRD}(p) = \left(\frac{1}{\sum_{q \in N(p)} \frac{\text{RD}(p,q)}{\text{k-distance}(p)}} \right) \tag{4.5}$$
   Where:

   - N*(p)* represents the set of data points in the neighbourhood of *p*.

6. Compute the local outlier factor (LOF) for each data point: the ratio of the average LRD of the K -neighbours of point p to its LRD, as shown in the following formula:

$$\mathrm{LOF}(p) = \frac{\sum_{q \in N(p)} \frac{\mathrm{LRD}(q)}{\mathrm{LRD}(p)}}{|N(p)|} \tag{4.6}$$

When applying this algorithm the following parameters, more relevant to the performance of the algorithm, should be defined:

- **Number of neighbours**: This refers to the number of neighbours to consider when detecting the anomalies.

- **Contamination Rate**: This refers to the percentage of anomalies present in the data. For most cases in real-world applications, this parameter is very hard to find and so usually a value equal to five or ten per cent is defined and then specific threshold techniques are used to identify anomalies.

Local Outlier Factor Modelling

An aggregation by an average of multiple models with different parameters was conducted for the LOF algorithm, for the reasons mentioned in the previous algorithms . The contamination rate, as in the other algorithms, was set to 10%. Similar to the KNN algorithm, the number of neighbours varied depending on the dataset's characteristics. For datasets with a time period, the number of neighbours started with 1,2,3,4,5 and 10 and then was incremented by 10 until reaching the total number of rows. For datasets with daily combined data, the number of neighbours started with 1, 2, 3, 4, and 5, and then increased by five until reaching the total number of rows.

To define a threshold for distinguishing anomalies from normal instances, as in the previous models, two techniques were employed: boxplot and interquartile range visualization, as well as the use of the 95th percentile. The anomaly scores were visualized using a boxplot, enabling the identification of potential outliers lying outside the whiskers. Additionally, a threshold was established using the 95th percentile of the anomaly scores, classifying instances above this threshold as anomalies.

4.2 Model Explainability

One important factor in detecting anomalies in a living being's behaviour is understanding the anomaly's cause.

To overcome this limitation and be able to provide a better explanation of why an instance is considered an anomaly by the proposed technique, one possibility is to resort to Shapley values Molnar (2023).

Shapley values were chosen to explain anomalies detected in pet data due to several reasons. They provide insights into feature contributions and their importance in anomaly detection after the model has made its predictions. This method offers a rigorous and interpretable approach with certain advantages compared to other techniques. It provides local and global explanations, enhancing the interpretability of the model's outputs.

This method enables us to comprehend the relative significance of each feature in influencing the model's predictions (Gopinath (2021)). The Shapley values idea, which comes from cooperative game theory, considers all potential feature combinations and their contributions to the projected outcome. Shapley values assign a distinct value to each feature in a model.

Calculating Shapley values involves evaluating the marginal contribution of each feature across all possible feature subsets. A fair distribution of contributions is obtained by considering all possible orders in which features are added to the subsets. This ensures that no feature is either overestimated or underestimated in its importance.

The following steps are performed to compute Shapley Values (Sundararajan and Najmi (2020)):

1. The game is defined: The prediction obtained by the model is treated as a cooperative game. The players in this game are the features the model uses, and the prediction is the outcome.

2. The coalition of the game is defined: The coalition corresponds to the set of all features used by the models. It includes all possible subsets of features.

3. Generate the permutations: For each subset of features in the coalition, all possible permutations of features in that subset are created. Each permutation is a specific ordering of the features.

4. Computing Marginal Contributions: For each feature in a certain permutation, calculate its marginal contribution, which is obtained by subtracting the prediction made by the model without the feature from the prediction made with the feature. This measures the individual impact of the feature. The marginal contributions are computed as follows:

$$\text{Marginal Contribution of Feature i} = \text{Prediction with } i - \text{Prediction without } i \tag{4.7}$$

5. Computation of Shapley Values: The average marginal contribution in all permutations for each feature is computed. That average regards the Shapley value of the feature, i.e., the average contribution of the feature in all permutations. The Shapley Values are computed as follows:

$$\text{SV of Feature } i = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} \binom{|S|}{n-1}^{-1} \left[\sum_{\sigma \in \Sigma(S \cup \{i\})} \left(V(\sigma(S \cup \{i\})) - V(\sigma(S)) \right) \right] \tag{4.8}$$

Where:

- N: represents the set of all features.
- $\Sigma(S)$: corresponds to the set of all permutations of features in the subset S.
- V: represents the prediction of the model.

4.2.1 Shapley Values - Study Case

In this study-specific case, the Shapley values for the best-performing solution were computed by calculating an average of the Shapley values for each feature in each specific model of the final aggregation. This global approach was performed to understand the model behaviour of each pet.

In this way, and since the best solution is an aggregation of different models of the same algorithm with different parameter values, all those models will also be considered when analyzing the feature contribution.

An average of the feature importance for the data of each animal was performed to have an overview of what explains the anomalies in general for that animal. However, in a business setting, ideally, when the model detects an anomaly, the Shapley value of that instance should be studied alone to understand what triggered the anomaly detector.

4.3 Experimental Setup

In this subsection, a more in-depth description of the process used for anomaly detection based on the data and methodology explained before will be given.

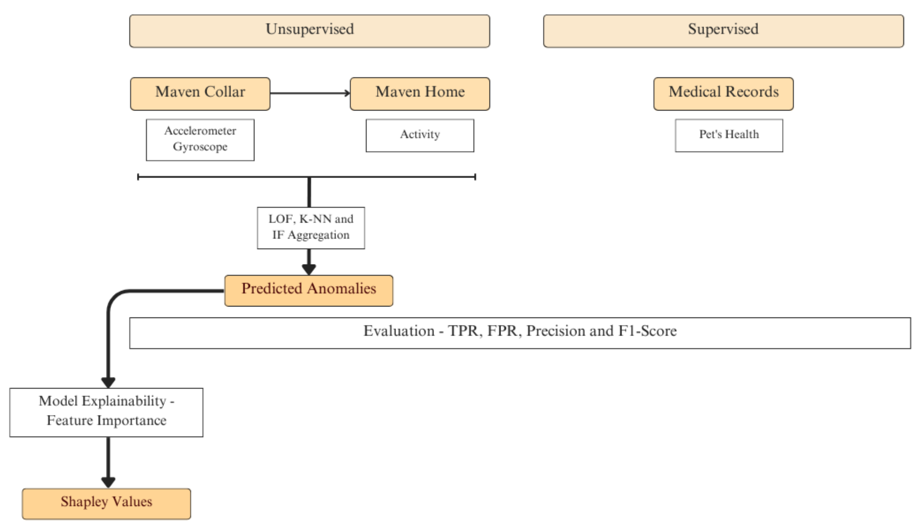

Figure 4.2: Approach Overview

The development of six new datasets, as mentioned in section 3.4, allows for the application of the algorithms.

As the techniques used are unsupervised and no labels are available regarding anomalies, the data used was not split into train and test. The available data was used as if there was only a training phase.

The data was first standardized to apply the three models' aggregations since not all features were based on the exact measurement.

Then the Isolation Forest, Local Outlier Factor and K-Nearest neighbour aggregations were applied, as described, for all six datasets. This allows the study of the best way to represent the data when detecting anomalies. Moreover, it also can be the path to distinguish which models perform best.

Besides the regular application of anomaly detection techniques, an ensemble of all models for each dataset was performed to test if joining different experts (the algorithms) could be a better option for the main task of this dissertation.

Furthermore, an aggregation of the results for each model for all datasets, i.e., aggregating by hard vote the results, for example, of Local Outlier Factor, considering all datasets, was performed to understand each model's performance independently of how the data is presented.

Lastly, Shapley Values were applied to the results obtained with the best data representation and the aggregation of algorithms that were most successful in detecting anomalies.

4.4 Evaluation Metrics

To evaluate the results obtained when using the data and methodologies, described in the previous sections, medical records for the ten animals studied were provided by the company.

For each animal, various periods/occurrences of health issues were obtained through the medical records.

Table 4.1: Health issues per pet

	Cats				
	Cat 1	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Number of health issues	7	5	4	5	2
Total available days in the medical records	156	155	162	179	36
	Dogs				
	Dog 1	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Number of health issues	4	7	3	13	5
Total available days in the medical records	156	151	108	86	192

The evaluation techniques, used to evaluate the results obtained when applying the algorithms were the True Positive Ratio (TPR)/Recall, False Positive Ratio (FPR), Precision and F1-score.

True Positive Ratio/Sensitivity/Recall: Corresponds to the number of occurrences correctly detected of the total health issues detected with the medical records. This evaluation metric is also known as sensitivity and is computed as follows.

$$\text{True Positive Ratio/Sensitivity/Recall}_i = \frac{\text{Anomalies detected correctly}_i}{\text{Total amount of health issues }_i} \tag{4.9}$$

Where i represents a specific animal. *Anomalies detected correctly* correspond to the true positives and the *Total amount of health issues* is the sum of true positives and false negatives.

As the aim of the study is to prevent anomalies, it is important to not only consider as a true positive the day when the health issue happened. In addition to that, it might also be important to detect an anomaly even after the exact moment that it happened, mostly because if this anomaly is not noticed by a pet owner or a veterinary, it can be a way of acting to prevent possible complications originating from that anomaly or even for providing the right care right after the anomaly occurs.

For the reasons already mentioned, a true positive was considered when the algorithms emitted an alert of an anomaly two days before the anomaly, the actual period, and two days after.

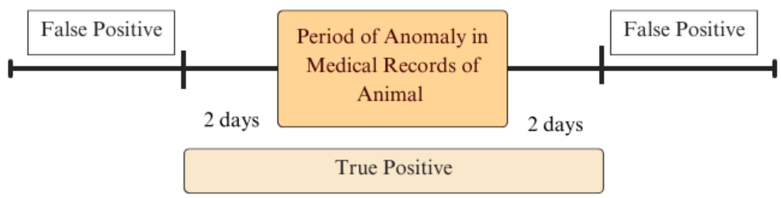

Figure 4.3: Evaluation Representation

False Positive Ratio: Corresponds to the proportion of the total number of false positives (wrongly detected anomalies) by the sum of false positives and true negatives (all normal instances). It's the probability that a false alarm is, a positive result will be given when the true value is negative. It is computed as follows:

$$\text{False Positive Ratio}_i = \frac{\text{Wrongly detected Anomalies }_i}{\text{Total normal instances}_i} \tag{4.10}$$

Where i represents a specific animal. *Wrongly detected anomalies* correspond to false positives and the *Total normal of health issues* is the sum of true negatives and false positives.

Precision: Corresponds to the ratio between the true positives and all the positive values (anomalies) detected by the proposed approaches. It's the percentage of time that the approach rightly predicts anomalies. It is computed as follows:

$$\text{Precision}_i = \frac{\text{Anomalies detected correctly }_i}{\text{Total anomalies detected}_i} \tag{4.11}$$

Where i represents a specific animal. *Anomalies detected correctly* correspond to true positives and the *Total anomalies detected* is the sum of true positives and false positives.

F1-score: It is an evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model providing a harmonic mean of both metrics. It is computed as follows:

$$\text{F1-score}_i = 2 \cdot \frac{(\text{Precision}_i \cdot \text{Recall}_i)}{(\text{Precision}_i + \text{Recall}_i)} \tag{4.12}$$

Where i represents a specific animal.

The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

To evaluate the anomalies obtained by using the methodology described and the data (as mentioned in 3.4), the True Positive Ratio/Sensitivity, False Positive Ratio, Precision and F1-score were used. The tables for each animal containing the metrics, on which the analysis was based, for each dataset when applying each aggregation of each technique can be found on .1 and .2.

Chapter 5

Obtained Results

In this section, the results obtained from the different techniques used are discussed, and the answer to the research questions are also provided.

Moreover, the results, that allow answering the research questions, are provided focusing on the best data representation for the specific problem and the best-performing method from the proposed solutions.

Lastly, attention is turned to the explanation of the greatest-performing model for each of the ten animals.

5.1 Data Representation

One of the main research questions of this study is to understand which type of representation of the data is best for detecting anomalies, given the format of the data available.

To answer this question, an aggregation of all the of algorithms for each dataset, using hard vote, was performed, i.e., for each of the six datasets, the anomalies detected with the aggregation by average of LOF, IF and K-NN using the two types of threshold were aggregated by majority voting.

The first observation possible to identify is that when the dataset used considers all days (week and weekend days), the results, as expected are better, however, some exceptions were noted (such as Dog 1).

Secondly, it is possible to conclude that including periods of time and not all day data allows for better results in terms of detecting true anomalies, for most cases.

Overall, from Table 5.1, it is evident that considering daily information divided into periods is the most suitable option for anomaly detection, as it presented the best results of recall for 80% of the animals.

For the remaining 20%, the results of daily data with periods of time are equal and not better than for other data representations with the highest true positive ratio.

If each aggregation of model is considered, and not an ensemble of all proposed techniques the same conclusion can be reached.

Table 5.1: Sensitivity/Recall of the Ensemble of Algorithms per Dataset

	Cats				
Datasets	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Daily	42.86%	20.00%	50.00%	80%	0%
Daily - Time Period	71.43%	20.00%	50.00%	100%	50.00%
Daily (weekdays)	42.86%	20.00%	50.00%	60.00%	0%
Daily (weekdays) - Time Period	71.43%	0%	50.00%	80.00%	0%
Daily (weekend days)	14.29%	0%	0%	20.00%	0%
Daily (weekend days) - Time Period	42.86%	20.00%	25.00%	60.00%	0%
	Dogs				
Datasets	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Daily	25.00%	42.86%	0%	23.08%	40.00%
Daily - Time Period	50.00%	71.43%	33.33%	46.15%	60.00%
Daily (weekdays)	50.00%	14.29%	0%	15.38%	40.00%
Daily (weekdays) - Time Period	50.00%	57.14%	0%	46.15%	60.00%
Daily (weekend days)	0%	28.57%	0%	23.08%	20.00%
Daily (weekend days) - Time Period	0%	57.14%	0%	30.78%	40.00%

Table 5.2: False Positive Ratios of the Ensemble of Algorithms per Dataset

	Cats				
Datasets	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Daily	0.93%	0%	4.84%	1.02%	6.43%
Daily - Time Period	8.33%	8.33%	16.94%	15.31%	21.43%
Daily (weekdays)	0.93%	0%	2.42%	1.02%	5.00%
Daily (weekdays) - Time Period	7.41%	4.17%	10.48%	13.27%	15.71%
Daily (weekend days)	0.93%	0%	2.42%	0%	2.14%
Daily (weekend days) - Time Period	0.93%	8.33%	5.65%	1.02%	7.14%
	Dogs				
Datasets	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Daily	4.35%	5.38%	6.25%	0%	3.57%
Daily - Time Period	18.26%	21.51%	17.50%	3.53%	17.86%
Daily (weekdays)	2.61%	3.23%	5.00%	0%	2.14%
Daily (weekdays) - Time Period	13.04%	19.35%	15.00%	3.49%	12.86%
Daily (weekend days)	0.87%	1.08%	2.50%	0%	2.14%
Daily (weekend days) - Time Period	6.09%	6.45%	5.00%	1.16%	6.43%

From table 5.2, which provides information regarding the false positive ratio, it is important to note that daily information divided into periods allowed for better results in what concerns detecting anomalies, however, it also leads to the greatest number of instances being wrongly classified as an anomaly.

In the context of this dissertation, as the aim is to be able to detect true anomalies in pet behaviour, higher true positive ratios are preferred, even if a greater number of occurrences are wrongly detected as an anomaly.

5.2 Algorithms Performance

The best data representation was used to evaluate the different algorithms' results. All models with different thresholds were performed for each animal, considering the daily dataset with periods of time. With this information, we can answer the third research question for this study.

Table 5.3: Sensitivity per Algorithm for Daily data with time period

	Cats				
Algorithms	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Isolation Forest (95% threshold)	71.43%	40.00%	75.00%	100%	50.00%
Isolation Forest (IQR threshold)	71.43%	20.00%	50.00%	100%	50.00%
K- Nearest Neighbour (95% threshold)	71.43%	20.00%	50.00%	100%	50.00%
K- Nearest Neighbour (IQR threshold)	71.43%	40.00%	75.00%	100%	100%
Local Outlier Factor (95% threshold)	71.43%	40.00%	50.00%	100%	50.00%
Local Outlier Factor (IQR threshold)	71.43%	40.00%	75.00%	100%	100%
	Dogs				
Algorithms	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Isolation Forest (95% threshold)	50.00%	57.14%	33.33%	38.46%	60.00%
Isolation Forest (IQR threshold)	50.00%	28.57%	0%	0%	60.00%
K- Nearest Neighbour (95% threshold)	50.00%	85.71%	33.33%	46.15%	60.00%
K- Nearest Neighbour (IQR threshold)	50.00%	85.71%	33.33%	53.85%	60.00%
Local Outlier Factor (95% threshold)	50.00%	71.43%	0%	46.15%	60.00%
Local Outlier Factor (IQR threshold)	75.00%	85.71%	33.33%	53.85%	80.00%

From the tables 5.3, it is possible to note that Local Outlier Factor and K-Nearest Neighbours are the algorithms that perform best at detecting anomalies for the majority of the animals. However, using LOF with a threshold based on the interquartile range seems the most optimal option when considering all animals.

It is important to note that although both K-NN and LOF appear more accurate in detecting anomalies than the IF, those also produce more false positives, as can be confirmed in the table 5.4. The isolation Forest with the 95% threshold appears to have the best overall performance, if only the FPR is considered, as it maintains relatively low false positive rates across all pets.

The false positives should not be perceived strictly as a negative point in this context, however, it is crucial that this is reported in case of usage.

Table 5.4: False Positive Rates per Algorithm for Daily data with time period

	Cats				
Algorithms	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Isolation Forest (95% threshold)	6.48%	16.67%	19.35%	13.27%	20.00%
Isolation Forest (IQR threshold)	6.48%	16.67%	9.68%	8.16%	31.43%
K- Nearest Neighbour (95% threshold)	8.33%	8.33%	16.13%	13.71%	22.14%
K- Nearest Neighbour (IQR threshold)	10.19%	12.50%	23.39%	34.69%	38.57%
Local Outlier Factor (95% threshold)	8.33%	4.17%	14.52%	17.35%	22.86%
Local Outlier Factor (IQR threshold)	14.81%	37.50%	35.48%	56.12%	45.00%
	Dogs				
Algorithms	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Isolation Forest (95% threshold)	20.00%	27.96%	18.75%	3.49%	18.57%
Isolation Forest (IQR threshold)	13.04%	6.45%	12.50%	0.00%	20.00%
K-Nearest Neighbour (95% threshold)	20.00%	23.66%	20.00%	5.81%	18.57%
K-Nearest Neighbour (IQR threshold)	28.70%	46.24%	30.00%	6.98%	22.86%
Local Outlier Factor (95% threshold)	22.61%	17.20%	16.25%	1.16%	14.09%
Local Outlier Factor (IQR threshold)	33.04%	41.94%	25.00%	3.53%	25.71%

One thing noted during the development of this dissertation is that most false positives in K-NN and LOF aggregations were similar. Even though, as mentioned, IF produces fewer false positives, those are also commonly present in the other aggregations (of K-NN and LOF).

Table 5.5: Precision per Algorithm for Daily data with time period

	Cats				
Algorithms	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Isolation Forest (95% threshold)	41.67%	33.33%	11.11%	27.78%	3.45%
Isolation Forest (IQR threshold)	41.67%	20.00%	14.29%	38.46%	2.22%
K-Nearest Neighbour (95% threshold)	35.71%	33.33%	9.09%	22.73%	3.13%
K-Nearest Neighbour (IQR threshold)	31.25%	20.00%	9.38%	12.82%	3.57%
Local Outlier Factor (95% threshold)	35.71%	33.33%	10.00%	22.73%	3.03%
Local Outlier Factor (IQR threshold)	23.81%	10.00%	6.38%	8.33%	3.08%
	Dogs				
Algorithms	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Isolation Forest (95% threshold)	8.00%	13.33%	6.25%	62.50%	10.34%
Isolation Forest (IQR threshold)	11.76%	25.00%	0.00%	0.00%	9.68%
K-Nearest Neighbour (95% threshold)	8.00%	21.43%	0.00%	54.55%	10.34%
K-Nearest Neighbour (IQR threshold)	5.71%	12.24%	4.00%	53.85%	11.11%
Local Outlier Factor (95% threshold)	7.14%	23.81%	7.14%	85.71%	12.50%
Local Outlier Factor (IQR threshold)	7.32%	13.33%	4.76%	70.00%	7.69%

If we consider the precision, present in table 5.5, i.e., the capacity of a model to correctly identify anomalies we note that the Isolation Forest algorithm with an 95% percentile threshold generally performed better for cats. For dogs, the Local Outlier Factor algorithm with a 95% percentile threshold seems to perform best in general (particularly achieving a high precision of 85.71% for Dog 4). The levels of precision, in general, are not as high as expected, however, this can be due to the quality of the medical records provided and considering that the aim is to prevent illness, false positives may be relevant and something not noted by either the pet or owner.

Table 5.6: F1-score per Algorithm for Daily data with time period

	Cats				
Algorithms	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Isolation Forest (95% threshold)	52.63%	36.36%	19.35%	43.48%	6.45%
Isolation Forest (IQR threshold)	52.63%	20.00%	22.22%	4.26%	2.22%
K-Nearest Neighbour (95% threshold)	47.62%	25.00%	15.38%	37.04%	5.88%
K-Nearest Neighbour (IQR threshold)	43.48%	20.00%	16.67%	22.73%	6.90
Local Outlier Factor (95% threshold)	47.62%	36.36%	16.67%	37.04%	5.71%
Local Outlier Factor (IQR threshold)	35.71%	13.33%	11.76%	15.38%	5.97%
	Dogs				
Algorithms	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Isolation Forest (95% threshold)	11.76%	25.00%	0.00%	0.00%	9.68%
Isolation Forest (IQR threshold)	8.00%	13.33%	6.25%	62.50%	10.34%
K-Nearest Neighbour (95% threshold)	5.71%	12.24%	4.00%	53.85%	11.11%
K-Nearest Neighbour (IQR threshold)	8.00%	21.43%	0.00%	54.55%	10.34%
Local Outlier Factor (95% threshold)	7.32%	13.33%	4.76%	70.00%	7.69%
Local Outlier Factor (IQR threshold)	7.14%	23.81%	7.14%	85.71%	12.50%

The F1-score values per algorithm, as presented in Table 5.6, can also offer relevant insights into the models' performance.

In what concerns the Cats, the Isolation Forest algorithm (specifically the one that uses the 95% threshold) tends to have higher F1-scores compared to the K-Nearest Neighbour and Local Outlier Factor algorithms.

In what concerns the Dogs, the F1-scores vary across algorithms. However, Local Outlier Factor with the IQR-based threshold stands out.

Based on the four evaluation metrics, if the aim is to detect anomalies using the best data representation it is possible to conclude that the most suitable option differs from the type of animal. The aggregation of Isolation Forest with a 95% threshold appears to be the strongest performer, out of all approaches, for the cats, while the aggregation of Local Outlier Factor with an IQR threshold performs the best for Dogs.

To further analyze the performance of the algorithms, an aggregation per algorithm of the results for each animal including all datasets was developed. In the tables 5.7 , 5.8, 5.9 and 5.10, the evaluation metric results for this approach are presented.

From the tables, we can observe the performance of the anomaly detection algorithms aggregation (Isolation Forest, K-Nearest Neighbour, and Local Outlier Factor) in detecting anomalies for cats and dogs.

Regarding true positive rates, the algorithms achieved varying results for the different pets. The Isolation Forest and K-Nearest Neighbour algorithms generally exhibited similar TPR values, while the Local Outlier Factor algorithm seemed to perform better overall in detecting anomalies.

Table 5.7: Sensitivity per Aggregation of Algorithms

	Cats				
Aggregation	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Isolation Forest	57.14%	20.00%	50.00%	80%	0%
K-Nearest Neighbour	42.86%	20.00%	50.00%	80%	0%
Local Outlier Factor	57.14%	20.00%	50.00%	100%	0%
	Dogs				
Aggregation	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Isolation Forest	25.00%	57.14%	0%	23.08%	40.00%
K-Nearest Neighbour	25.00%	42.86%	0%	30.77%	40.00%
Local Outlier Factor	50.00%	57.14%	0%	30.77%	60.00%

The false positive rates varied across the algorithms and pets as well. The Isolation Forest algorithm had relatively low FPRs for cats but slightly higher FPRs for dogs. The K-Nearest Neighbour algorithm had comparable FPRs to the Isolation Forest algorithm for cats but lower for dogs. The Local Outlier Factor algorithm had higher FPRs for cats and dogs in general.

Table 5.8: False Positive Rates per Aggregation of Algorithms

	Cats				
Aggregation	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Isolation Forest	1.85%	0%	4.84%	0%	10%
K-Nearest Neighbour	1.85%	0%	4.84%	1.02%	10%
Local Outlier Factor	1.85%	12.50%	8.87%	2.04%	10.71%
	Dogs				
Aggregation	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Isolation Forest	6.96%	5.38%	8.75%	0%	5.71%
K-Nearest Neighbour	5.22%	4.30%	7.50%	0%	5%
Local Outlier Factor	5.22%	8.60%	8.75%	1.16%	7.14%

When considering precision, Isolation Forest performs the best for Cat 3 and Cat 4, while Local Outlier Factor outperforms the other aggregations for Cat 1 and Cat 2. Regarding dogs, Isolation Forest achieves the highest precision for Dog 2, Local Outlier Factor performs better for Dog 5, K-NN is best for Dog 1. For Dog 3, all algorithms performed poorly.

Table 5.9: Precision per Aggregation of Algorithms

	Cats				
Aggregation	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Isolation Forest	60.00%	0%	25.00%	100%	0%
K-Nearest Neighbour	60.00%	0%	25%	80%	0%
Local Outlier Factor	66.67%	25.00%	15.38%	71.43%	0%
	Dogs				
Aggregation	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Isolation Forest	11.11%	44.44%	0%	100%	20.00%
K-Nearest Neighbour	25.00%	42.86%	0%	100%	22.22%
Local Outlier Factor	14.29%	33.33%	0%	80%	23.08%

The F1-score for cats reveals that the Isolation Forest algorithm performs well for Cat 3 (33.33%) and Cat 4 (88.89%), while Local Outlier Factor shows relatively better results for Cat 1 (61.54%). However, none of the aggregations achieves significant F1-scores for Cat 2 and Cat 5.

For dogs, the performance is more varied. The Isolation Forest algorithm shows relatively good results for Dog 2 (50.00%) in terms of the F1-score. Local Outlier Factor performs better for Dog 5 (33.33%). K-NN ends up standing out for Dog 1 (25.00%) and Dog 4 (47.06%). However, none of the aggregations achieves considerable F1-scores for Dog 3.

Table 5.10: F1- score per Aggregation of Algorithms

	Cats				
Aggregation	**Cat 1**	**Cat 2**	**Cat 3**	**Cat 4**	**Cat 5**
Isolation Forest	50.00%	0%	33.33%	88.89%	0%
K-Nearest Neighbour	50.00%	0%	33.33%	80%	0%
Local Outlier Factor	61.54%	22.22%	23.53%	83.33%	0%
	Dogs				
Aggregation	**Dog 1**	**Dog 2**	**Dog 3**	**Dog 4**	**Dog 5**
Isolation Forest	15.38%	50.00%	0%	37.50%	26.67%
K-Nearest Neighbour	25.00%	42.86%	0%	47.06%	28.57%
Local Outlier Factor	22.22%	42.11%	0%	44.44%	33.33%

Overall, the performance of the algorithms varied across different animal categories.

The Local Outlier Factor algorithm aggregation generally performed well, when using daily data with periods of time, in terms of precision and TPRs, however, had higher FPRs. The Isolation Forest and K-Nearest Neighbour algorithms performed similarly, with variations in TPR, FPR, and precision.

Isolation Forest is the preferred algorithm for anomaly detection when an aggregation of the same algorithm for all datasets is performed. Its ability to achieve high sensitivity, low false positive ratios, and high precision make it a reliable choice for identifying genuine anomalies.

However, if the main goal is to be able to detect the real anomalies then the chosen method should be the Local Outlier Factor, as it also performed well in the second approach.

5.3 Model Explainability

To answer the final research question of this study, the methodology detailed in section (4), was applied to the Local Outlier Factor (IQR-based threshold) aggregation for daily data with periods of time. This particular algorithm aggregation was selected because it demonstrated superior performance when using the best way to represent data for true anomaly detection.

After computing the Shapley values, these were plotted for better interpretation and can be seen in the following figures.

It is important to understand that high Shapley values (either positive or negative) correspond to more relevant features. The number's polarity refers to how it contributes to the target variable.

In this case, high positive Shapley Values correspond to relevant variables in the definition of an anomaly and that positively contribute to higher anomaly scores. High negative values also correspond to relevant variables in the definition of an anomaly, but that contribute negatively to the anomaly score.

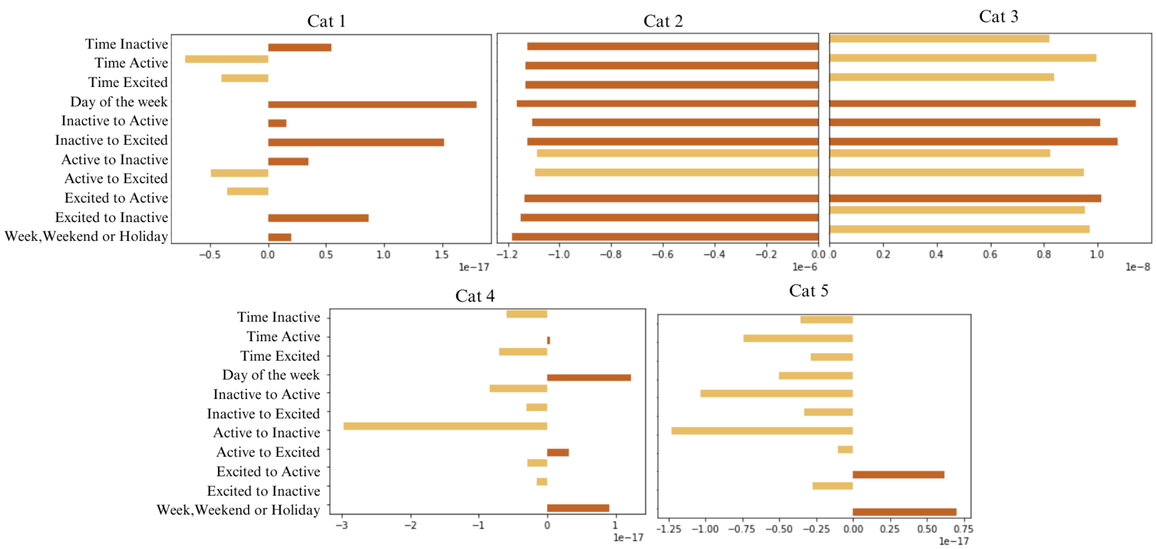

Figure 5.1: Shapley Values - Cats

In Cat 1, 3 and 4, " Day of the week" is significant in defining anomalies since, on average, it contributes positively to the anomaly scores. The transitions between Inactive and Excited contribute to defining anomalies for Cat 1 and 3. In Cat 2, the variables generally have a decreasing impact on the anomaly scores. However, this impact is smaller for "Active to Inactive" and "Active to Excited" being more likely responsible for high anomaly scores. For Cat 4, "Week, weekend, holiday" and "Day of the week" are, on average, the features that determine anomalies. Cat 5 demonstrates that "Excited to Active" and "Week, weekend or holiday" are significant variables in detecting anomalies, as opposed to "Active to Inactive," "Inactive to Active," and "Time Active".

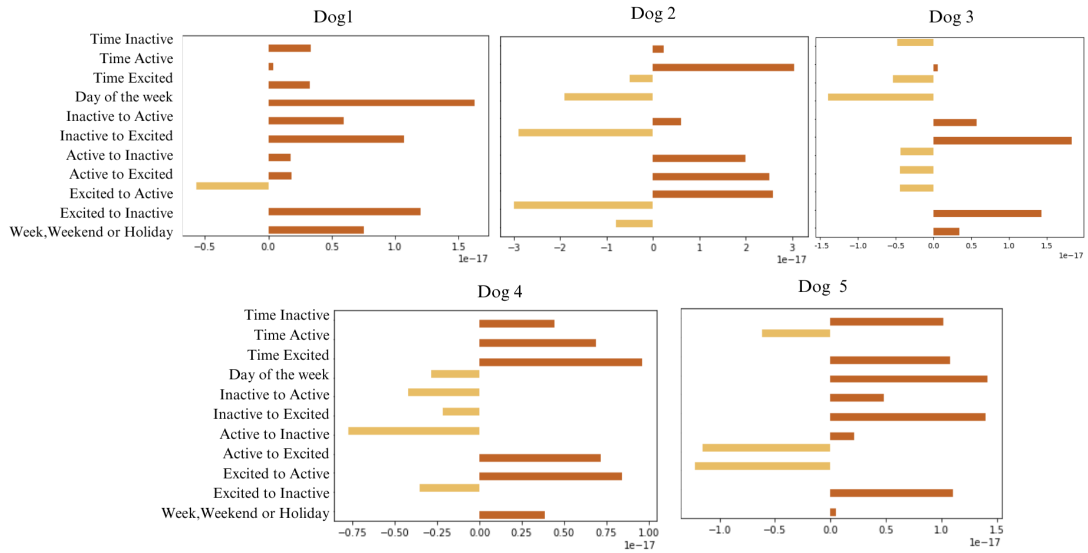

Figure 5.2: Shapley Values - Dogs

Different dogs exhibited various features, on average, relevant to detecting anomalies. For Dog 1, anomalies were associated with temporal patterns and transitions between Inactive and Excited. Dog 2, showed anomalies related to time spent Active and transitions between different states. Transitions from Excited to Inactive and vice versa are the most relevant for Dog 3. Dog 4 displayed diverse features, including Time Excited or Active, transitions between states, and differentiation between weekdays/weekends/holidays. Dog 5 had anomalies linked to Time Inactive or Excited, the day of the week, and transitions between Inactive and Excited.

It is important to note that, although in this case an average for all instances of the feature importance was considered to explain the predictions of the model, the relevance of this technique regards the usage in a business scenario of Shapley Values to understand what caused an anomaly in a certain moment so the vets can further investigate and take action.

5.4 Discussion

To evaluate the results, medical records for ten animals were used, and the true positive ratio, false positive ratio and precision were computed. The evaluation considered detecting health issues and identifying anomalies two days before and two days after their occurrence. The results were presented in tables, showing the evaluation metrics for each animal, dataset, and aggregation technique.

To sum up, the evaluation section provided insights into the performance of the anomaly detection techniques, the best data representation methods, and the effectiveness of ensemble models in detecting anomalies. It was possible to conclude that applying an aggregation of Isolation Forest (with a 95% threshold) to cats and an aggregation of Local Outlier Factor (with IQR as a threshold to separate abnormal from normal instances) to dogs is the most appropriate solution out of the ones proposed for the specific problem. The LOF aggregation also performed well in what concerns detecting anomalies in daily data with periods of time. When the main goal is to detect true anomalies, Local Outlier Factor stands out in all approaches, however, always produces a high number of false positives.

When every way of representing data is considered, in order to identify the most common anomalies detected by each algorithm aggregation, by performing an aggregation for each algorithm results, the Isolation Forest presents good results.

Chapter 6

Conclusion

This chapter provides an overview of the experimental setup, evaluation of anomaly detection models, and the application of Shapley Values. Furthermore, it discusses the limitations encountered during the study and suggests potential areas for future research.

6.1 Final Remarks

This dissertation focused on anomaly detection in pet activity data, addressing several research questions.

Datasets were created with varying time ranges, from the original data.

The experimental setup involved the application of three aggregations of unsupervised anomaly detection models, namely Isolation Forest, Local Outlier Factor, and K-Nearest Neighbours, to six developed datasets. These datasets were used without splitting into train and test sets, as the techniques employed were unsupervised in nature.

Through the evaluation process, it was determined that different data representations play a significant role in detecting anomalies effectively. An ensemble of all algorithm aggregations for each dataset, using majority voting, was performed to identify the most efficient representation. It was found that dividing daily information into time periods yielded the most effective data representation.

The true positive ratios, false positive ratios, precision and F1-score obtained for each dataset and animal were presented, revealing the performance of the algorithms in detecting anomalies. Furthermore, the evaluation included an aggregation of the results across all datasets to evaluate the individual performance of each model. This analysis provided valuable insights into the effectiveness of each technique, regardless of the dataset.

Performance varied across datasets and animals, but the Local Outlier Factor algorithm showed promise when considering daily data divided into periods of time and also when an aggregation of the results across all datasets was performed if the main goal is only to detect true anomalies and false positives are not viewed in a negative way.

When considering all four evaluation metrics and the results obtained for all datasets aggregated by hard vote, Isolation Forest stands out for the great balance in what concerns the precision and recall trade-off.

Overall, the aggregation of the same model with different parameters approach improved accuracy and robustness in anomaly detection, demonstrating the potential for detecting health-related anomalies in pet activity data.

Lastly, the application of Shapley Values to the results obtained with the best data representation and the most successful algorithm aggregation offered an additional understanding of the contributions of each feature in the anomaly detection process.

In conclusion, the findings of this dissertation contribute to the field of anomaly detection by providing a comprehensive evaluation of different techniques and their performance in detecting anomalies in the pet healthcare field. The results demonstrate the importance of selecting appropriate data representations and highlight the potential benefits of aggregating the same algorithm for multiple hyper-parameter values.

6.2 Limitations and Future Work

This project has shed light on various anomaly detection techniques and their performance on the provided datasets. However, some limitations during the development of this dissertation affected the possibility of better outcomes. Moreover, there are several points that could be explored in future research to further improve the task of anomaly detection in pet behavioural data.

In the future, the data used for anomaly detection should be more granular, as in this case, only three options of activity were available, in which one was most prominent for all pets (as noted in 3.3).

Data, such as the axis values of the accelerometer and even the gyroscope that is also incorporated in the collar that the company produces, could be essential for better and more feasible results.

If more granular data would be provided, a topic of interest to develop would be missing values imputation. Since in this type of product, it is quite usual that the pet does not wear the collar for long complete periods of time - something noticed during the process of developing the dissertation.

Further on, an investment in the quality of the medical records should be explored to have a better approach and evaluation metrics for analyzing the results obtained using the models mentioned.

The adoption of techniques focused on time series, such as long-short term memory, would be more in conformity with some of the literature reviewed, however, those also depend on a strong processing power, which the company does not have.

Bibliography

AAVMC. (2021). *Veterinary workforce.* (https://www.aavmc.org/wp-content/uploads/2021/03/AAVMC-Veterinary-Workforce-Report-2021.pdf)

Ahmed, S., Lee, Y., Hyun, S.-H., & Koo, I. (2019). Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest. *IEEE Transactions on Information Forensics and Security*, *14*(10), 2765-2777. doi: 10.1109/TIFS.2019.2902822

Albert, M. V., Kording, K., Herrmann, M., & Jayaraman, A. (2012). Fall classification by machine learning using mobile phones. *PLoS ONE*, *7*, 36556. Retrieved from www.plosone.org doi: 10.1371/journal.pone.0036556

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., … Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, *58*, 82-115. Retrieved from https://app.dimensions.ai/details/publication/pub.1123669031 (https://bird.bcamath.org/bitstream/20.500.11824/1166/1/24.%20main.pdf) doi: 10.1016/j.inffus.2019.12.012

Auret, L., & Aldrich, C. (2012). Interpretation of nonlinear relationships between process variables by use of random forests. *Minerals Engineering*, *35*, 27-42. Retrieved from https://www.sciencedirect.com/science/article/pii/S0892687512001987 doi: https://doi.org/10.1016/j.mineng.2012.05.008

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., … Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, *58*, 82-115. Retrieved from https://www.sciencedirect.com/science/article/pii/S1566253519308103 doi: https://doi.org/10.1016/j.inffus.2019.12.012

Bauder, R., da Rosa, R., & Khoshgoftaar, T. (2018). Identifying medicare provider fraud with unsupervised machine learning. In *2018 ieee international conference on information reuse and integration (iri)* (p. 285-292). doi: 10.1109/IRI.2018.00051

Bhattacharya, A. (2022). *Applied machine learning explainability techniques*. Packt Publishing.

Boukerche, A., Zheng, L., & Alfandi, O. (2020, 6). Outlier detection. *ACM Computing Surveys (CSUR)*, *53*. Retrieved from https://dl.acm.org/doi/10.1145/3381028 doi: 10.1145/3381028

Bouman, R., Bukhsh, Z., & Heskes, T. (2023). *Unsupervised anomaly detection algorithms on real-world data: how many do we need?*

Breiman, L. (1984). *Classification and regression trees* (1st ed.). Routledge. Retrieved from https://

doi.org/10.1201/9781315139470

Budiarto, E., Permanasari, A., & Fauziati, S. (2019, 07). Unsupervised anomaly detection using k-means, local outlier factor and one class svm. In (p. 1-5). doi: 10.1109/ICST47872.2019.9166366

Campos, G., Zimek, A., Sander, J., Campello, R., Micenková, B., Schubert, E., … Houle, M. (2016, 07). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, *30*. doi: 10.1007/s10618-015-0444-8

Candás, J. L. C., Peláez, V., López, G., Ángel Fernández, M., Álvarez, E., & Díaz, G. (2014, 12). An automatic data mining method to detect abnormal human behaviour using physical activity measurements. *Pervasive and Mobile Computing*, *15*, 228-241. doi: 10.1016/j.pmcj.2014.09.007

Chandola, V., Banerjee, A., & Kumar, V. (2009). *Anomaly detection : A survey.*

Chehade, N. H., Ozisik, P., Gomez, J., Ramos, F., & Pottie, G. (2012). Detecting stumbles with a single accelerometer. In (p. 6681-6686). doi: 10.1109/EMBC.2012.6347527

Dataman, C. K. (2023, Jan). *Handbook of anomaly detection: With python outlier detectionnbsp; introduction.* Dataman in AI. Retrieved from https://medium.com/dataman-in-ai/handbook-of-anomaly-detection-with-python-outlier-detection-1-introduction-c8f30f71961c

Dhinakaran, A. (2021, Dec). *What are the prevailing explainability methods?* Towards Data Science. Retrieved from https://towardsdatascience.com/what-are-the-prevailing-explainability-methods-3bc1a44f94df

Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., … Ranjan, R. (2023, jan). Explainable ai (xai): Core ideas, techniques, and solutions. *ACM Comput. Surv.*, *55*(9). Retrieved from https://doi.org/10.1145/3561048 doi: 10.1145/3561048

D'Mello, R., Melcher, J., & Torous, J. (2022, 12). Similarity matrix-based anomaly detection for clinical intervention. *Scientific Reports*, *12*. doi: 10.1038/s41598-022-12792-3

Edinger, J. D., Means, M. K., Carney, C. E., & Manber, R. (2011). Chapter 80 - psychological and behavioral treatments for insomnia ii: Implementation and specific populations. In M. H. Kryger, T. Roth, & W. C. Dement (Eds.), *Principles and practice of sleep medicine (fifth edition)* (Fifth Edition ed., p. 884-904). Philadelphia: W.B. Saunders. Retrieved from https://www.sciencedirect.com/science/article/pii/B9781416066453000803 doi: https://doi.org/10.1016/B978-1-4160-6645-3.00080-3

Ester, M., Kriegel, H. P., Sander, J., & Xiaowei, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Retrieved from https://www.osti.gov/biblio/421283

Fahim, M., & Sillitti, A. (2019). *Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review* (Vol. 7). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ACCESS.2019.2921912

Fuster-Garcia, E., Juan-Albarracín, J., Bresó, A., & García-Gómez, J. M. (2013). *Monitoring changes in daily actigraphy patterns of free-living patients.* Retrieved from http://www.ibime.upv.es

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., Moustakidis, S., Tsaopoulos, D. E., & Plagianakos, V. P. (2022). Deep hybrid learning for anomaly detection in behavioral monitoring. In (Vol. 2022-July). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/IJCNN55064.2022.9892769

Gohel, P., Singh, P., & Mohanty, M. (2021). Explainable AI: current status and future directions. *CoRR*, *abs/2107.07045*. Retrieved from https://arxiv.org/abs/2107.07045

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, *11*(4), e0152173.

Gopinath, D. (2021, Oct). *Picking an explainability technique.* Towards Data Science. Retrieved from https://towardsdatascience.com/picking-an-explainability-technique-48e807d687b9

Gunning, D., & Aha, D. (2019, 06). Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, *40*, 44-58. doi: 10.1609/aimag.v40i2.2850

Habeeb, R. A. A., Nasaruddin, F., Gani, A., Hashem, I. A. T., Ahmed, E., & Imran, M. (2019, 4). Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management*, *45*, 289-307. doi: 10.1016/J.IJINFOMGT.2018.08.006

He, Z., Xu, X., & Deng, S. (2003, jun). Discovering cluster-based local outliers. *Pattern Recogn. Lett.*, *24*(9–10), 1641–1650. Retrieved from https://doi.org/10.1016/S0167-8655(03)00003-5 doi: 10.1016/S0167-8655(03)00003-5

Ho, J., Hussain, S., & Sparagano, O. (2021, 5). Did the covid-19 pandemic spark a public interest in pet adoption? *Frontiers in Veterinary Science*, *8*. doi: 10.3389/FVETS.2021.647308/FULL

Huang, D.-S., Li, K., Irwin, G. W., Zhang, T., Wang, J., Xu, L., & Liu, P. (2006). Fall detection by wearable sensor and one-class svm algorithm. *LNCIS*, *345*, 858-863.

Huang, Y. (2021, Sep). *Demystify your ml model with shap.* Towards Data Science. Retrieved from https://towardsdatascience.com/demystify-your-ml-model-with-shap-fc191a1cb08a

Iivari, A. (2022). *Anomaly detection techniques for unsupervised machine learning* (No. 1324).

John, H., & Naaz, S. (2019). Credit card fraud detection using local outlier factor and isolation forest. *Int. J. Comput. Sci. Eng*, *7*(4), 1060–1064.

Khan, S. S., & Hoey, J. (2017). Review of fall detection techniques: A data availability perspective. *Medical Engineering Physics*, *39*, 12-22. Retrieved from https://www.sciencedirect.com/science/article/pii/S1350453316302600 doi: https://doi.org/10.1016/j.medengphy.2016.10.014

Kieseberg, P., Weippl, E., & Holzinger, A. (2016). Trust for the doctor-in-the-loop. *European Research Consortium for Informatics and Mathematics (ERCIM) News: Tackling Big Data in the Life Sciences*, *104*(1), 32–33.

Koren, O., Koren, M., & Peretz, O. (2023, 1). A procedure for anomaly detection and analysis. *Engineering Applications of Artificial Intelligence*, *117*. doi: 10.1016/J.ENGAPPAI.2022.105503

Krichen, M. (2021, 3). *Anomalies detection through smartphone sensors: A review* (Vol. 21). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/JSEN.2021.3051931

Lee, R. Y., & Carlisle, A. J. (2011, 11). Detection of falls using accelerometers and mobile phone technology. *Age and Ageing*, *40*, 690-696. doi: 10.1093/ageing/afr050

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *Proceedings of the 2008 eighth ieee international conference on data mining* (pp. 413–422).

Ma, M. X., Ngan, H. Y., & Liu, W. (2016). Density-based outlier detection by local outlier factor on largescale traffic data. *Electronic Imaging*, *2016*(14), 1–4.

Mahfuz, S., Zulkernine, F., & Nicholls, P. (2018). *Detecting irregular patterns in iot streaming data for* *fall detection.*

Masís, S. (2021). *Interpretable machine learning with python*. Packt.

Mavuduru, A. (2022, Apr). *How to perform anomaly detection with the isolation forest algorithm.* Towards Data Science. Retrieved from https://towardsdatascience.com/how-to-perform-anomaly-detection-with-the-isolation-forest-algorithm-e8c8372520bc

Medrano, C., Igual, R., Plaza, I., & Castro, M. (2014). Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS ONE*, *9*, 94811. Retrieved from http://ec.europa.eu/esf/home.jsp,the"GobiernodeAragon",http://www.aragon.es/ doi: 10.1371/journal.pone.0094811

Mench, J. (1998). Why it is important to understand animal behavior. *Institute of Laboratory Animal Research*, *39*, 20-26.

Micucci, D., Mobilio, M., Napoletano, P., & Tisato, F. (2017, 2). Falls as anomalies? an experimental evaluation using smartphone accelerometer data. *Journal of Ambient Intelligence and Humanized Computing*, *8*, 87-99. doi: 10.1007/s12652-015-0337-0

Molnar, C. (2023, Mar). *Interpretable machine learning.* Retrieved from https://christophm.github.io/interpretable-ml-book/global.html

Moses, L., Malowney, M. J., & Boyd, J. W. (2018, 11). Ethical conflict and moral distress in veterinary practice: A survey of north american veterinarians. *Journal of Veterinary Internal Medicine*, *32*, 2115-2122. Retrieved from https://www.theatlantic.com/health/archive/2022/07/not-enough-veterinarians-animals/661497/ doi: 10.1111/JVIM.15315

Mougan, C. (2020, May). *Isolation forest from scratch.* Towards Data Science. Retrieved from https://towardsdatascience.com/isolation-forest-from-scratch-e7e5978e6f4c

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). *Machine learning for anomaly detection: A systematic review* (Vol. 9). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ACCESS.2021.3083060

Niu, Z., Shi, S., Sun, J., & He, X. (2011). *A survey of outlier detection methodologies and their applications* (Vol. 7002).

Rajani, N. F., McCann, B., Xiong, C., & Socher, R. (2019). Explain yourself! leveraging language models for commonsense reasoning. *CoRR*, *abs/1906.02361*. Retrieved from http://arxiv.org/abs/1906.02361

Saeed, W., & Omlin, C. (2023). Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, *263*, 110273. Retrieved from https://www.sciencedirect.com/science/article/pii/S0950705123000230 doi: https://doi.org/10.1016/j.knosys.2023.110273

Sgueglia, A., Sorbo, A. D., Visaggio, C. A., & Canfora, G. (2022, 9). A systematic literature review of iot time series anomaly detection solutions. *Future Generation Computer Systems*, *134*, 170-186. doi: 10.1016/J.FUTURE.2022.04.005

Steinbuss, G., & Böhm, K. (2021, apr). Benchmarking unsupervised outlier detection with realistic synthetic data. *ACM Trans. Knowl. Discov. Data*, *15*(4). Retrieved from https://doi.org/10.1145/3441453 doi: 10.1145/3441453

Sundararajan, M., & Najmi, A. (2020, 13–18 Jul). The many shapley values for model explanation. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (Vol. 119, pp. 9269–9278). PMLR. Retrieved from https://proceedings.mlr.press/v119/

sundararajan20b.html

Tana, J., Forss, M., & Hellstén, T. (2017). *The use of wearables in healthcare-challenges and opportunities*.

Tron, T., Resheff, Y. S., Bazhmin, M., Weinshall, D., & Peled, A. (2018, 4). Arima-based motor anomaly detection in schizophrenia inpatients. In (Vol. 2018-January, p. 430-433). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/BHI.2018.8333460

Waele, T. D., Fontaine, J., Eerdekens, A., Deruyck, M., Govaere, J., Joseph, W., … Poorter, E. D. (2021, 6). Internet of animals: Unsupervised foaling detection based on accelerometer data. In (p. 315-320). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/WF-IoT51360.2021.9595172

Wagner, N., Antoine, V., Mialon, M. M., Lardy, R., Silberberg, M., Koko, J., & Veissier, I. (2020, 3). Machine learning to detect behavioural anomalies in dairy cows under subacute ruminal acidosis. *Computers and Electronics in Agriculture*, *170*. doi: 10.1016/J.COMPAG.2020.105233

Xu, M., David, J. M., & Kim, S. H. (2018, 4). The fourth industrial revolution: Opportunities and challenges. *International Journal of Financial Research*, *9*, 90-95. doi: 10.5430/ijfr.v9n2p90

Zhong, S., Fu, S., Lin, L., Fu, X., Cui, Z., & Wang, R. (2019a). A novel unsupervised anomaly detection for gas turbine using isolation forest. In *2019 ieee international conference on prognostics and health management (icphm)* (p. 1-6). doi: 10.1109/ICPHM.2019.8819409

Zhong, S., Fu, S., Lin, L., Fu, X., Cui, Z., & Wang, R. (2019b). A novel unsupervised anomaly detection for gas turbine using isolation forest. In *2019 ieee international conference on prognostics and health management (icphm)* (p. 1-6). doi: 10.1109/ICPHM.2019.8819409

Appendix

.1 Recall for all aggregations of algorithms and data representations

Table 1: Sensitivity/Recall for all models and datasets - Dog 1

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	0%	25%	25%	25%	25%	25%
Daily with Time Period	50%	50%	50%	50%	75%	50%
Daily (Week Days)	25%	50%	25%	25%	50%	50%
Daily (Week Days) Time Period	50%	50%	50%	50%	50%	50%
Daily (Weekend Days)	0%	0%	0%	0%	0%	0%
Daily (Weekend Days) Time Period	0%	0%	25%	0%	25%	0%

Table 2: Sensitivity/Recall for all models and datasets - Dog 2

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	14.28%	28.57%	14.28%	42.85%	28.57%	42.85%
Daily with Time Period	28.57%	57.14%	85.71%	85.71%	85.71%	71.42%
Daily (Week Days)	0%	14.28%	85.71%	14.28%	14.28%	14.28%
Daily (Week Days) Time Period	14.28%	57.14%	57.14%	57.14%	100%	42.85%
Daily (Weekend Days)	0%	14.28%	28.57%	28.57%	28.57%	28.57%
Daily (Weekend Days) Time Period	28.57%	57.14%	85.71%	85.71%	85.71%	57.14%

Table 3: Sensitivity/Recall for all models and datasets - Dog 3

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	0%	0%	0%	0%	0%	0%
Daily with Time Period	0%	33.33%	33.33%	33.33%	33.33%	0%
Daily (Week Days)	0%	0%	0%	0%	0%	0%
Daily (Week Days) Time Period	0%	0%	33.33%	33.33%	33.33%	33.33%
Daily (Weekend Days)	0%	0%	0%	0%	0%	0%
Daily (Weekend Days) Time Period	0%	0%	0%	0%	0%	0%

Table 4: Sensitivity/Recall for all models and datasets - Dog 4

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	15.38%	30.77%	7.69%	23.17%	7.69%	23.17%
Daily with Time Period	0%	38.46%	53.84%	46.15%	53.84%	46.15%
Daily (Week Days)	0%	23.17%	0%	30.77%	15.38%	15.38%
Daily (Week Days) Time Period	0%	38.46%	46.15%	46.15%	46.15%	46.15%
Daily (Weekend Days)	0%	15.38%	7.69%	23.17%	7.69%	23.17%
Daily (Weekend Days) Time Period	0%	38.46%	46.15%	38.46%	46.15%	38.46%

Table 5: Sensitivity/Recall for all models and datasets - Dog 5

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	15.38%	30.77%	7.69%	23.17%	7.69%	23.17%
Daily with Time Period	0%	38.46%	53.84%	46.15%	53.84%	46.15%
Daily (Week Days)	0%	23.17%	0%	30.77%	15.38%	15.38%
Daily (Week Days) Time Period	0%	38.46%	46.15%	46.15%	46.15%	46.15%
Daily (Weekend Days)	0%	15.38%	7.69%	23.17%	7.69%	23.17%
Daily (Weekend Days) Time Period	0%	38.46%	46.15%	38.46%	46.15%	38.46%

Table 6: Sensitivity/Recall for all models and datasets - Cat 1

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	71.42%	57.14%	71.42%	42.85%	71.42%	42.85%
Daily with Time Period	71.42%	71.42%	71.42%	71.42%	71.42%	71.42%
Daily (Week Days)	14.28%	57.14%	71.42%	42.85%	71.42%	42.85%
Daily (Week Days) Time Period	42.85%	71.42%	71.42%	71.42%	71.42%	71.42%
Daily (Weekend Days)	28.57%	14.28%	42.85%	14.28%	57.14%	14.28%
Daily (Weekend Days) Time Period	14.28%	42.85%	57.14%	42.85%	57.14%	42.85%

Table 7: Sensitivity/Recall for all models and datasets - Cat 2

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	20%	20%	20%	20%	20%	20%
Daily with Time Period	20%	40%	40%	20%	20%	40%
Daily (Week Days)	0%	20%	20%	20%	20%	20%
Daily (Week Days) Time Period	20%	20%	20%	20%	20%	20%
Daily (Weekend Days)	0%	0%	0%	0%	0%	20%
Daily (Weekend Days) Time Period	0%	20%	20%	20%	20%	20%

Table 8: Sensitivity/Recall for all models and datasets - Cat 3

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	25%	50%	50%	50%	50%	50%
Daily with Time Period	50%	40%	40%	50%	75%	50%
Daily (Week Days)	0%	50%	50%	50%	50%	50%
Daily (Week Days) Time Period	50%	50%	75%	50%	75%	50%
Daily (Weekend Days)	0%	0%	0%	0%	25%	0%
Daily (Weekend Days) Time Period	0%	0%	50%	0%	0%	25%

Table 9: Sensitivity/Recall for all models and datasets - Cat 4

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	20%	80%	100%	80%	100%	80%
Daily with Time Period	100%	100%	100%	100%	100%	100%
Daily (Week Days)	0%	60%	60%	60%	60%	60%
Daily (Week Days) Time Period	40%	80%	80%	80%	80%	60%
Daily (Weekend Days)	0%	20%	40%	20%	40%	20%
Daily (Weekend Days) Time Period	60%	80%	60%	60%	60%	60%

Table 10: Sensitivity/Recall for all models and datasets - Cat 5

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	0%	0%	0%	0%	50%	0%
Daily with Time Period	50%	50%	100%	50%	100%	50%
Daily (Week Days)	0%	0%	0%	0%	0%	0%
Daily (Week Days) Time Period	0%	0%	100%	0%	50%	0%
Daily (Weekend Days)	0%	0%	0%	0%	0%	0%
Daily (Weekend Days) Time Period	0%	0%	50%	0%	50%	0%

.2 False Positive Rate for all aggregations of algorithms and data representations

Table 11: FPR for all models and datasets - Dog 1

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	0.87%	5.22%	3.48%	4.35%	5.22%	0.87%
Daily with Time Period	13.04%	20.00%	28.70%	20.00%	33.04%	22.61%
Daily (Week Days)	0.87%	2.61%	0.87%	2.61%	3.48%	2.61%
Daily (Week Days) Time Period	15.65%	13.04%	19.13%	13.91%	25.22%	13.04%
Daily (Weekend Days)	0.87%	1.74%	3.48%	1.74%	0.87%	1.74%
Daily (Weekend Days) Time Period	7.83%	6.09%	13.04%	6.09%	12.17%	6.96%

Table 12: FPR for all models and datasets - Dog 2

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	1.08%	4.30%	2.15%	3.23%	5.38%	5.38%
Daily with Time Period	6.45%	27.96%	46.24%	23.66%	41.94%	17.20%
Daily (Week Days)	1.08%	4.30%	2.15%	2.15%	5.38%	4.30%
Daily (Week Days) Time Period	4.30%	18.28%	25.81%	19.35%	31.18%	18.28%
Daily (Weekend Days)	0.00%	2.15%	1.08%	1.08%	2.15%	1.08%
Daily (Weekend Days) Time Period	1.08%	7.53%	8.60%	7.53%	12.90%	7.53%

Table 13: FPR for all models and datasets - Dog 3

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	0.00%	7.50%	8.75%	7.50%	6.25%	7.50%
Daily with Time Period	12.50%	18.75%	30.00%	20.00%	25.00%	16.25%
Daily (Week Days)	0.00%	5.00%	5.00%	5.00%	5.00%	5.00%
Daily (Week Days) Time Period	11.25%	15.00%	20.00%	15.00%	18.75%	16.25%
Daily (Weekend Days)	1.25%	2.50%	6.25%	2.50%	6.25%	2.50%
Daily (Weekend Days) Time Period	6.25%	5.00%	7.50%	5.00%	6.25%	6.25%

Table 14: FPR for all models and datasets - Dog 4

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	1.16%	2.33%	0.00%	0.00%	0.00%	0.00%
Daily with Time Period	0.00%	3.49%	6.98%	5.81%	3.53%	1.16%
Daily (Week Days)	1.16%	1.16%	0.00%	0.00%	0.00%	0.00%
Daily (Week Days) Time Period	0.00%	2.33%	3.49%	3.49%	3.49%	3.49%
Daily (Weekend Days)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Daily (Weekend Days) Time Period	0.00%	1.16%	2.33%	1.16%	3.49%	1.16%

Table 15: FPR for all models and datasets - Dog 5

Datasets	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	0.00%	3.57%	4.29%	3.57%	4.29%	4.29%
Daily with Time Period	20.00%	18.57%	22.86%	18.57%	25.71%	14.09%
Daily (Week Days)	0.00%	2.14%	2.14%	2.14%	1.43%	3.57%
Daily (Week Days) Time Period	12.86%	11.43%	15.71%	12.86%	16.43%	12.14%
Daily (Weekend Days)	0.00%	2.14%	4.29%	2.14%	1.43%	2.14%
Daily (Weekend Days) Time Period	7.86%	7.14%	8.57%	7.14%	7.86%	5.71%

Table 16: FPR for all models and datasets - Cat 1

Dataset	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	7.41%	0.00%	10.19%	0.93%	0.93%	0.93%
Daily with Time Period	6.48%	6.48%	10.19%	8.33%	14.81%	8.33%
Daily (Week Days)	0.00%	0.93%	8.33%	0.93%	8.33%	0.93%
Daily (Week Days) Time Period	6.48%	8.33%	13.89%	7.41%	13.89%	7.41%
Daily (Weekend Days)	0.93%	0.93%	0.93%	0.93%	0.93%	0.93%
Daily (Weekend Days) Time Period	0.00%	0.93%	1.85%	0.93%	1.85%	0.93%

Table 17: FPR for all models and datasets - Cat 2

Dataset	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Daily with Time Period	16.67%	16.67%	12.50%	8.33%	37.50%	4.17%
Daily (Week Days)	0.00%	0.00%	4.17%	0.00%	4.17%	4.17%
Daily (Week Days) Time Period	0.00%	4.17%	8.33%	4.17%	4.17%	4.17%
Daily (Weekend Days)	0.00%	4.17%	0.00%	0.00%	0.00%	0.00%
Daily (Weekend Days) Time Period	0.00%	8.33%	12.50%	8.33%	4.17%	8.33%

Table 18: FPR for all models and datasets - Cat 3

Dataset	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	4.03%	4.03%	4.84%	4.84%	4.84%	4.84%
Daily with Time Period	9.68%	19.35%	23.39%	16.13%	35.48%	14.52%
Daily (Week Days)	0.81%	1.61%	4.84%	2.42%	3.23%	3.23%
Daily (Week Days) Time Period	9.68%	11.29%	16.94%	10.48%	14.52%	11.29%
Daily (Weekend Days)	0.00%	2.42%	2.42%	2.42%	3.23%	2.42%
Daily (Weekend Days) Time Period	5.65%	5.65%	9.68%	4.84%	5.65%	5.65%

Table 19: FPR for all models and datasets - Cat 4

Dataset	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	0.00%	0.00%	2.04%	1.02%	1.02%	1.02%
Daily with Time Period	8.16%	13.27%	34.69%	13.71%	56.12%	17.35%
Daily (Week Days)	0.00%	2.04%	2.04%	1.02%	2.04%	1.02%
Daily (Week Days) Time Period	0.00%	14.29%	27.55%	13.27%	21.43%	13.27%
Daily (Weekend Days)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Daily (Weekend Days) Time Period	1.02%	2.04%	8.16%	1.02%	6.12%	2.04%

Table 20: FPR for all models and datasets - Cat 5

Dataset	Aggregation					
	IF (IQR Threshold)	IF (95% Threshold)	K-NN (IQR Threshold)	K-NN (95% Threshold)	LOF (IQR Threshold)	LOF (95% Threshold)
Daily	16.43%	6.43%	8.57%	6.43%	27.14%	6.43%
Daily with Time Period	31.43%	20.00%	38.57%	22.14%	45.00%	22.86%
Daily (Week Days)	0.00%	5.00%	4.29%	5.00%	2.14%	5.00%
Daily (Week Days) Time Period	2.86%	14.29%	38.57%	15.71%	26.43%	15.71%
Daily (Weekend Days)	6.43%	2.14%	5.71%	2.14%	6.43%	2.14%
Daily (Weekend Days) Time Period	5.71%	5.71%	8.57%	7.14%	11.43%	5.71%