CMP7200 — Individual Master's Project

# Decoding Disease Dynamics

## Machine Learning Classification in Clinical Signals

Alexander Samuel Roberts

M.Sc. Big Data Analytics

Submitted: September 2025

Word count: 12,643

Supervisor: Professor Atif Azad

BIRMINGHAM CITY
University

School of Computing and Digital Technology
Faculty of Computing, Engineering and the Built Environment
Birmingham City University

**Abstract**

For sexually transmitted infections (STIs), such as gonorrhoea and chlamydia, current testing requires patients to submit samples to subsequently be examined in scientific labs, to test to see if they are infected with the disease or not. This process can be time-consuming, taking days to weeks for results to be conversed to the patient and their General Practitioner. Linear Diagnostics, a biochemistry organisation, are attempting to leverage a process called reverse transcription-free exponential amplification reaction (RTF-EXPAR) as a faster and cheaper alternative, which can test for certain STIs, such as those previously listed and more. It is intended to be a part of a new point-of-care-testing (PoCT) initiative. However, the use of machine learning is required to examine and classify the time-series curves recorded from the reaction as either positive or negative. This report investigated how data science and machine learning can be applied to this problem, including evaluating the final model(s) for its explainability and interpretability. Several different approaches, such as using neural networks, statistics and machine learning, dynamic time warping and k-nearest neighbour, and dynamic-sigma time thresholding, were used to tackle this challenge. Results were then subsequently benchmarked against eight classification algorithms. For the COVID-19 dataset, the ridge classifier was considered the best model with consistent 93% accuracy. For the chlamydia dataset, random forest was found to be the best model, maintaining 86% accuracy. This work, a proof of concept in a sense, has enabled Linear Diagnostics to progress the project further towards a production environment.

**Acknowledgements**

I would like to express my deepest thanks and appreciation to my fellow associates, Dipinder Saini and Rahman Ullah. Together, we approached the challenges described in this report from diverse angles and perspectives. It was truly a pleasure to collaborate with them.

I am also sincerely grateful to my project supervisor, Professor Atif Azad, whose continuous support and guidance have been invaluable. His profound insight and thoughtful suggestions helped shape the direction of this project and pushed it toward its full potential.

Finally, I would also like to thank the expert representatives from Linear Diagnostics for their support and feedback throughout this project.

# Contents

# Glossary

| | |
|---|---|
| **AI** | Artificial Intelligence |
| **API** | Application Programming Interface |
| **ANN** | Artificial Neural Network |
| **DL** | Deep Learning |
| **LIME** | Local Interpretable Model-agnostic Explanations |
| **LRP** | Layer-wise Relevance Propagation |
| **MAE** | Mean Absolute Error |
| **MAPE** | Mean Absolute Percentage Error |
| **MCC** | Matthews Correlation Coefficient |
| **ML** | Machine Learning |
| **MLOps** | Machine Learning Operations |
| **MSE** | Mean Squared Error |
| **NRMSE** | Normalised Root Mean Square Error |
| **PCA** | Principal Component Analysis |
| **PII** | Personally Identifiable Information |
| **POCT** | Point-of-care-testing |
| **SHAP** | SHapley Additive exPlanations |
| $R^2$ | Coefficient of determination |
| **RMSE** | Root Mean Square Error |
| **RNN** | Recurrent Neural Network |
| **STI** | Sexually Transmitted Infection |
| **XAI** | Explainable AI |

# List of Tables

# List of Figures

# 1 Introduction

This report is structured as follows. Sections 1.1 to 1.5 will discuss the background information and rationale, problem definition, scope, and aims and objectives of this report. Section 2 will showcase the results of a literature review into the problem domain. Section 3 will establish requirements for a final solution to this report's problem definition, and then initial designs and plans to achieve this. Section 4 will implement the proposed solution, while also noting any new developments or changes required. Section 5 will include the outcomes from the implementation, a discussion regarding the implications or new insights from the results, and an evaluation of the results. Finally, Section 6 will conclude the extent to which this report's aim and objectives have been met, and any additional conclusions. This will also include recommended future work.

## 1.1 Background Information and Rationale

Technological advances within the past few decades have catalysed numerous revolutionary advances across the globe. One particularly important area of advancement is healthcare. More recently, advances in areas such machine learning (ML) and deep learning (DL) have brought about new and improved public health benefits, such as breast cancer screening (Triggle, 2025), the prediction of childhood lead poisoning, the detection of diabetes retinopathy, and helping manage public health emergencies (Alanazi, 2022, p. 100926).

An especially sensitive category of healthcare involves treating sexual transmitted infections/diseases (STIs/STDs). Some of the most widespread STIs are gonorrhoea, chlamydia, syphilis, trichomonas, and genital herpes (Zheng et al., 2022, pp. 541-542). In 2024, in England alone, there were over 70,000 patients diagnosed with gonorrhoea and over 165,000 patients diagnosed with chlamydia (UK Health Security Agency, 2025). These diseases can be very difficult to treat, mainly due to limited treatments, or problems with diagnosis. Some STIs can often be asymptomatic, or have very few symptoms, and hence be undetected until serious symptoms start occurring (Tuddenham, Hamill and Ghanem, 2022, pp. 161-162). Additionally, even if a person may suspect they have a STI, they may feel opposed to seeking treatment due to reasons such as peer pressure or

stereotypical judgements. According to Caruso et al. (2021, pp. 1041-1042), UK (2025) and NHS UK (2025), to test for an STI currently, a doctor would take a sample from the patient, then send this off to a laboratory for testing. However, this can lead to stress for the patient as they await their results, and risks their condition degrading until they can access the proper treatment after the diagnosis is complete. Furthermore, this highlights the value of point-of-care-testing (PoCT), where tests can be quickly completed outside a laboratory (Caruso et al., 2021, pp. 1041-1042).

Linear Diagnostics (2025), a biochemist organisation, is seeking to develop a device that can provide POCT for STIs in General Practices (GPs), namely gonorrhoea and chlamydia. They have successfully developed a method for this. The test is based off of the reverse transcription-free exponential amplification reaction (RTF-EXPAR) put forward by Carter et al. (2022). In essence, the patient sample is reacted with a reagent under special conditions. During the reaction, fluorescent light (measured in relative fluorescent units) is released and monitored to determine how the reaction is progressing. Regular readings are taken, which are subsequently joined to form a time series. The characteristics of the time series determines whether a patient has tested positive for gonorrhoea or chlamydia, or negative otherwise. Additionally, the usage of RTF-EXPAR is not limited to gonorrhoea or chlamydia, it can also be used to diagnose other STIs, or even non-STIs such as COVID-19. Further details of this can be found in the final results by Carter et al. (2022).

## 1.2 Problem Definition

The diagnostic accuracy of the statistical methods is 90% on the data observed; however, the organisation is keen to improve it further to ameliorate patient outcomes. Therefore, ML will be investigated in this report for its effectiveness in being able to accurately classify patient samples for the POCT testing method provided by Linear Diagnostics. Also, the reliability, explainability and interpretability of any trialled ML techniques and processes will be described and examined.

## 1.3 Scope

Some key features of the scope of this project are as follows:

- The provided data from Linear Diagnostics is anonymous and not personally identifiable information (PII).
- Ethical and moral implications from applying Artificial Intelligence (AI) to healthcare data should be evaluated, given any final solution could influence future developments and real-world practical settings.
- Any implemented solution should meet computational resource constraints, which should also be minimised as much as possible. This is for the benefit of current experimentation and future experimentation by others.

## 1.4 Key Scope Limitation

One of the main limitations of the scope of this project is the effects of using disease-focused data. There can be several variants of the same disease(s), caused by mutations and gene variation. Moreover, all diseases can be subject to further mutation, albeit at different rates and significance. This ultimately means that any models developed or insights gained from this report could likely become subject to data drift. However, the methods themselves and the associated analysis would still be invaluable for determining effective approaches and future directions.

## 1.5 Project Aims and Objectives

The aim of this report is to expand research and practical usage of machine learning for diagnosing diseases. The objectives of this report are as follows:

1. Produce a review of the literature that will explore machine learning in healthcare.
    (a) This should include evaluating ethical considerations of applying machine learning for healthcare settings.
2. Establish requirements that must be fulfilled so that any developed implementation meets expectations and requirements. This should be informed based on the results

from the literature review set out in Objective 1, and requirements set out by Linear Diagnostics.

3. Design and plan the features of the final solution to be implemented that meets the requirements set out in Objective 2. This will include a proposed standalone or hybrid machine learning model and associated preprocessing steps.
   (a) Explainability, interpretability and uncertainty must also be included throughout this process.
   (b) An optional addition includes creating a machine learning Operations (MLOps) pipeline to help automate model creation and usage. This would mainly serve as a proof of concept for future development possibilities for Linear Diagnostics.
4. Implement an artifact that meet requirements, and uses the technologies and designs, set out in Objective 3.
5. Evaluate the artifact, both as a whole and the individual components of it. Subsequently, establish conclusions and next steps for refining and optimising the artifact further.
6. Refine the artifact into the final solution based on the conclusions in Objective 5.

# 2 Literature Review

## 2.1 Research Methodology

### 2.1.1 Research Questions and Themes.

Table 2.1 helps show the association between the research questions and their respective themes. These research questions have been thematically devised.

Table 2.1: Research questions and associated themes.

| Question ID (QID) | Research question | Theme |
|---|---|---|
| 1 | What approaches can be taken to classify time-series data? | Time series classification |
| 2 | What synthetic data generation techniques exist to help generate more usable data? | Synthetic data generation |
| 3 | What ethical and moral implications can arise from the use of machine learning in healthcare? | Machine Learning ethics in healthcare |
| 4 | What methods exist to help assess any developed machine learning models for time-series classification? | Model evaluation |

### 2.1.2 Research Scope

The scope of the research material regards its publication date and publication venue. Publications released between January 2020 and May 2025 will be collected. This time period was selected to capture contemporary advancements. This will be conducted during June 2025. The selected publication venues are:

- the Institute of Electrical and Electronic Engineering (IEEE) Xplorer,
- the Association for Computing Machinery (ACM) Digital Library,
- and Science Direct.

### 2.1.3 Search Terms

Table 2.2 declares the search terms that will be used for each research theme.

Table 2.2: Literature search terms.

| Question ID | Search terms |
|---|---|
| QID 1 | (”time series” AND (”classification” OR ”predictive”) AND ”algorithm”) OR (”time series” AND ”explainable AI” AND (”machine learning” OR ”deep learning”)) |
| QID 2 | (”synthetic data generation” OR ”synthetic data”) OR (”time series” AND (”synthetic” OR ”generation”)) |
| QID 3 | ”healthcare” AND ((”AI ethics” OR ”AI morals”) OR (”machine learning” OR ”artificial intelligence”)) |
| QID 4 | (”time series” AND ”classification” AND (”evaluation” OR ”metrics”)) OR (”machine learning” AND ”uncertainty” AND ”evaluation”) |

### 2.1.4 Selection Criteria

Once a collection of sources have been identified, the exclusion criteria described in Table 2.3 will be used to filter the results.

Table 2.3: Exclusion criteria of research publications.

| No. | Criterion |
|---|---|
| EC1 | Papers without full text available. |
| EC2 | Duplicate research. |
| EC3 | Publications not written in English. |
| EC4 | Does not include an abstract or introduction. |
| EC5 | Is unfairly critical or has ambiguous reasoning. |

### 2.1.5 Source Selection

Figure 2.1 shows a Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) diagram of the discovered sources. Additionally, Table B.1 in Appendix B

illustrates a meta-analyses of key sources.

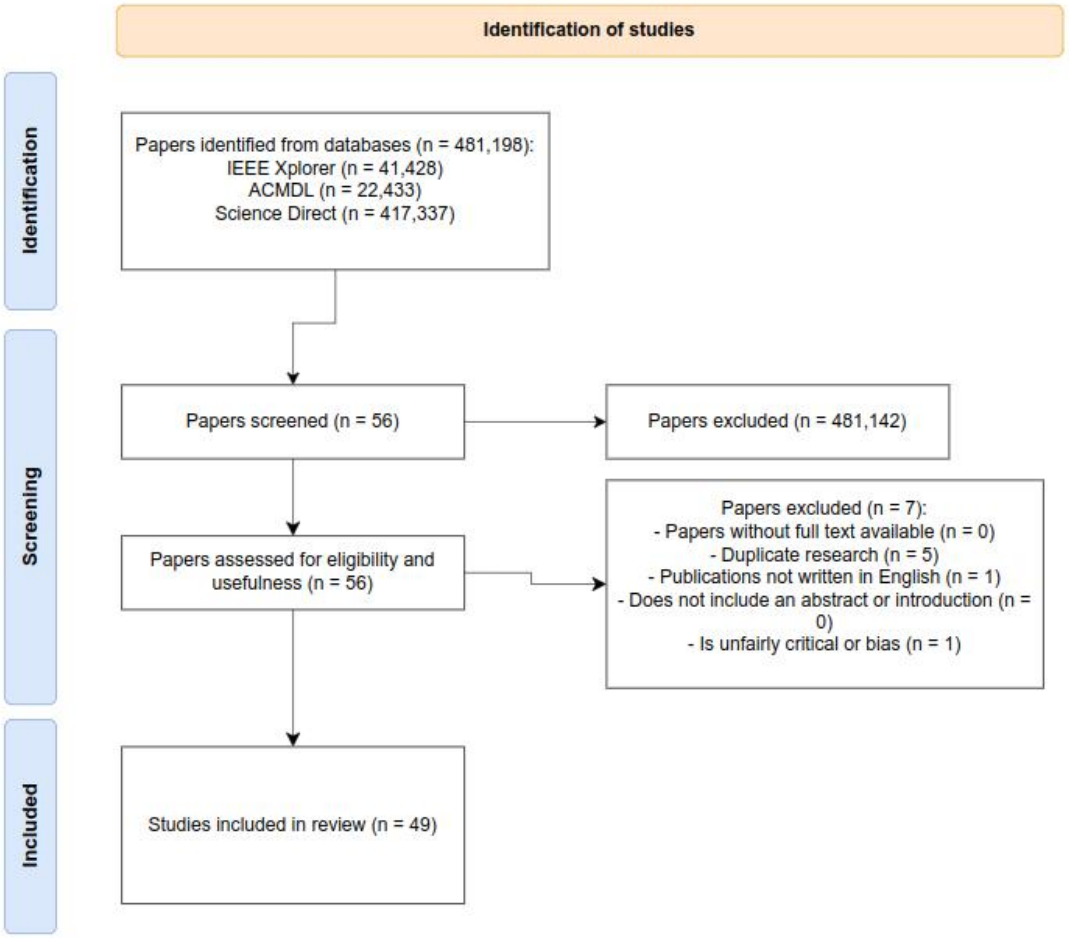


Figure 2.1: PRISMA diagram of identified sources for review of the literature.

## 2.2 Results

### 2.2.1 Time-series Classification

There are a variety of tools and techniques that can be applied for leveraging time series data for analysis and predictive analytics. As highlighted by Syed et al. (2025), both traditional approaches, using statistical models and classical Machine Learning (ML) models, and more modern advancements in artificial neural networks (ANNs) are useful for a range of time series tasks. For instance, they found that autoregressive (AR) models are typically better for fault detection compared to long short-term memory (LSTM)

networks, while LSTM networks outshined AR when it came to sequence prediction and sequence classification. Their systematic research helps validate these findings. In addition, special derivatives of ANN architectures provide key advantages for certain tasks. A sliding-window convolutional variational autoencoder for example, is highly useful for detecting anomalies in time series sequences, as reconstructions by the variational autoencoder will spike. They also implied preprocessing stages can be just as important as the algorithm or model used, as large or even miniscule steps could heavily influence the outcome. For instance, Martínez-Agüero et al. (2022) discusses replacing missing values using zeros, linear interpolation, or statistical imputation. Syed et al. (2025) agrees with using linear techniques (also including exponential smoothing) to fill in missing values. But Syed et al. (2025) disagrees with imputing using zeros, as this can heavily distort the temporal relationships. Finally, Wang and Zhao (2021) demonstrated how dynamic time warping can be used for time-series classification. Their results were novel and presented a different perspective besides traditional time-series methods. Although, the study would have benefitted from including noisy or imbalanced data to see how their evolved dynamic time warping approach held.

Explainability and interpretability are cornerstone for determining the reliability of any developed models. As highlighted by Geyer, Singh and Chen (2024), white-box models (e.g., linear regression) are transparent and very interpretable as the logic of how they came to that conclusion is clear. On the other hand, black-box models (e.g., Random Forest, XGBoost, ANNs) are difficult to interpret and explain due to their process complexity. Geyer, Singh and Chen (2024) and Tanveer and Latif (2024) describe Local Interpretable Model-agnostic Explanations (LIME), SHapley Additive exPlanations (SHAP), and Sensitivity Analysis (SA) are specific methods for analysing feature importance in regard to a model's prediction — enabling explainability through knowing what features the model more heavily relies on. Although, as highlighted by Tanveer and Latif (2024), the aforementioned methods are generalised and can therefore have limited feature importance analysis results. Instead, specifically for ANNs, a decomposition technique called Layer-wise Relevance Propagation (LRP) can be used to determine the most important nodes in a neural network's decision-making process, and feature

importance. Alternative ways of enabling better explainability include structural modifications in the algorithm or model itself. For instance, Bagheri et al. (2025) attempted to use physics-informed ANNs, where they essentially incorporated physics rules directly into the learning process. However, this approach would be difficult to apply to other situations as it would require both specific and proven rules, and specific domain knowledge. An alternative technique by Folgado et al. (2023) involved grouping an ensemble of more explainable and less complexity models together to create a more explainable and less complex overall model. Although, this would require both an abundance of data and careful feature engineering. Despite improved explainability, their final model results evidently had poorer evaluation metrics compared to not applying the method. On the other hand, Folgado et al. (2023) incorporated prediction uncertainty into this approach, as stronger models were based on models with lower prediction uncertainty. Uncertainty can be classed into aleatoric and epistemic. Ways to determine aleatoric uncertainty involve finding the likelihood a given observation belongs to a predicted class based on the class probability distribution, or evaluating the Shannon entropy of the predictive posterior. Other techniques to determine epistemic uncertainty include examining the variation ratio of predictions versus actual values, or comparing predictions against the predictive posterior distribution, assuming a Bayesian perspective.

### 2.2.2 Synthetic Data Generation

Synthetic data generation has a range of applications. As explained by Hernandez et al. (2022), synthetic data can help resolve missing values, mitigate target class imbalances, and help with data pooling of privacy-preserved data publishing. A variety of techniques, from multiple linear regression to more complex methods such as generative adversarial networks (GANs), are useful for synthetic data generation. Hernandez et al. (2022) proposed some standardisation categories for classifying how useful synthetic data is, such as its resemblance to the original data, its privacy compliance, and its actual utility. However, despite their systematic approach, their research was broad and lacked in-depth discussion on key areas such as preprocessing and specific use cases of specialised architectures. For example, Wasserstein Generative Adversarial Networks

with Gradient Penalty (WGAN-GP) is a good approach for data augmentation and generation as it helps address critical generative adversarial network (GAN) problems such as mode collapse, non-convergence and decreasing gradients (König et al., 2024). König et al. (2024) utilised another special architecture, consisting of the Long Short-Term Memory (LSTM) structure and GAN to form LSTM-GAN, for creating synthetic time series data. After principal component analysis (PCA), it was apparent the synthetic data points were discernible from the real data — despite looking visually acceptable and conclusive when superimposed on the original data. In addition, their final results of 100% accuracy (from 65% previously) for their classification task is inconclusive, as the test samples were limited. Besides direct visualisations, some evaluation metrics for determining the similarity between the original and synthetic data include missing values similarity, which measures how close percentages missing values are in the synthetic data and real data, and statistical similarity, which measures the similarity between the distributions using the standard deviation and variance (Ranja, Nababan and Candra, 2023). Moreover, other metrics specifically for time-series data include Jensen-Shannon divergence, Wasserstein distance, Rajski distance, and dynamic time warping (Ungureanu et al., 2024).

### 2.2.3 Machine learning ethics in healthcare

Gani et al. (2024) spotlights the key advantages Artificial Intelligence (AI) can play in healthcare settings besides that of disease diagnosis, including improving productivity by handling more basic tasks by healthcare professionals, such as automatic chart generation, AI-assisted documentation, and data entry processing. Moreover, some of these features are already being leveraged in areas such as radiology, robotic surgery, and diabetic retinopathy. However, they point out that over 64% of medical professionals have never encountered AI before. Furthermore, over 85% of those who have do not know the difference between ML and DL. It is important to note, that their study did not investigate AI training programmes and policies to guide and advice the usage of AI. On the other hand, Sahoo et al. (2025) explains that a key problem with using AI for more crucial tasks are transparency and traceability of its predictions. The issues

presented by Gani et al. (2024) and Sahoo et al. (2025) are what limit AI usage in healthcare. Some white-box approaches (e.g., multiple linear regression) are much more likely to be used due to their explainability. Especially because, as shown by Costa and Georgieva (2023), their exact processes can be combined with statistical methods (such as hypothesis testing) to help validate predictions.

### 2.2.4 Model evaluation

As demonstrated by Fang et al. (2025), common evaluation metrics for classification tasks include accuracy, precision, recall, F1-score, and Log Loss (for neural networks specifically). However, for specific aspects of any developed system, pre-existing metrics can be customised to suit the circumstances. For example, Lee et al. (2025) uses a custom uncertainty quantification based on the predictive posterior, and an informativeness score, in line with their use of the Batch Active Learning Method for Time-Series Classification with Multi-Mode Exploration (BALT). Additionally, they illustrated the importance of using statistical tests in their results to help validate the findings, while also benchmarking the results against existing techniques where possible. They unfortunately did not investigate how differences in the datasets themselves had an effect on the overall results. Blasco, Sanchez and Garcia (2024) made it evident that the evaluation stage may be spread somewhat across the overall process or pipeline.

## 2.3 Summary

To summarise, for synthetic data generation, simpler and more interpretable methods should be trialled first, but utilising more complex GAN models could also be done to help improve efficacy. Ultimately, model explainability is key for facilitating the use of these models in practical settings. There was little research on automating and scaling the processing of time series healthcare data using machine learning operations. This report will help fill this gap.

# 3 Implementation Design and Methods

## 3.1 Methodology

There are two common implementation methodologies used for project management: the waterfall method and the agile method. The former is a more straightforward approach for taking on projects with clear objectives and aims from the beginning, while the latter is designed for flexibility to account for changes in requirements. Figure 3.1 visualises these processes. The Waterfall method will be used for this project as the stakeholders (Linear Diagnostics) have set out clear and concrete aims which will not be significantly changed (as discussed in Section 1.2).

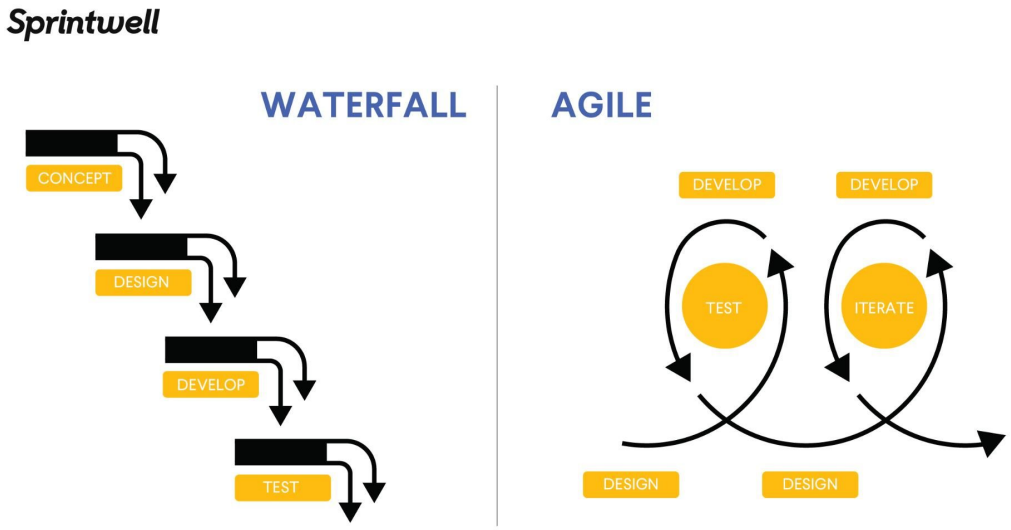


Figure 3.1: Waterfall vs. Agile project management

Source: AIO Photos (2025).

## 3.2 Data Collection

Linear Diagnostics have supplied three key datasets, of which, none contain any personally

- Chlamydia trachomatis (CT) dataset (Dataset 1): this contains time-series data

regarding the reaction described in Section 1.1 for a group of patients who tested positive or negative for CT. This is real-world data that has been completely anonymised; only the clinical signals, the region of the patient's body where the sample originates from, and gender information are included.

- COVID-19 dataset (Dataset 2): this contains time-series data related to the experiments done by Carter et al. (2022) for patients potentially infected with COVID-19. These are real-world clinical samples. No additional information regarding the samples is included (e.g., gender, region, ethnicity).

- Synthetic experimental dataset (Dataset 3): similarly to the COVID-19, this dataset is directly related to the experimentation done by Carter et al. (2022). However, this data is synthetic as it was synthesised entirely in a lab. The main focus It may be referred to as Proceedings of the National Academy of Sciences (PNAS) data.

Figures A.1, A.3 and A.4 show a snapshot of each dataset respectively in their original state. As shown in Figure A.1, the 'Type of Sample' refers to where the sample was collected from. This is as follows: vaginal (V), high vaginal (HV), endocervical (EC) and male urine (M). The former three refer exclusively to females. Furthermore, each patient sample (denoted in the format '1-320-129' for example) contains four time series. Each of these time series originates from the same patient sample and have been reacted on the same day at the same time, side-by-side. Other samples that have a suffix of '-1' or '-2' or equivalent on the end are considered duplicates, where they originate from the sample patient sample, but have been reacted at different times (potentially on different days). Finally, for this dataset, LD have advised to ignore the 'seconds' column as the varying times should be in increments of 30 seconds, rather than examples such as 462 seconds. Figure A.2 shows additional information that was extracted from the original CT dataset, but is not apparent in Figure A.1. For Figure A.3, each column refers to a single time-series sample, which has only been tested once. All samples are split into three batches (plates), as only approximately 96 samples could be reacted at a time. This should not impact the results in particular, however. Column names such as 'A2',

'A3', and so on, do not hold any significance as they are just label names. Finally, for Figure A.4, labels, which represent concentrations, such as '1450 Copies/ $\mu$L' signifies 1450 copies of ribonucleic acid (RNA) per microlitre. There are three time series per concentration, which were recorded at the same time.

The main focus will be investigating and creating successful models for Datasets 3.2 and 3.2, as LD preferred to prioritise investigating these datasets since they contain information that would be leveraged in real-world, practical settings. Dataset 3.2 will be used to provide additional insights.

## 3.3 Limitations and Options

### 3.3.1 Data

The limited data available (even with all three datasets) is a major limitation to model creation. Limited data could obscure or present misleading patterns, reduce results significance, and makes the application of DL difficult. One way to try to resolve this is to use external data candidates to help supplement the ground truth data, assuming they operate under similar conditions. Unfortunately, given the specific nature of the data, no extra datasets were found. Alternatively, the data could be synthetically generated, as mentioned in Section 2.2.2. For mimicking entire sequences, it could likely require more complex methods such as LSTM-GAN. Evaluation methods such as a missing values summary, statistical summary, dynamic time warping (DTW), measuring the Wasserstein distance, and a visualisations break-down would need to be implemented to measure the synthetic sequence success. For other aspects such as missing value imputation or sequence padding, it would preferably be better to use more white-box approaches such as linear interpolation, or last observation carried forward (LOCF), or exponential smoothing, due to their excellent explainability and transparency.

Another less problematic issue would be the format of the time series. Each sequence is currently in a wide-column format, which is easier for human-level interpretability, but this is not ideal for programmatic data manipulation purposes. Therefore, each sequence must be transformed into a long-column format, where each row represents a specific data point along a time series for a specific sample or observation. Figure 3.2 visualisations

the end result of this process. Tables C.2, C.1, C.3 and C.4 illustrate the statistical data types for each feature in the long-column format for each dataset.

| | | Type of Sample | V | V | V | V | EC | EC | EC | EC | EC | EC | EC | EC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time Axis | | Sample Number | 1-320-129 | | | | 1-320-132-1 | | | | 1-320-132-2 | | | |
| Mins | Seconds | | | | | | | | | | | | | |
| 0 | 0 | | 3.01129 | 2.885547 | 2.875749 | 2.934538 | 2.238667 | 2.195359 | 2.008804 | 2.012135 | 2.613958 | 2.703352 | 2.999678 | 3.011266 |
| 0.5 | 30 | | 3.09784 | 2.967198 | 2.970464 | 3.022721 | 2.350267 | 2.305294 | 2.100416 | 2.107079 | 2.76957 | 2.877174 | 3.198332 | 3.198332 |
| 1 | 60 | | 3.141931 | 3.009657 | 3.02762 | 3.068446 | 2.43022 | 2.396906 | 2.180368 | 2.190362 | 2.913594 | 3.02782 | 3.360566 | 3.3556 |
| 1.5 | 120 | | 3.186023 | 3.058647 | 3.088042 | 3.11417 | 2.560142 | 2.543485 | 2.301963 | 2.313622 | 3.148668 | 3.272827 | 3.598951 | 3.579086 |
| 2 | 150 | | 3.228482 | 3.106005 | 3.158262 | 3.166427 | 2.766686 | 2.775014 | 2.495181 | 2.508506 | 3.380432 | 3.493002 | 3.762841 | 3.734698 |
| 2.5 | 189 | | 3.251344 | 3.132133 | 3.203986 | 3.190922 | 2.941582 | 2.97323 | 2.701725 | 2.716716 | 3.521145 | 3.63206 | 3.875411 | 3.840647 |
| 3 | 228 | | 3.27094 | 3.154996 | 3.236647 | 3.213784 | 3.041522 | 3.088161 | 2.868292 | 2.884949 | 3.643648 | 3.757874 | 3.983016 | 3.943285 |
| 3.5 | 267 | | 3.290536 | 3.172959 | 3.266041 | 3.235014 | 3.12314 | 3.18477 | 2.979892 | 2.994883 | 3.764496 | 3.873756 | 4.080687 | 4.039301 |
| 4 | 306 | | 3.310133 | 3.194188 | 3.297068 | 3.256243 | 3.206424 | 3.276382 | 3.074836 | 3.093158 | 3.875411 | 3.986327 | 4.176704 | 4.12704 |
| 4.5 | 345 | | 3.332995 | 3.21705 | 3.326463 | 3.277472 | 3.279714 | 3.361332 | 3.168114 | 3.181439 | 3.98136 | 4.09062 | 4.266098 | 4.214779 |
| 5 | 384 | | 3.352591 | 3.23828 | 3.349325 | 3.295435 | 3.349672 | 3.447947 | 3.253063 | 3.271385 | 4.083998 | 4.194914 | 4.352181 | 4.294241 |
| 5.5 | 423 | | 3.37382 | 3.259509 | 3.377086 | 3.318298 | 3.41963 | 3.526234 | 3.338012 | 3.356335 | 4.18167 | 4.29093 | 4.438265 | 4.373702 |
| 6 | 462 | | 3.393417 | 3.280738 | 3.404848 | 3.342793 | 3.482926 | 3.606186 | 3.416299 | 3.441284 | 4.274375 | 4.380324 | 4.511105 | 4.448198 |
| 6.5 | 501 | | 3.416279 | 3.301967 | 3.430976 | 3.362389 | 3.551219 | 3.67781 | 3.499583 | 3.517905 | 4.358803 | 4.468063 | 4.5856 | 4.517727 |

(a) Wide-column format.

| mins | seconds | result | replicate | sample_id |
|---|---|---|---|---|
| 0 | 0 | 3.01129 | 0 | 1-320-129 |
| 0 | 0 | 2.885547 | 1 | 1-320-129 |
| 0 | 0 | 2.875749 | 2 | 1-320-129 |
| 0 | 0 | 2.934538 | 3 | 1-320-129 |
| 0 | 0 | 2.238667 | 0 | 1-320-132 |
| 0 | 0 | 2.195359 | 1 | 1-320-132 |
| 0 | 0 | 2.008804 | 2 | 1-320-132 |
| 0 | 0 | 2.012135 | 3 | 1-320-132 |
| 0 | 0 | 2.613958 | 0 | 1-320-132-dup |
| 0 | 0 | 2.703352 | 1 | 1-320-132-dup |
| 0 | 0 | 2.999678 | 2 | 1-320-132-dup |
| 0 | 0 | 3.011266 | 3 | 1-320-132-dup |
| 0 | 0 | 3.009828 | 0 | 1-320-133 |
| 0 | 0 | 2.627494 | 1 | 1-320-133 |
| 0 | 0 | 2.699918 | 2 | 1-320-133 |
| 0 | 0 | 2.479276 | 3 | 1-320-133 |
| 0 | 0 | 2.94963 | 0 | 1-320-135 |
| 0 | 0 | 3.015896 | 1 | 1-320-135 |
| 0 | 0 | 2.846191 | 2 | 1-320-135 |
| 0 | 0 | 2.818715 | 3 | 1-320-135 |
| 0 | 0 | 2.721877 | 0 | 1-320-136 |
| 0 | 0 | 2.468006 | 1 | 1-320-136 |
| 0 | 0 | 2.435871 | 2 | 1-320-136 |
| 0 | 0 | 2.434264 | 3 | 1-320-136 |
| 0 | 0 | 2.895903 | 0 | 1-320-136-dup |
| 0 | 0 | 2.700893 | 1 | 1-320-136-dup |
| 0 | 0 | 3.003159 | 2 | 1-320-136-dup |
| 0 | 0 | 2.783772 | 3 | 1-320-136-dup |
| 0 | 0 | 2.133994 | 0 | 1-320-137 |
| 0 | 0 | 2.145345 | 1 | 1-320-137 |

(b) Long-column format.

Figure 3.2: Sample of transforming the CT/MG dataset from a wide-column format to a long-column format.

### 3.3.2 MLOps Pipeline

Figure 3.3 visualises a proposed Machine Learning Operations (MLOps) pipeline implementation for automating model development and monitoring over time. Table 3.3 describes the purpose of each technology in more detail. However, this custom pipeline implementation would lack flexibility for usage by non-technical users due to the amount of extensive modifications required — which would be outside the scope of this report. Alternative solutions for more practical flexibility and more support include mainstream cloud solutions, such as Microsoft Azure (Microsoft, 2025), Google Cloud (Google, 2025), and Amazon Web Services (Amazon Web Services, 2025). Hence, this pipeline will not be implemented within this report as it would not provide any additional benefits nor insights.

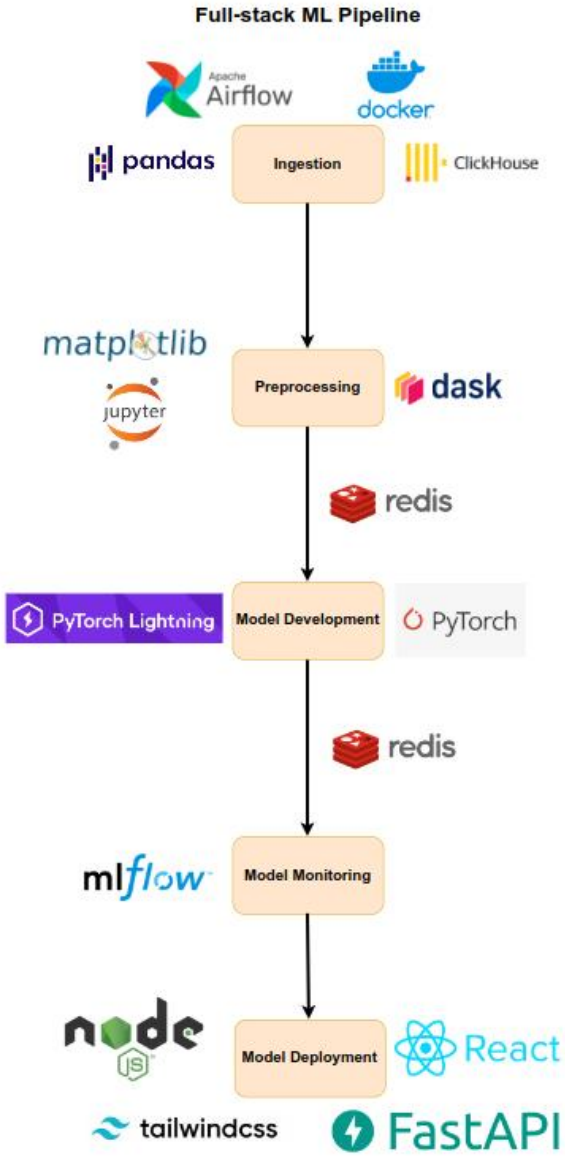


Figure 3.3: Proposed full-stack MLOps pipeline.

## 3.4 Exploratory Data Analysis

Before commencing exploratory data analysis (EDA), it is important to split the available data into train-validation-test sets to help avoid train-test data leakage. Stratified sampling of the outcome class will also be used to help avoid sampling bias. The split will be 70% for training, 10% for validating, and 20% for testing. This is because any insights

gained from the EDA that includes the testing set will lead to bias in the decision-making and analysis. To clarify, the testing set is meant to remain an "unseen" data until the model evaluation. Moreover, each dataset will be examined separately. Sections 3.4.1 to 3.4.3 will highlight key points, while Appendix D will contain any referenced EDA figures and additional EDA.

### 3.4.1 Dataset 1: Chlamydia

There are apparent varying differences in sequence lengths for the Chlamydia data. This is highlighted in the pie chart in Figure D.1 and the unscaled curves in Figure D.2. All sequences must be the same length, not only for data integrity purposes, but to help prevent target data leakage, which is where data that would not be available at prediction time is used during ML training. This would be caused by two reasons: using differing sequence lengths and using data outside a set cut-off. For the former, different sequence lengths invite bias into the process as, for example, longer sequences provide more information, which an ML algorithm could then use to infer trends regarding shorter sequences. Shorter sequences could also disrupt the training as inferences by the algorithm could be unfairly discarded due to more erratic behaviours discovered in shorter sequences. Furthermore, Linear Diagnostics would like the tests to have a set amount of time required to complete. For instance, for a Chlamydia sample, 25 minutes worth of data points will be used for prediction. Therefore, using 30 minutes worth of data points would be outside this range, and hence, be considered target data leakage. To resolve, all the sequences will be cut to 22.5, as the majority of cases exceed this. Any sequences shorter than this (such as 19 minutes) will be discarded. Moreover, this will be done for the validation and testing sets as well. This effectively means the majority of data is retained for training, but it is important to acknowledge that some patterns may be lost because of this. Figures D.3 and D.4 show the resulting curves from applying this process, but the former curves have been normalised, while the latter curves have been standardised.

Although there are no noticeable outliers present in a visual examination of Figures D.3 and D.4, it is important to still do outlier detection (as not all outliers are visually

discernable). One approach for this is leveraging the three-sigma rule (Nascimento et al., 2021, pp. 8326-8327), which assumes samples at the extreme ends of a distribution are outliers. This is shown in Figure 3.4, where 99.7% of data is usually within three standard deviations of the mean. Unfortunately, this method cannot directly be applied to the time series themselves. Instead, for each time series, the standard deviation, the absolute mean difference across the sequence, and the autocorrelation between adjacent data points will be used. The last method works by measuring the Pearson correlation of all the differences between each adjacent data point. The correlation is given as a single value. The differences, essentially the different rates at which the sequences progress, should form a sort of normal distribution, which is why the Pearson correlation method is applicable (as it expects the data to be normally distributed). Theoretically, the Pearson correlation for similar curves should be the same, regardless of what correlation value it has. For more irregular curves, their correlation value will be significantly different. Moving forwards, for each metric, the values are standardised across the sequences, then the series which fall outside the mean plus three standard deviation range, for any of the metrics, are considered outliers, and are subsequently removed. Figures D.5 and D.6 show the results of applying the three-sigma rule to the data, where the former represents the data that has been normalised, and the latter represents data that has been standardised. For the normalised data, 19 outliers were detected, while 16 were detected for the standardised data. However, only three samples were present in both sets, all other outliers were different. This difference is due to the scaling method used. Standardisation can distort the originality of the sequences as it discourages more significant shifts in values (as they become reshaped to fit a normal distribution). This does not necessarily mean either method should be ruled out yet.

Figure D.8 shows the normalised mean curves by gender. As shown in the figure, both the positive and negative female curves progress very similarly, while there is a clear difference between the mean curves for the positive and negative male curves. This suggests female samples could be harder to diagnose compared to male ones. Techniques such as dynamic time warping (DTW) could be leveraged to exploit these differences. This will be discussed further in Section 3.6.

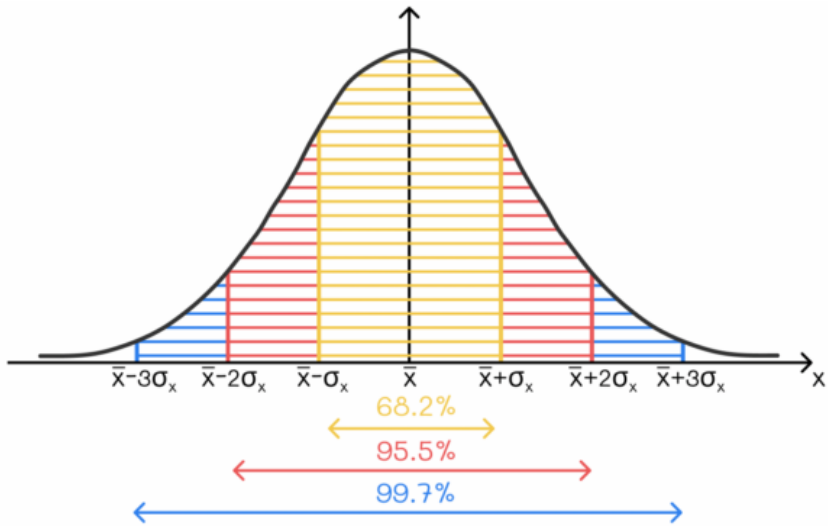


Figure 3.4: The three-sigma rule visualised.

Source: Efimov (2024).

### 3.4.2 Dataset 2: COVID-19

Figure D.10 shows all the COVID-19 curves, of which, have been normalised — so scale does not distort trends (as shown in Figure D.9). As shown in Figure D.10, there are numerous outliers present. This is also apparent but less evident in Figure D.11, where the curves have been standardised (using Z-score standardisation) instead of normalised. The same outlier removal method discussed in Section 3.4.1 will also be applied here. However, an additional outlier removal step will be used which leverages the total amount of variation is present across the time series. This works by taking the absolute difference between each data point, then totalling them up. More erratic curves will have a much higher total variation compared to non-outliers. However, this method requires subjective input to determine what percentage of the highest variation scorers should be removed, as the curves must be visually examined. In this case, the top eight percent were assumed to be outliers, and hence removed. Figures D.12 and D.13 illustrates the results of applying these two outlier methods. It is very noticeable that applying outlier removal has mostly removed the visual outliers. For both the normalised curves and standardised curves, sixteen outliers where detected. The seemingly few outliers that remain should not impact results tremendously, and may help improve generalisation as

the algorithms are exposed to a wider range of sequences.

As shown in Figure D.14, which shows a pie chart of the distribution between positive and negative samples, the class distributions are unequal. Random undersampling will be used to balance this difference, with reasoning similar to Section 3.4.1. Interestingly, as shown in Figure D.15, which shows the mean curve trends for positive and negative samples, the positive curves tend upwards sooner compared to negative curves. Similarly to Chlamydia, DTW could be used to exploit this difference.

### 3.4.3 Dataset 3: PNAS

Given the main amount of data within Dataset 3.2 regards positive artificial COVID-19 samples, no data splitting will be performed. Figure D.17 shows the curves with no data scaling. Interestingly, the shape is not too dissimilar to the normalised and standardised curves in Figures D.18 and D.19 respectively. This implies the trends in the data are fairly consistent.

This dataset will not be used for ML development as it does not supply enough negative sample cases to train an ML algorithm to differentiate. Although, this data could be useful for examining the effect of different concentrations of samples on reaction volatility, this would fall outside the scope of this report. Additionally, as it is synthetic data that was created under controlled conditions, any insights gained may be misleading when applied to real patient sample data.

## 3.5 Data Preparation

As detailed by Anderson (2024), data preparation is a series of stages, such as cleaning, transforming, merging, analysing, interpreting, critiquing, and more, to transform raw data into useful processed data. The figures in Appendices ... and ... indicate some key data preprocessing steps that would be required, such as those listed below.

- Data scaling: to remove scale from the data, so that developed models do not examine the orders of magnitude of difference as significant. This should be done using normalisation (to match experiments carried out by Linear Diagnostics), but also with standardisation to help mitigate outliers.

- Label encoding: encoding each sequence class from 'positive' and 'negative' to '1' and '0'. This is so they can be processed by ML algorithms.
- Class balancing: to prevent majority bias, each target class should have equal amounts of instances. For undersampling the majority class, random undersampling should be used as it presents no bias into selection. For oversampling the minority class, this should be done using either Synthetic Minority Oversampling Technique (SMOTE) for averaged sequence metrics, but not for creating sequences themselves.
- Feature engineering: selecting specific attributes, or creating new ones through data enrichment techniques (such as dimensionality reduction). This will be discussed further in Section 3.5.1.
- Train-validation-test splitting: an essential method for ML modelling. However, this should also be done for any exploratory data analysis (EDA) steps to help prevent train-test data leakage. This is because any findings would have been influenced by the testing set — which is meant to represent "unseen" data. This is not just for ML models, as insights originating from the training set can be compared against the testing set. The train-validation-test split will be 70-10-20 (70% train, 10% validation, 20% test).
- Randomisation seeding: initialise a constant random seed to help improve reproducibility. This will be set to '42' across all seeds. It is important this is not changed especially during the data preparation stages, as it could lead to train-test data leakage when getting insights into the data (as mentioned above).

### 3.5.1 Feature Engineering

Feature engineering is one of the most important stages during the data transformation process. For example, insights from a dataset that consists of ten "weak" (random) attributes and three "strong" (highly correlated) attributes can be greatly distorted by the weaker attributes in relation to examining specific relationships. For investigating the importance of independent variables (both categorical and continuous) in relation

to a target variable, a few methods can be trialled. These include correlation analysis, Mutual Information Gain, and ML-related techniques, such as permutation importance. Ideally, the best set of features are deduced from combing multiple methods together — as only examining one can be misleading. The first two approaches are more consistent than ML feature reduction as, depending on the approach and configuration, the latter adjusts the set of features to suit that specific ML model instance. However, this randomness can be mitigated by keeping constant randomisation seeding and performing hyperparameter tuning. As described by Dasari and Devarakonda (2022)[p. 63], Pearson, Spearman, and Kendall Tau correlation are the most common correlation analysis methods. Pearson correlation analysis assumes the data is normally distributed, which can be problematic for time-series sequences depending on what key metrics are analysed. However, a simple statistical hypothesis test such as the Shapiro-Wilk test (Monter-Pozos and González-Estrada, 2024)[p. 115650-115651] can be used to determine if the data is normally distributed. It is a one-tailed test where the null hypothesis assumes the data is normally distributed. Otherwise, non-parametric approaches such as Spearman and Kendall Tau can be employed. Kendall Tau is more robust and efficient than Spearman for smaller datasets (Dasari and Devarakonda, 2022)[p. 63]. Therefore, Kendall will be used, particularly as it is more robust to outliers as it focuses less on the magnitude of rank differences.

### 3.6 Model Development

There are several strategies that can be taken to classify a time series. For example, many traditional approaches involved taking averaged or representative statistics of the sequences, then feeding the insights gained into ML models. If these statistics had a significant degree of linear correlation, linear models such as simple linear regression could be used. Moreover, if more stable training was required (by mitigating exploding gradients), regularised linear approaches such as L1-regularisation (Lasso) and L2-regularisation (Ridge) could be applied. Furthermore, generalised linear models such as logistic regression, which are more frequently applied to probability-based problems, are very applicable to binary classification problems. Given all these algorithms follow a set

of clear and intuitive steps, they are both easily explainable and interpretable. Unfortunately, the aforementioned statistics may not have a linear relationship; they could have a quadratic or non-linear pattern, especially if the data operates in a higher dimensional space. ML models such as decision trees, k-nearest neighbour, and Naive Bayes algorithms can handle and interpret non-linear data while also remaining explainable and interpretable. Support vector machines (SVMs) innately operate in linear spaces, but they can also work in non-linear space using the kernel trick (Du et al., 2024, pp. 3935-3938).

More recently, as described in Section 2.2.1, recurrent neural networks (RNNs) such as LSTM networks can analyse the time series directly, removing the need to do most of the feature engineering all together. However, while ANNs are powerful, their explainability is restricted and not as transparent as simpler ML models. Furthermore, ANNs require significant amounts of data to learn from, which will be problematic given the limited data. Both categories of approaches will be trialled and compared. Before attempting synthetic sequence generation (refer to Section 3.3.1), LSTM should be applied and evaluated to gain a preliminary conclusion of the likely effectiveness of applying LSTM-GAN. Generating whole new sequences with a misguided neural network can be counterproductive as it risks jeopardising the data integrity and authenticity — which can mislead the data scientist and ML algorithms.

Another technique discussed in Section 2.2.1 was utilising DTW and k-Nearest Neighbour to group time series based on their DTW distances between each other. Given its simplicity and reliance on assuming the curves have significant differences between, it can be quite volatile and reliable. Nevertheless, this method can be trialled on its own, and potentially integrated with other methods through an ensemble method, such as model stacking or collective model voting.

### 3.6.1 Dynamic Sigma Time Thresholding

One of the methods trialled by Linear Diagnostics was applying 10-sigma time thresholding to classify time series as positive or negative. This was leveraged in Carter et al. (2022) for Dataset 3.2. It is a rule-based method that uses two thresholds, the "sig-

nificance threshold" and "classification threshold", to classify samples. The method is described as follows.

1. For a given (ordered) time series, take the first ten data points and subsequently calculate their mean and standard deviation.

2. Calculate the "significance threshold" for that specific time series by using Equation 1. The multiplier being set to ten ensures there is little chance that any time series that reach the threshold are outliers. This is because the multiplier decreases the probability that a time series reaches the threshold by chance, similarly to the three-sigma rule (Nascimento et al., 2021, pp. 8326-8327).

3. Set a "classification threshold" to a certain number of minutes.

4. If the time series exceeds the "significance threshold" before reaching the "classification threshold", it is considered positive, and negative otherwise. This is summarised in Equation 2. Additionally, this is visually demonstrated in Figure 3.5 using some fake data.

$$\theta_s = \mu + 10\sigma \tag{1}$$

Where:

- $\theta_s$: significance threshold
- $\mu$: mean of the time series
- $\sigma$: standard deviation of the time series

$$\text{Class}(x) = \begin{cases} \text{Positive}, & \text{if } \exists\ t_s < t_c \text{ such that } x(t_s) > \theta_s \\ \text{Negative}, & \text{otherwise} \end{cases} \tag{2}$$

Where:

- $x(t)$: time series signal at time $t$

- $\theta_s$: significance threshold
- $t_s$: time when signal first exceeds $\theta_s$
- $t_c$: classification threshold (time cutoff)

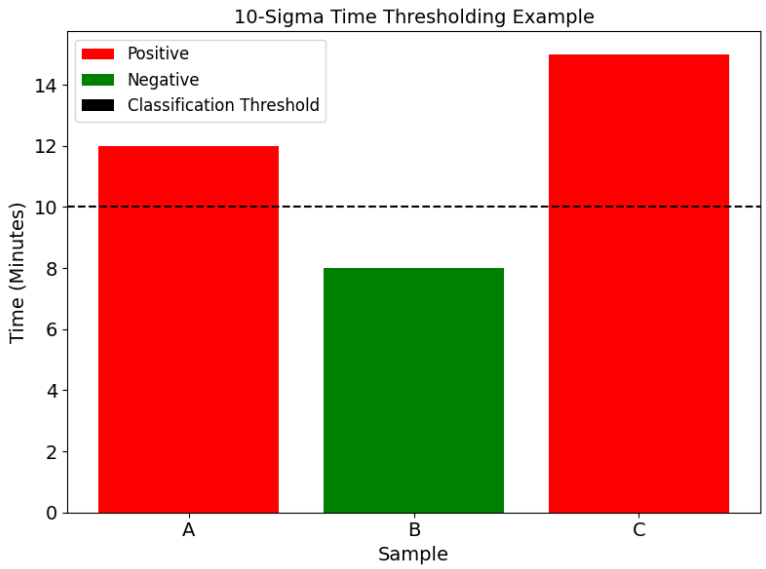


Figure 3.5: An example of the applying the classification step of the ten-sigma thresholding on some fake data.

Linear Diagnostics attempted to apply this same technique on Datasets 3.2 and 3.2, but unfortunately, their results became invalidated due to train-test data leakage. This is because they used the whole dataset to optimise the "classification threshold", instead of splitting the data into train-validation-test sets. Moreover, they did not perform class balancing, which would bias the results to potentially favour the majority class. In addition, besides optimising the "classification threshold", there are two other variables which can be optimised: the number of base points to use in Step 1 and the multiplier (10) in Equation 1. This therefore becomes a multi-objective optimisation challenge. The "classification threshold" can be considered a discrete variable as the indices in Datasets 3.2, 3.2, and 3.2 progresses in set increments. For example, for Dataset 3.2, each transition between adjacent data points for each time series represents 30 seconds (0.5 minutes). Hence, setting the "classification threshold" as a continuous variable

would not make a difference. For instance, setting the threshold as 6.25 minutes would practically be the same as setting it as 6 minutes. Moving on, the number of base points must also be considered a discrete variable as the index is discrete. Assuming the multiplier is maintained as a discrete value (such as 8, 9, 10 and so on), a simple brute force algorithm — with a time complexity of $n^3$ as there are three objectives to optimise — would find the optimum combination. The time complexity would not matter in this case as the set of values for each objective would be small. On the other hand, keeping the multiplier as a discrete value limits the solution space. Alternatively, if the multiplier is treated as a continuous variable it infinitely expands the solution space. Unfortunately, searching the entirety of such a space would not be feasible. This can be resolved however using evolutionary optimisation (Emmerich et al., 2023), as it enables directed searching of the space towards an optimum set of solutions. Table 3.1 summarises these objectives, including the range of possible values they can be set to.

Table 3.1: Evolutionary objectives.

| **Objective** | **Range** | **Variable type** |
|---|---|---|
| Number of base points | 3-15 | Discrete |
| Sigma multiplier | 5.0-15.0 | Continuous |
| Classification threshold | 3-30 | Discrete |

For multi-objective optimisation, based on surveys by Verma, Pant and Snasel (2021) and Q. Xu, Z. Xu and Ma (2020), three effective multi-objective algorithms include Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002), Strength Pareto Evolutionary Algorithm II (SPEA-II) (Zitzler, Laumanns and Thiele, 2001), and Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) (Zhang and Li, 2007). Although, NSGA-II's computational efficiency mixed with its diversity continuation due to its Pareto Front and crowding distances strategies, makes it typically a more ideal candidate compared to SPEA-II and MOEA/D. Therefore, this will be used to optimise the three aforementioned objectives. To evaluate each solution, an objective

function that measures sensitivity, specificity, and the area under curve-receiver-operator characteristic curve (AUC-ROC) will be used. Sensitivity will be used as it is better to capture the true positive rate for disease diagnosis (rather than not). Specificity will help maintain the precision of detected positives, preventing a scenario where a false strong candidate, that has tried to put all samples into the positive class, is given a high evaluation. Finally, AUC-ROC will help balance the trade-off between sensitivity and specificity — helping mitigate instances where the sensitivity is 0.9, but the specificity is 0.1, which is a false strong candidate like previously mentioned.

Other evolutionary parameters include the selection operator, the mutation rate and type, crossover, population size, and the number of generations. Binary Tournament Selection (a non-parametric approach) will be used as it presents little bias into the process, helps maintain fair diversity, and balances exploration versus exploitation by introducing randomness (of selection picks). The mutation rate will be subject to linear decay, which will help encourage lots of exploration in the earlier generations, and more exploitation in the later generations. This helps prevent premature convergence. The discrete objectives (the number of base points and "classification threshold") will be subject to a uniform mutation, while the continuous objective (the multiplier) will be subject to a Gaussian mutation. Similarly, the discrete objectives will be subject to a uniform crossover, while the continuous objective is subject to a simulated binary crossover (which is principally the same as the uniform crossover, but for continuous spaces). Uniform practices will be used in this case because they offer little bias into the process and promotes diversity. Table 3.2 summarises the range (if applicable) for each evolutionary parameter discussed. In addition, this includes the population size and the number of generations.

Table 3.2: Evolutionary parameters.

| **Evolutionary parameter** | **Setting/range** |
|---|---|
| Mutation rate | 0.4-0.1 |
| Categorical crossover rate | 0.5 |
| Continuous crossover simulated binary | 20 |
| Population size | 200 |
| Number of generations | 100 |

Ultimately, the original ten-sigma time thresholding method has been transformed into dynamic sigma-time thresholding (DSTT), which will attempt to optimise its parameters to the training data. This can be considered a custom ML algorithm as it requires no explicit input from the programmer.

## 3.7 Tools Summary

Table 3.3 summarises the proposed tools that will be needed to complete this implementation.

Table 3.3: Summary of tools and packages required for the implementation.

| **Tool** | **Usage** |
|---|---|
| Conda (Anaconda) | For creating and managing isolated package environments. |
| DEAP | A framework for utilising evolutionary algorithms. |
| Docker | A containerisation platform that will be used for creating container instances of a ClickHouse server, MariaDB server, and a Redis store. |
| Docker Compose | For constructing an all-in-one file that can manage Docker container collection. |
| Imbalanced Learn | This helps handle imbalanced datasets. |
| Jupyter Notebook | A platform for editing IPython notebooks. |
| Matplotlib and Seaborn | Python data visualisation packages that will be used for visualisations. |
| MLflow | A platform specialised for logging and tracking developed models. This will be used to help monitor any models developed, particularly for things such as model drift and data drift. |
| Numpy | A powerful library for numerical computing. |
| PyTorch | A Deep Learning framework that will be used to construct an LSTM neural network and model. |
| PyTorch Lightning | A Deep Learning framework which serves as a sort of add-on to PyTorch, which can make code development and code organisation more straightforward and standardised. |
| Pandas | A data manipulation and analysis library that will be used during the Ingestion stage to help ingest the data. |
| SciPy | A scientific library that contains a range of numerical analysis tools. |

## 3.8 Model Evaluation

All developed models should have hyperparameter tuning applied so that their predictive power is maximised. This helps greatly reduce the likelihood that hyperparameters are severely afflicting model progression. This is because, by default, ML algorithms have default hyperparameters set, which are not necessarily the best either independently or combined. Moreover, cross-validation should be used during the process to help enhance the generalisation strength the model has towards the training data. Cross-validation is a statistical technique used to help get an idea of model performance on some "unseen" data, without having to input more data, by splitting the training set into sets of mini-training folds and mini-testing folds. However, for smaller datasets, this method should be used cautiously as the testing fold results can become volatile as there are fewer samples to test against. Although, this is mitigated by averaging the testing fold results.

For evaluating the model performance, the classification metrics that will be used include:

- accuracy,
- precision,
- recall of positive class (sensitivity),
- recall of negative class (specificity),
- F1-score,
- area under curve-receiver-operator characteristic curve (AUC-ROC),
- and log loss (for neural networks and XGBoost).

Additionally, for neural networks, learning curves will also be plotted and visualised to help determine if the model has overfitted or underfitted. For both the model creation and prediction pipelines, they will be time-evaluated to determine an estimate of the minimum and maximum time it takes them to process.

### 3.8.1 Explainability and Interpretability

Some model-agnostic methods that can be used for estimating global feature importance (across the training/testing set) include partial dependency plots, permutation importance, and SHAP. Moreover, SHAP and LIME can be used for local feature importance (across a subset of samples from the training/testing set). Certain ML algorithms, such as decision trees and random forest include an in-built feature importance function, which determines global feature importance based on how much each feature contributes to reducing impurity within the branches — measured using entropy or Gini index. For estimating aleatoric and epistemic uncertainty in successful models, for each method the predictive posterior will be generated so that the uncertainty can be estimated. This can be done for neural networks by integrating Bayesian methods into the architecture. Another approach for estimating epistemic uncertainty would be looking at variance ratio of predictions versus the actual values.

# 4 Implementation and Results

All the following experiments were run on a laptop with an Intel Core i5-1035G1x8 processor, along with 16 GiB of Random Access Memory. Unlike the data splitting process in Section 3.4, outlier removal will be performed before the data splitting stage. This is because when the data is split, it is more difficult to identify specific outliers (as there is less data to compare overall). The smaller the dataset, the more impact outliers can have. This is especially true when evaluating how well a model has generalised, as the results from the validation and/or testing sets have outliers present — which can falsely degrade model performance. Additionally, these steps likely would not make a tremendous difference to data integrity as the number of outliers in Section 3.4 was relatively low. Furthermore, no other step changes are going to be made using the newly processed data (helping mitigate train-test data leakage).

## 4.1 Application of Dynamic Sigma-time Thresholding

As discussed in Section 3.6.1, DSTT was to be applied on the training set data for both the COVID-19 and chlamydia datasets. The train-validation-test sets remained consistent across all stages of this and subsequent processes, to help prevent train-test data leakage. Figures 4.1 and 4.2 illustrate the NSGA-II Pareto Fronts for the COVID-19 and chlamydia datasets respectively. The Pareto Front represents the set of best candidates at the end of the evolutionary process. Each blue dot represents a possible solution. The red highlighted point is considered the best solution within the respective set. This was calculated by,

1. normalising the objectives (evaluation metrics) for each solution,
2. then, setting the ideal solution point (which is 1.0 for all objectives when normalised),
3. and subsequently, calculating the Euclidean distance (in a higher dimensional space) between the ideal solution and each point,
4. and finally, the point with the smallest distance to the ideal solution is assumed

to be the best solution. This typically has a good trade-off between objectives.

It is clear that the DSTT method has been highly effective for the COVID-19 dataset, given the arched (convex) curve in Figure 4.1. A number of traits can be deduced based on the curve shape. For example, there is a fairly even spread of solutions, suggesting the healthy diversity within the final population. Although, there appears to be a more concentrated cluster of solutions at the centre of curvature — suggesting these solutions converged to offer more well-balanced trade-offs between the objectives. This is particular evident given the sensitivity and specificity values are much more balanced for this cluster, compared to another solution with a high sensitivity but extremely low specificity. In stark contrast to Figure 4.1, Figure 4.2 provides a mostly linear line with a negative correlation. This implies the DSTT method was not particularly effective for the chlamydia data, as there was a persistent trade-off between the objectives, most apparently the sensitivity and specificity. As there is no convex curve (or anything similar to a "knee point"), this suggests there is no particularly promising cluster of solutions. This is further highlighted by how dispersed and distributed the solutions are across the line. In summary, the DSTT method was very effective for the COVID-19 data, but not for the chlamydia data. Now the best solution has been identified, it can be used to further enrich analysis and ML development, most likely through model stacking. This can be achieved by passing the predictions from an ML model as a feature to another ML algorithm.

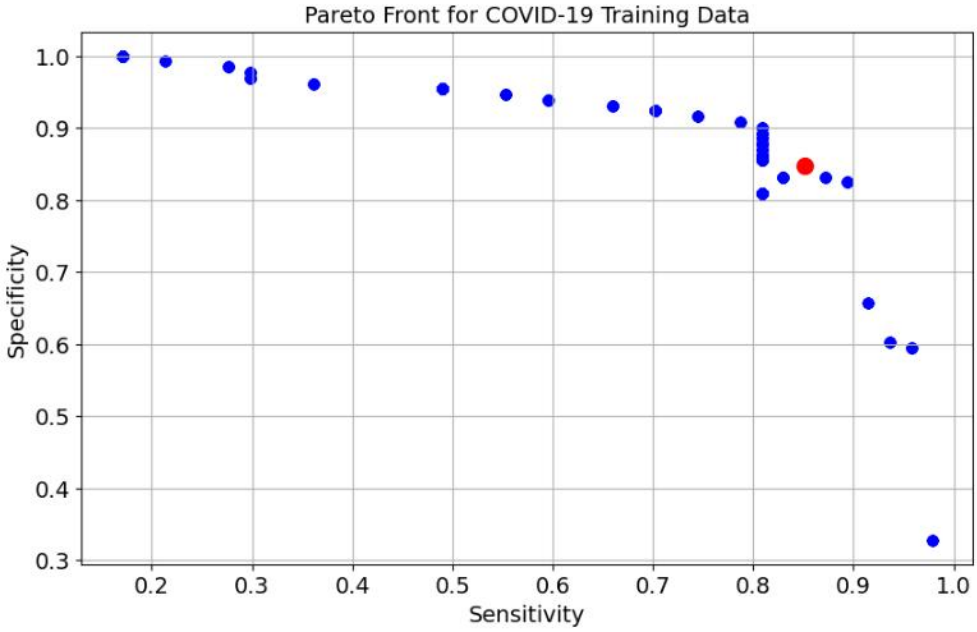


Figure 4.1: Pareto Front from applying DSTT on the normalised COVID-19 training set.

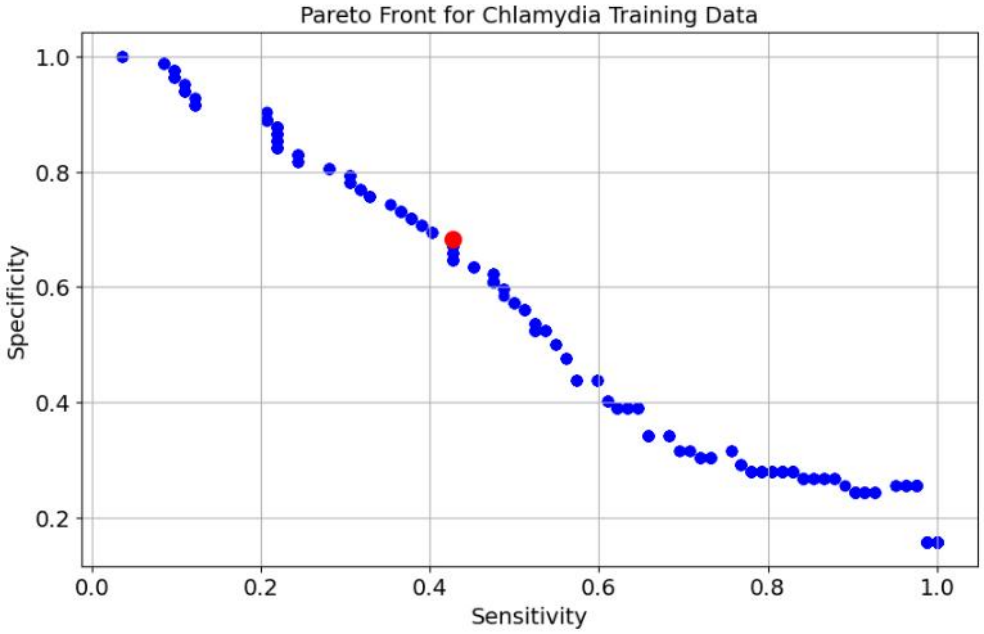


Figure 4.2: Pareto Front from applying DSTT on the normalised chlamydia training set.

## 4.2 Dynamic Time Warping Classification

As stated in Section 3.6, DTW and k-nearest neighbour classification works by calculating the DTW distances for every time series compared to all other time series. Therefore, each time series will have a group of distances, with the group size being determined by $n(n-1)$ where $n$ represents the number of time series. It is $n(n-1)$ and not $n^2$ because a time series is not compared against itself. Once these groups have been created, k-nearest neighbour can then start to classify them. Although k-nearest neighbour is difficult to interpret (due to the multitude of calculations done), it is highly reproducible as long as the dataset and 'k' is the same. Initially, 'k' (which represents how many neighbours to consult) was set to 3 for the k-nearest neighbour classifier for initial experimentation. However, this is not ideal as it is not guaranteed to be the best value to set. As demonstrated by Irawan, Fahmi and Zamzami (2024), the Elbow method can be employed to optimise the 'k' value in the ML algorithm. This was done using five-fold cross-validation as well, to help make the results more generalised and representative. Figures E.3 and E.4 illustrates the Elbow curves for the COVID-19 and chlamydia datasets respectively. Furthermore, Figures E.1 and E.2 show the classification results from applying the optimised 'k' value on the training sets; which happens to be '3' in both cases. As k-nearest neighbour essentially only calculates distances using a given metric and does not generate a model, these results on the training data are very promising for applying this method alone. Figures E.5 and E.6 show the validation results for the COVID-19 and chlamydia datasets respectively. Interestingly, the validation results for Figure E.5 exceed the training data (reaching 100% accuracy), while the validation results in Figure E.6 are significantly less than the training results (lowering to 68% accuracy). This will be discussed further in Section 5. Similarly to Section 4.1, the predictions by this method can be passed onto another model through model stacking.

## 4.3 Feature Analysis

Both the features from the generalised statistics and ARIMA methods discussed in Section 3.6 were combined, along with predictions from Sections 4.1 and 4.2 for easier attribute analysis. Table E.1 specifies all the ML input features (derived from both

datasets) along with their descriptions. Numerous statistics were gathered to provide a comprehensive set of attributes, this is to help enrich the analysis further and provide more pathways for ML algorithms to learn from. Before analysing the correlation of the features, the type of correlation analysis technique needs to be decided. As stated in Section 3.5.1, the Shapiro-Wilk test will be applied to all the features to determine if any of them are normally distributed. Figures 4.3 and 4.4 show the results of applying this test on the COVID-19 and chlamydia training sets respectively. Predictions by DSTT and DTW classification were excluded from this (as binary variables cannot directly be normally distributed). As shown in both figures, none of the features are normally distributed. Therefore, Kendall Tau correlation analysis would be more appropriate instead of Pearson correlation. Figures E.7 and E.8 together show the correlation heatmap for the COVID-19 dataset. In addition, Figures E.11 and E.12 simplify these figures as they only show the correlation in relation to 'result'. Figures E.9 and E.10 together show the correlation heatmap for the chlamydia dataset. Figures E.13 and E.14 simplify these figures as they only show the correlation in relation to 'result'. Some features have a negative correlation, which does not automatically mean the feature would be detrimental to utilise. Instead, this means the feature progresses with the curve outcome, but in an altered correlation direction. Any features with a correlation close to zero likely do not relate to the curve outcome. Despite containing similar features, the attributes in the COVID-19 dataset have a significantly higher correlation in relation to 'result' compared to the chlamydia dataset. This suggests the features in the former have a much more apparent relation individually to 'result' compared to the latter, as Kendall Tau measures monotonicity. Furthermore, this also emphasises the trends in the COVID-19 curves are more clear compared to the chlamydia curves. However, a low correlation does not necessarily mean the features would be ineffective for model predictions, as ML algorithms can uncover hidden or higher-dimensional trends that are not shown in correlation analysis.

| | Feature | P-value (4 d.p.) | Normally Distributed |
|---|---|---|---|
| 0 | mean | 0.0000 | False |
| 1 | std | 0.0000 | False |
| 2 | median | 0.0000 | False |
| 3 | iqr | 0.0000 | False |
| 4 | skew | 0.0470 | False |
| 5 | kurtosis | 0.0000 | False |
| 6 | spectral_entropy | 0.0001 | False |
| 7 | max_jump | 0.0000 | False |
| 8 | max_index | 0.0000 | False |
| 9 | auc | 0.0000 | False |
| 10 | early_mean | 0.0000 | False |
| 11 | after_mean | 0.0000 | False |
| 12 | autocorr_lag1 | 0.0000 | False |
| 13 | mean_abs_change | 0.0000 | False |
| 14 | mean_change | 0.0000 | False |
| 15 | inflection_index | 0.0000 | False |
| 16 | max_growth | 0.0000 | False |
| 17 | time_to_half | 0.0000 | False |
| 18 | ar_coef | 0.0000 | False |
| 19 | ma_coef | 0.0000 | False |
| 20 | resid_mean | 0.0000 | False |
| 21 | resid_std | 0.0000 | False |
| 22 | aic | 0.0000 | False |
| 23 | hqic | 0.0000 | False |

Figure 4.3: Shapiro-Wilk test results for COVID-19 ML input features.

| | Feature | P-value (4 d.p.) | Normally Distributed |
|---|---|---|---|
| 0 | mean | 0.0000 | False |
| 1 | std | 0.0000 | False |
| 2 | median | 0.0000 | False |
| 3 | iqr | 0.0000 | False |
| 4 | skew | 0.0001 | False |
| 5 | kurtosis | 0.0000 | False |
| 6 | spectral_entropy | 0.0130 | False |
| 7 | max_jump | 0.0001 | False |
| 8 | max_index | 0.0000 | False |
| 9 | auc | 0.0000 | False |
| 10 | early_mean | 0.0000 | False |
| 11 | after_mean | 0.0000 | False |
| 12 | autocorr_lag1 | 0.0000 | False |
| 13 | mean_abs_change | 0.0000 | False |
| 14 | mean_change | 0.0000 | False |
| 15 | inflection_index | 0.0000 | False |
| 16 | max_growth | 0.0001 | False |
| 17 | time_to_half | 0.0000 | False |
| 18 | gender_F | 0.0000 | False |
| 19 | gender_M | 0.0000 | False |
| 20 | ar_coef | 0.0000 | False |
| 21 | ma_coef | 0.0000 | False |
| 22 | resid_mean | 0.0000 | False |
| 23 | resid_std | 0.0000 | False |
| 24 | aic | 0.0000 | False |
| 25 | hqic | 0.0000 | False |

Figure 4.4: Shaprio-Wilk test results for chlamydia ML input features.

It is worth noting that the time-series index "cut-off" point for the 'early_mean' and 'after_mean' is variable as it could be set for ten minutes, twenty minutes, or something else entirely. To optimise the "cut-off", the golden search optimisation algorithm was selected. Noroozi et al. (2022) leveraged this approach to minimise the number of input features for ML algorithms. Instead of doing an exhaustive approach which can be

more computationally expensive, this algorithm leverages the golden ratio to reduce the number of evaluations while keeping a stable convergence trend. Given the small ranges to trial (1-99 for the COVID-19 data, and 1-44 for the chlamydia data), this could have been brute forced. However, if in future work the dataset size increases significantly, this can become inefficient. In addition, utilising EAs could be applied here, but given the simplicity of the problem, it could overcomplicate the process unnecessarily. Finally, the golden search algorithm is better at search compared to other non-exhaustive approaches such as random search. For this approach, the tolerance level was set to '2'. This means that if the next solution is only two levels over, the algorithm will stop (as it has reached the ideal solution region), which also helps prevent wasting time on minor improvements. Moreover, the maximum number of iterations was set to '10'. As well as this, the starting point was always set to index 1, and the maximum point was always set to the index size of the dataset minus one ($max = n - 1$ where $n$ equals the dataset size). For the COVID-19 training data, the optimum "cut-off" was found to be 42 minutes, while for the chlamydia training data, it was found to be 3.5 minutes.

In addition to correlation analysis, Mutual Information Gain (MIG) will also be used be. As illustrated by Alduailij et al. (2022, pp. 1101-1105), MIG is useful for capturing more complex dependencies as it measures how knowing one variable lowers uncertainty in another (in a non-linear fashion). Moreover, it is non-parametric, making it fitting for these datasets (as shown previously). Figures 4.5 and 4.6 show the MIG results for the COVID-19 training set and the chlamydia training set respectively. Some features have both a low correlation (closer to zero) and low MIG (closer to zero) in relation to the target, suggesting these features would not be effective for predicting the outcome. These will be removed alongside features which have a zero MIG. This is done even if they exhibit a correlation with the outcome somewhat because correlation could be present due to coincidence. The features to be removed for the COVID-19 scenario based on their correlation and MIG include 'std', 'max_index', 'kurtosis', 'resid_mean', 'ar_coef', 'ma_coef', 'iqr', 'resid_std', 'aic', 'hqic', 'mean_change', 'mean_abs_change', and 'max_jump'. The features to be removed for the chlamydia scenario based on their correlation and MIG include 'meets_thresold', 'autocorr_lag1', 'gender_F', 'spec-

tral_entropy', 'max_index', 'aic', 'hqic', and 'knn_prob_0'.

```
knn_prob_1          0.531992
knn_prob_0          0.498074
inflection_index    0.329109
meets_threshold     0.262622
median              0.239790
after_mean          0.231724
mean                0.220468
auc                 0.217864
skew                0.151079
autocorr_lag1       0.150161
spectral_entropy    0.147394
time_to_half        0.146848
mean_abs_change     0.098205
max_growth          0.096414
early_mean          0.087172
max_jump            0.086968
aic                 0.079895
hqic                0.079895
resid_std           0.070747
std                 0.002209
max_index           0.000000
iqr                 0.000000
kurtosis            0.000000
mean_change         0.000000
resid_mean          0.000000
ar_coef             0.000000
ma_coef             0.000000
dtype: float64
```

Figure 4.5: Mutual Information Gain feature results for the COVID-19 training set.

```
knn_prob_1          0.237953
knn_prob_0          0.224394
auc                 0.162344
iqr                 0.153391
time_to_half        0.128630
mean_abs_change     0.125796
resid_std           0.113892
std                 0.108402
mean                0.095835
kurtosis            0.089308
after_mean          0.083422
mean_change         0.079524
inflection_index    0.077878
ma_coef             0.076825
max_growth          0.062359
max_jump            0.062359
max_index           0.058285
skew                0.047124
resid_mean          0.039845
ar_coef             0.035279
meets_threshold     0.026191
gender_M            0.019402
early_mean          0.017075
median              0.010623
hqic                0.008263
aic                 0.008263
spectral_entropy    0.000000
autocorr_lag1       0.000000
gender_F            0.000000
dtype: float64
```

Figure 4.6: Mutual Information Gain feature results for the chlamydia training set.

## 4.4 Model Development

As discussed in Section 3.8, all models will be hyperparameter tuned so that the results are the best they can be, and so they are more consistent (and not subject to any default settings). Tables E.2 to E.8 specify the hyperparameters to be tuned for decision tree, random forest, XGBoost, k-nearest neighbour, ridge regression, logistic regression, and support vector classifier. To reiterate, all random states across the ML algorithms and other randomness-involved algorithms are set to '42'. This helps improve reproducibility of results. As mentioned in Section 3.6, another type of model ensemble technique (besides model stacking) is model voting, where a number of different types of models collectively vote on a prediction. This can be done through two forms: the 'soft' approach

or the 'hard' approach. The 'soft' approach is where the predicted class probability distributions for each model are compared and combined, and the class with the highest mean average probability is selected as the predicted class. The 'hard' approach is where each model predicts the class for the given sample, then these are tallied together and the class with the most votes is assumed to be the predicted class. Both approaches have their own advantages. For example, the 'soft' approach ensures that weak model predictions are mitigated, so if the class probability is just above 50%, this will not impact the overall voting performance as significantly as the 'hard' approach — where that model would be given a whole vote regardless of whether the class probability is 90% or 60% and so on. Unfortunately, not all models, such as ridge regression and support vector machines, natively support providing probability distributions (as they are not probabilistic methods). Therefore, these might not be usable with the 'soft' approach, but they would be compatible with the 'hard' approach, as this only requires the predicted class (not its probability). The voting ensemble was used with a 'hard' approach and was made up of the hyperparameter tuned models, which included decision tree, k-nearest neighbour, logistic regression, and support vector classifier. These were chosen as they take significantly different approaches in their predictions. It would be unwise to use ensemble methods such as random forest with this ensemble approach as it degrades the explainability of the process by increasing its dependencies and complexity.

To further optimise the set of ideal features to use for each dataset, initially the dataset features from Section 4.3 (that were not removed) will be used. Subsequently, each model (except the voting ensemble) will be trained and hyperparameter tuned on the training data. Then, the accuracy, cross-validation accuracy, and the validation accuracy will be used to determine which model(s) performed the best. From here, permutation importance analysis will be used to remove features that offer little predictive power. This process is then recursively done until a certain number of features remain that offer the highest performance. It is important to note, for calculating the cross-validation score, stratified five-fold cross validation is done. This is to help reproducibility as the proportion of samples per fold will be consistent. Figures 4.7 and 4.8 show the initial results for the initial features passed to the models for the COVID-19 and chlamydia

datasets respectively.

| | Model | Training Accuracy | Validation Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|
| 4 | ridge | 95.8 | 100.0 | 93.8 |
| 5 | logistic | 94.8 | 100.0 | 91.7 |
| 1 | random_forest | 95.8 | 92.9 | 93.7 |
| 3 | knn | 100.0 | 92.9 | 86.5 |
| 2 | xgboost | 93.8 | 92.9 | 93.7 |
| 6 | svc | 94.8 | 92.9 | 92.7 |
| 0 | decision_tree | 94.8 | 85.7 | 91.7 |

Figure 4.7: Initial model results from the initial features passed for the COVID-19 dataset.

| | Model | Training Accuracy | Validation Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|
| 3 | knn | 100.0 | 80.0 | 78.0 |
| 1 | random_forest | 92.9 | 68.0 | 86.3 |
| 0 | decision_tree | 97.6 | 68.0 | 87.0 |
| 2 | xgboost | 94.0 | 68.0 | 86.9 |
| 4 | ridge | 86.3 | 68.0 | 85.7 |
| 6 | svc | 85.7 | 68.0 | 84.5 |
| 5 | logistic | 86.3 | 64.0 | 84.5 |

Figure 4.8: Initial model results from the initial features passed for the chlamydia dataset.

Remarkably, even without reducing the number of features, all models, particularly ridge regression and random forest have above 90% accuracy. As shown in Figures E.15 and E.16, which represent the permutation importances for the ridge and logistic regression models for the COVID-19 dataset, both models are picking up on the high correlation between the DSTT thresholding predictions, the DTW and k-nearest neighbour predictions, and the final outcome, alongside a few other attributes (likely refining the prediction result). Although the results are extremely promising, recursive feature

selection will still be completed using random forest to determine if similar results can be achieved without relying on the DSTT thresholding and DTW and k-nearest neighbours methods. Random forest was chosen as it is less susceptible to individual feature bias, unlike decision tree. This will also be done for the chlamydia data using random forest, as these models appear to show the most initial promise. Even though the results from the ridge and logistic models are excellent, the foundation that these predictions were made is less robust due to having to rely on predictions from other ML models. In other words, the ridge model for instance is making a prediction based on two other predictions (and some real data). It also makes the ridge and logistic models dependent on the DSTT thresholding and the DTW and k-nearest neighbour performing well, rather than just relying on the time series itself.

### 4.4.1 LSTM

Unlike the other ML approaches, LSTM can process the time series directly without manual feature engineering. However, certain preprocessing steps such as data scaling, label encoding, sequence shortening (exclusively for the chlamydia data), outlier removal, and data splitting will remain the same. This is to help improve result comparability between LSTM and the other algorithms. On the other hand, some additional steps are required to make the data compatible with LSTM. These include converting the independent variables (feature sets) into tensors. When converting the data into a tensor, the original long-formatted shape of the time-series data is transformed. For example, the preprocessed shape of the feature training set for the COVID-19 dataset was '[96, 1]', meaning there are ninety-six entries present, where each entry is a time series. After transforming this into a tensor, the new shape becomes '[96, 1, 100]', meaning there are ninety-six entries, only 1 feature inputted (the time series), and each time series consists of one hundred data points. Unfortunately, LSTM requires specific formats such as '[96, 100, 1]'. However, this can be accomplished by permuting the tensor dimensions.

To improve reproducibility, randomisation seeds for PyTorch were set to 42, and the backend was configured to be deterministic. Unfortunately, this may limit the model's ability to learn as it cannot explore further reaches of the solution space. A single-layer

LSTM layer was used with an allocated '64' neural units, which was connected to a prediction layer with a single neural unit (as it is binary classification). Additionally, to help encourage the model to generalise better and mitigate overfitting, a dropout was included at the end of the epoch with a threshold of '0.3' (30%). Other parameters including the learning rate which was set to '0.001', and weight decay was also included to help regularise the training (by helping penalise larger weights), which was set to '0.001'. Moreover, binary cross-entropy was used as the loss function, and the AdamW method was used as the optimiser. AdamW operates the same as the normal Adam optimiser does except it applies regularisation parameters such as weight decay separately from gradient updates. The process was run for 100 epochs with gradient clipping (to help prevent exploding gradients), but an early stopping mechanism (with a patient of 15 epochs) was also included to prevent unnecessary additional training and mitigate overfitting to the data. It was often found the process would only run for an estimated 20-30 epochs for the chlamydia data with a batch size of '10', and 50-60 epochs for the COVID-19 data with the same batch size.

Tables E.9 and E.11 illustrate the test results for LSTM on the COVID-19 and chlamydia datasets respectively. Similarly, Figures E.26 and E.37 show the learning curves for each dataset in the same order. It is clear that the LSTM performed better on the COVID-19 dataset compared to the chlamydia dataset in terms of higher accuracy, lower log loss, and higher AUC-ROC. The COVID-19 learning curve, although still slightly erratic, it fundamentally heads in the direction to minimise log loss and improve validation accuracy. The chlamydia learning curve on the other hand, is entirely erratic throughout the process, suggesting the LSTM could not pick up on defining points that characterise the curves. To account for underfitting, each experiment was repeated with 128 neural units for the LSTM instead of 64. Tables E.10 and E.12 show the test results for LSTM (with a hidden size of 128) on the COVID-19 and chlamydia datasets respectively. Furthermore, Figures E.27 and E.38 present the learning curves for each dataset in the same order. For the COVID-19 dataset, all metrics have improved, particularly accuracy (from 78% to 82%) and log loss (from 0.501 to 0.411). While the accuracy has also improved for the chlamydia dataset (from 69% to 71%), the overall

model performance has degraded as the log loss worsened (from 0.546 to 0.582) and the sensitivity-specificity trade-off has become more imbalanced, with sensitivity at 100% (from 83.3%) and specificity at 40% (from 51.7%). Additionally, the training and validation learning curves for the COVID-19 dataset appear to converged better, but for the chlamydia dataset, the curves are still extraordinarily erratic.

### 4.4.2 Final Results

For the COVID-19 data, the best feature combination discovered is as follows:

- mean,
- median,
- standard deviation,
- interquartile range,
- skew,
- area under curve,
- early mean,
- after mean,
- time to half,
- and the DSTT thresholding predictions.

Figures E.17 to E.24 illustrate the overall performance evaluation for each method for the COVID-19 dataset. Figure 4.9 illustrates the overall accuracy results for all models for the COVID-19 dataset. In addition, Figure E.25 shows the performance evaluation for the DTW and k-nearest neighbour method on the COVID-19 dataset. It is important to clarify that normalisation was used for data scaling for the COVID-19 dataset.

| | Model | Training Accuracy | Validation Accuracy | Testing Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|---|
| 0 | decision_tree | 92.7 | 85.7 | 92.9 | 92.7 |
| 1 | random_forest | 95.8 | 85.7 | 92.9 | 92.7 |
| 2 | xgboost | 99.0 | 92.9 | 92.9 | 92.7 |
| 4 | ridge | 93.8 | 92.9 | 92.9 | 92.7 |
| 7 | voting | 94.8 | 92.9 | 92.9 | 92.7 |
| 6 | svc | 94.8 | 85.7 | 92.9 | 92.7 |
| 5 | logistic | 90.6 | 100.0 | 89.3 | 88.5 |
| 3 | knn | 100.0 | 92.9 | 82.1 | 85.4 |

Figure 4.9: Accuracy summary of results for all models for the COVID-19 dataset.

For the chlamydia data, the best feature combination discovered is as follows:

- mean,
- median,
- interquartile range,
- area under curve,
- early mean,
- after mean,
- time to half,
- female gender designation,
- and the residual mean.

Figures E.28 to E.35 illustrate the overall performance evaluation for each method for the chlamydia dataset. Figure 4.10 illustrates the overall accuracy results for all models for the chlamydia dataset. In addition, Figure E.36 shows the performance evaluation for the DTW and k-nearest neighbour method on the chlamydia dataset. To note, z-score standardisation was used on the original chlamydia time-series data, as it enabled slightly improved performance compared to normalisation.

| | Model | Training Accuracy | Validation Accuracy | Testing Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|---|
| 0 | decision_tree | 88.1 | 76.0 | 87.8 | 78.0 |
| 1 | random_forest | 100.0 | 84.0 | 85.7 | 83.8 |
| 3 | knn | 100.0 | 80.0 | 83.7 | 78.5 |
| 2 | xgboost | 100.0 | 72.0 | 75.5 | 82.7 |
| 6 | svc | 88.7 | 76.0 | 75.5 | 76.2 |
| 7 | voting | 81.0 | 76.0 | 73.5 | 72.6 |
| 5 | logistic | 66.7 | 68.0 | 65.3 | 65.5 |
| 4 | ridge | 66.7 | 68.0 | 65.3 | 65.5 |

Figure 4.10: Accuracy summary of results for all models for the chlamydia dataset.

## 4.5 Project Timeline

Figures G.1 and G.2 (in Appendix G) illustrates the originally proposed project timeline for the completion of this report. This has been refined as shown in Figures G.3 and G.4. The main difference is that the model development phase was expanded to encompass the MLOps portion — as this became too large to be done within the scope of this project. Most of the timeline was kept to, with the exception that the model development phase took longer to uncover better ML results for the chlamydia dataset.

# 5 Evaluation and Discussion

The promise of the proposed solutions in Section 4 is clear, where a number of the COVID-19 ML models exceed 90% accuracy and ROC-AUC, and the steady model for the chlamydia data that is near 90% accuracy. Interestingly, both the COVID-19 and chlamydia experiments offered similar yet vastly different issues. For example, as many of the critical features in the COVID-19 had a high correlation with the outcome, this made it difficult to encourage the model to generalise better without them. In contrast, the features in the chlamydia data did not obviously correlate or relate to the outcome, making it challenging to build up the accuracy. However, given the good results from the chlamydia experimentation in Section 4.4.2, this suggests the patterns within the curves of the chlamydia data are potentially hidden in a higher dimensional space. Despite differences in both data and approach, typically, the validation set would perform more poorly than the testing set in a number of instances across both experiment settings. This likely would be due to the small samples sizes (ranging from about 14-25 across both cases) incurred due to the limited amount of data overall.

Throughout the experiments, strict preprocessing was followed to ensure model results were representative, data integrity remained intact, and so there was little chance of data leakage being introduced. This included but was not limited to data splitting during the EDA, maintaining the same train-test splitting throughout, set randomisation seeding, avoiding trialling hypothesis and promising ML results until the final overall testing stage (using the testing set), verifying distribution normality, and more. Ultimately, this all helps encourage fair and reproducible results. In addition to this, as there was little to no literature covering these specific scenarios involving RTF-EXPAR and disease curves, different models (such as decision tree, random forest, and so on) were created and subsequently compared against each other to help benchmark the results.

## 5.1 Final Results

For the COVID-19 dataset, the top performing models with consistent train-validation-test results were XGBoost, ridge regression, and the voting ensemble classifier (see Figure 4.9). Out of these three, ridge regression can arguably be considered the most ideal as it

is a linear model, and is therefore entirely explainable and easily interpretable. This is in contrast to the other two methods which are ensemble-based, which can often generalise better but at the cost of interpretability — as it is more difficult to trace how each model came to that conclusion.

For the chlamydia dataset, the top performing models included decision tree, random forest, and k-nearest neighbour (see Figure 4.10). Random forest in particular performed the best overall, as its train-validation-test scores are fairly consistent and it was slightly higher than k-nearest neighbour. Although the decision tree model had a slightly higher testing accuracy, its lower validation accuracy suggests it did not generalise as well overall. The chlamydia data was typically used by ML algorithms better when the original time series was standardised instead of being normalised. Z-score standardisation slightly alters the original data by discouraging more extreme values (due to the data becoming normally distributed). This may have highlighted new core trends within the data, leading to an improved likelihood an ML model would pick up on this and generalise better.

As previously highlighted, the LSTM was likely not going to be very effective due to the limited data available. However, for the COVID-19 data the LSTM appeared to be generalising somewhat quite well as the learning curves became smoother. With a larger dataset, LSTM performance could significantly improve. For the chlamydia data on the other hand, not only did the LSTM perform notably worse, it clearly did not show signs of generalising well due to its erratic learning curves. This latter point in particular helps emphasise the challenges in finding and utilising key relationships within the chlamydia data.

## 5.2 Explainability and Uncertainty

The best models from the experiments will be explained and interpreted to help improve transparency of their predictions. For the COVID-19 data, ridge regression is selected given its high performance, consistent results, and linear nature. For the chlamydia data, random forest is chosen given its performance and consistent results. Furthermore, this combination will help illustrate the varying degrees of explainability that can be used

for explainability and uncertainty. Ridge regression is a singular, linear model while random forest is a non-linear ensemble model.

### 5.2.1 Ridge Classification

Figures E.39, E.40 and E.41 shows the coefficient analysis, permutation importance for the COVID-19 validation set, and the SHAP values for the ridge classifier for the COVID-19 data. Each black line for the permutation importance represents the standard deviation of that specific bar. The SHAP values were formed by testing the model against the validation set. LIME was not applied in this case because ridge is already a linear model — it does not require another surrogate linear model on top. As shown in Figure E.39, the DSTT predictions alongside the area under curve features have the strongest influence on model predictions, followed by the median. As demonstrated in Section 4.1, DSTT predictions were already very effective for the COVID-19 data (with sensitivity and specificity at almost 90%), but alongside some additional information, a linear trend is clearly present. This is partly confirmed by E.40, with the DSTT predictions and area under curve contributing the most predictive power. Although in contrast, it appears the median is actually detrimental to the predictive performance. Explainability disagreements can have a number of factors and causes. For example, coefficient analysis becomes less reliable when multicollinearity is present for certain features, which is clearly present when referring back to Figures E.7 and E.8. Moreover, the coefficients may not represent the true importance of each feature, or more specifically the combination of them. Coefficient analysis assumes each feature is independent, that is, it does not try to model the relationships between them. Permutation importance on the other hand does take into account the combination of features as it measures the overall predictive power when shuffling the feature values. However, permutation importance is not perfect either, particularly when features are correlated between each other. This is because if two or more features are correlated, theoretically the model performance will not drop as it still has the other feature. Therefore, it can lead to underestimations of feature importance for correlated features. The SHAP values in Figure E.41 appears to agree more so with the permutation importance, in that those

features have a bigger impact on prediction power. Nonetheless, the coefficient analysis is based off of the training data, while the permutation importance and SHAP are based on the validation data, which might highlight why there is a slight disagreement in feature importance.

### 5.2.2 Random Forest

Figures E.42, E.43 and E.44 show the feature importance (using Gini), permutation importance for the chlamydia validation set, and a LIME test case for the chlamydia data. Unfortunately, the SHAP values were not configuring correctly, so they were not included.

Figure E.45 shows some summarised ensemble variance values based on the variance between the decision trees that make up random forest, with the chlamydia testing set being used for the predictions. This is a measure of how much epistemic uncertainty there is (Hüllermeier and Waegeman, 2021, pp. 457-460). Epistemic uncertainty is a measure of the model's ignorance (not knowing or learning enough). The lower the uncertainty, the better. The more decision trees that agree on a prediction, the more confident the random forest model becomes. The mean ensemble variance is 0.165 (3 s.f.), which indicates some disagreements between the decision trees that make up random forest. Another measure of epistemic uncertainty is confidence scores, which is derived from the predictive posterior of the model. Figure E.46 shows the summarised confidence scores when random forest is applied to the chlamydia testing set. The mean predictive entropy is 0.495 (3 s.f.), which indicates there is a notable amount of uncertainty with the model's predictions. Moreover, based on the aforementioned figures, as the predictive entropy goes up so does the ensemble variance, suggesting those samples are more difficult to classify. Measuring aleatoric uncertainty is about how well the model can respond to added noise or fluctuations in the data (Hüllermeier and Waegeman, 2021, pp. 457-463). One way to do this is as follows.

1. Take a test case sample and add a certain amount of noise (randomness) into the data point.

2. Have the model make a prediction regarding the sample. It does not matter

whether it is right or wrong, only if it is the same prediction.

3. Repeat Steps 1-2 for a set number of times. For example, for 100, then at the end of the cycles, there should be 100 predictions for the given sample. However, it is important to note, the test case sample is returned to its original state, and a new amount of noise is added each time. This means the model has been tested on whether it gives the same response each time when a small amount of noise is added to the original data point (and returned to its original state when making a new prediction).

The aforementioned steps are repeated for each sample in the chlamydia testing set. If the model is robust to small amounts of perturbations or noise in the data, then there should be little variance present between the 100 predictions per sample. This would mean aleatoric uncertainty would be low. Otherwise, aleatoric uncertainty would increase the more times the model predicted a different answer to previous times. The mean aleatoric uncertainty when using the chlamydia testing set was measured at 0.0688 (3 s.f.). The maximum uncertainty was measured at 0.25 (3 s.f.), and 7 instances of the 49 test case samples were detected to have had uncertainty above 0.2. Overall, this low aleatoric uncertainty means the random forest model is fairly robust and less susceptible to minor alterations in the data.

## 5.3 Practical Implications

Despite excellent ML results, it would not be viable for Linear Diagnostics to move these models into a production environment due to issues such as model drift and/or data drift (Bayram, Ahmed and Kassler, 2022, pp. 108632-108641). In essence, these issues arise when trends in data change overtime, which makes ML models outdated, and hence suffer from degraded performance. These issues are particularly exacerbated when it comes to certain diseases, as they can frequently mutate and exhibit different behaviours. Fortunately, this can be mitigated and monitored using machine learning operations (MLOps), as it enables monitoring how both models and data change over time. Fundamentally, the excellent results from the models should not be discarded, especially as they form a proof of concept; that it is achievable to make up for the

shortcomings of RTF-EXPAR using ML. Moreover, this includes integration with cloud services (such as Azure, Google Cloud, and/or Amazon Web Services) so that prototypes in future can be updated dynamically (through the Internet). This would also enable easier data collection as any samples tested could be added to a pool of resource in a cloud service (pending ethical and legal arguments).

Additionally, more experiments should be conducted regarding RTF-EXPAR and a wider range of diseases. This should also include certain cases where a patient sample has both or multiple diseases, so that a new avenue of investigation can be unlocked. For example, this report could not produce a multi-class classifier (a single model fits all approach) so that both COVID-19 and chlamydia could be diagnosed at once because there was no data regarding how RTF-EXPAR is affected when a patient sample has both. This could be expanded to examine (and verify) the finer nuances of how RTF-EXPAR works.

# 6 Conclusions and Future Work

This report illustrated that machine learning has been highly effective for classifying the time-series curves produced by the RTF-EXPAR reaction. Additionally, the use of explainable and interpretable methods has improved transparency and reliability of how the best performing models made their decisions. This solidifies the importance of more traditional ML approaches, particularly on small datasets and/or when explainability is key. Hence, this report ultimately contributes to wider efforts to integrate ML into healthcare settings, with such environments requiring explainability as necessity. The final results gave over 93% accuracy for the COVID-19 dataset, and over 86% accuracy for the chlamydia dataset. It was clear that even though the curves looked visually similar, hidden patterns distinguished them apart. In other words, the widely different ease of application of ML was due to the type of disease.

Future work includes integrating MLOps with this report's work to help shift it from a more theoretical standpoint to a production one. Additionally, more experiments should be conducted regarding RTF-EXPAR and a wider range of diseases, including certain cases where a patient may have multiple ones.

## Reference List

AIO Photos (2025). *Agile Methodology Vs Waterfall Model*. Available at: https://www.aiophotoz.com/photos/agile-methodology-vs-waterfall-model.html [Accessed on 25th May 2025].

Alanazi, A. (2022). Using machine learning for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, 30 (1), p. 100924. DOI: 10.1016/j.imu.2022.100924.

Alduailij, M., Khan, Q. W., Tahir, M. et al. (2022). Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method. *Symmetry*, 14 (6), p. 1095. DOI: 10.3390/sym14061095.

Amazon Web Services (2025). *Amazon Web Services*. Available at: https://aws.amazon.com/ [Accessed on 22nd Aug. 2025].

Anderson, S. (2024). Expanding data literacy to include data preparation: building a sound marketing analytics foundation. *Journal of Marketing Analytics*, 12 (2), pp. 227–234. DOI: 10.1057/s41270-024-00293-3.

Bagheri, A., Patrignani, A., Ghanbarian, B. et al. (2025). A hybrid time series and physics-informed machine learning framework to predict soil water content. *Engineering Applications of Artificial Intelligence*, 144 (2025), p. 110105. DOI: 10.1016/j.engappai.2025.110105.

Bayram, F., Ahmed, B. S. and Kassler, A. (2022). From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 245 (2022), p. 108632. DOI: 10.1016/j.knosys.2022.108632.

Blasco, T., Sanchez, J. S. and Garcia, V. (2024). A survey on uncertainty quantification in deep learning for financial time series prediction. *Neurocomputing*, 576 (2024), p. 127339. DOI: 10.1016/j.neucom.2024.127339.

Carter, J. G., Iturbe, L. O., Duprey, J.-L. et al. (2022). Ultrarapid detection of SARS-CoV-2 RNA using a reverse transcription–free exponential amplification reaction, RTF-EXPAR. *Applied Physical Sciences*, 118 (35). DOI: 10.1073/pnas.2100347118.

Caruso, G., Giammanco, A., Virruso, R. et al. (2021). Current and Future Trends in the Laboratory Diagnosis of Sexually Transmitted Infections. *International Journal of Environmental Research and Public Health*, 18 (3), p. 1038. DOI: 10.3390/ijerph18031038.

Costa, B. and Georgieva, P. (2023). Explainable Artificial Intelligence in Healthcare Applications: A Systematic Review. In: *2023 International Scientific Conference on Computer Science (COMSCI)*. 18-20 September 2023. Sozopol, Bulgaria: IEEE Xplorer, pp. 1–8. DOI: 10.1109/COMSCI59259.2023.10315829.

Dasari, K. B. and Devarakonda, N. (2022). TCP/UDP-Based Exploitation DDoS Attacks Detection Using AI Classification Algorithms with Common Uncorrelated Feature

Subset Selected by Pearson, Spearman and Kendall Correlation Methods. *Revue d'Intelligence Artificielle*, 36 (1), pp. 61–71. DOI: 10.18280/ria.360107.

Deb, K., Pratap, A., Agarwal, S. et al. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, pp. 182–197. DOI: 10.1109/4235.996017.

Du, K.-L., Jiang, B., Lu, J. et al. (2024). Exploring Kernel Machines and Support Vector Machines: Principles, Techniques, and Future Directions. *Mathematics*, 12 (24), p. 3935. DOI: 10.3390/math12243935.

Efimov, V. (2024). *Demystifying Confidence Intervals with Examples*.

Emmerich, M., Deutz, A., Wang, H. et al., eds. (2023). *Evolutionary Multi-Criterion Optimization - 12th International Conference*. Leiden, Netherlands, 20-24 March 2023. New York, NY: Springer.

Fang, J., Li, R., Zhang, T. et al. (2025). Time series identification, classification, and display methods for periodic flow patterns. *Flow Measurement and Instrumentation*, 106 (2025), p. 102977. DOI: 10.1016/j.flowmeasinst.2025.102977.

Folgado, D., Barandas, M., Famiglini, L. et al. (2023). Explainability meets uncertainty quantification: Insights from feature-based model fusion on multimodal time series. *Information Fusion*, 100 (2023), p. 101955. DOI: 10.1016/j.inffus.2023.101955.

Gani, R., Isty, M. N., Rimi, R. A. et al. (2024). Impact of AI in Healthcare Services: Analysis Using Medical Synthetic Data. In: *Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*. 8-9 March 2024. Dhaka, Bangladesh: IEEE Xplorer, pp. 1–6. DOI: 10.1109/iCACCESS61735.2024.10499613.

Geyer, P., Singh, M. M. and Chen, X. (2024). Explainable AI for engineering design: A unified approach of systems engineering and component-based deep learning demonstrated by energy-efficient building design. *Advanced Engineering Informatics*, 62 (2024), p. 102843. DOI: 10.1016/j.aei.2024.102843.

Google (2025). *Google Cloud*. Available at: https://cloud.google.com/ [Accessed on 22nd Aug. 2025].

Hernandez, M., Epelde, G., Alberdi, A. et al. (2022). Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493 (2022), pp. 28–45. DOI: 10.1016/j.neucom.2022.04.053.

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110 (2021), pp. 457–506. DOI: 10.1007/s10994-021-05946-3.

Irawan, B., Fahmi, F. and Zamzami, E. M. (2024). Optimizing K-Nearest Neighbor Values Using The Elbow Method. In: *IEEE International Conference on Informatics and Computing (ICIC)*. 24-25 October 2024. Medan, Indonesia: IEEE Xplorer. DOI: 10.1109/ICIC64337.2024.10957541.

König, T., Ramananda, A. M., Wagner, F. et al. (2024). A LSTM-GAN Algorithm for Synthetic Data Generation of Time Series Data for Condition Monitoring. *Procedia Computer Science*, 246 (2024), pp. 1508–1517. DOI: 10.1016/j.procs.2024.09.602.

Lee, S., Choi, C., Do, H. et al. (2025). Batch active learning for time-series classification with multi-mode exploration. *Information Sciences*, 711 (2025), p. 122109. DOI: 10.1016/j.ins.2025.122109.

Linear Diagnostics (2025). *Next generation of near-patient diagnostics*. Available at: https://www.lineardiagnostics.com/ [Accessed on 3rd June 2025].

Martínez-Agüero, S., Soguero-Ruiz, C., Alonso-Moral, J. M. et al. (2022). Interpretable clinical time-series modeling with intelligent feature selection for early prediction of antimicrobial multidrug resistance. *Future Generation Computer Systems*, 133 (2022), pp. 68–83. DOI: 10.1016/j.future.2022.02.021.

Microsoft (2025). *Microsoft Azure*. Available at: https://azure.microsoft.com/en-us/get-started/azure-portal/ [Accessed on 22nd Aug. 2025].

Monter-Pozos, A. and González-Estrada, E. (2024). On testing the skew normal distribution by using Shapiro–Wilk test. *Journal of Computational and Applied Mathematics*, 440 (2024), p. 115649. DOI: 10.1016/j.cam.2023.115649.

Nascimento, G. F. M., Wurtz, F., Kuo-Peng, P. et al. (2021). Outlier Detection in Buildings' Power Consumption Data Using Forecast Error. *Energies*, 14 (24), p. 8325. DOI: 10.3390/en14248325.

NHS UK (2025). *Chlamydia*. Available at: https://www.nhs.uk/conditions/chlamydia/ [Accessed on 14th May 2025].

Noroozi, M., Mohammadi, H., Efatinasab, E. et al. (2022). Golden Search Optimisation Algorithm. *IEEE Access*, 10 (2022), pp. 37515–37532. DOI: 10.1109/ACCESS.2022.3162853.

Ranja, F., Nababan, E. B. and Candra, A. (2023). Synthetic Data Generation Using Time-Generative Adversarial Network (Time-GAN) to Predict Cash ATM. In: *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. 4-5 October 2023. Bandung, Indonesia: IEEE Xplorer, pp. 418–423. DOI: 10.1109/IC3INA60834.2023.10285809.

Sahoo, R. K., Sahoo, K. C., Negi, S. et al. (2025). Health professionals' perspectives on the use of Artificial Intelligence in healthcare: A systematic review. *Patient Engineering and Counseling*, 134 (2025), p. 108680. DOI: 10.1016/j.pec.2025.108680.

Syed, A. B., Hasan, R., Chowdhury, N. I. et al. (2025). A systematic review of time series algorithms and analytics in predictive maintenance. *Decision Analytics Journal*, 15 (15), p. 100573. DOI: 10.1016/j.dajour.2025.100573.

Tanveer, H. and Latif, S. (2024). Decoding Stock Market Predictions: Insights from Explainable AI Using Layer-wise Relevance Propagation. In: *2024 26th International*

*Multi-Topic Conference (INMIC)*. 30-31 December 2024. Karachi, Pakistan: IEEE Xplorer, pp. 1–6. DOI: 10.1109/INMIC64792.2024.11004356.

Triggle, N. (2025). Major breast cancer screening AI trial to begin. *BBC News*. 4 February. Available at: https://www.bbc.co.uk/news/articles/cly7gx2gx3eo [Accessed on 1st June 2025].

Tuddenham, S., Hamill, M. M. and Ghanem, K. G. (2022). Diagnosis and Treatment of Sexually Transmitted Infections. *JAMA*, 327 (2), pp. 161–172. DOI: 10.1001/jama.2021.23487.

UK, N. (2025). *Gonorrhoea*. Available at: https://www.nhs.uk/conditions/gonorrhoea/ [Accessed on 14th May 2025].

UK Health Security Agency (2025). *Sexually transmitted infections and screening for chlamydia in England: 2024 report*. Available at: https://www.gov.uk/government/statistics/sexually-transmitted-infections-stis-annual-data-tables/sexually-transmitted-infections-and-screening-for-chlamydia-in-england-2024-report [Accessed on 14th May 2025].

Ungureanu, I. L., Portase, R.-L., Dinsoreanu, M. et al. (2024). Hybrid Statistical and AI Methods for Synthetic Time Series Generation in Practical Applications. In: *2024 IEEE 20th International Conference on Intelligent Computer Communication and Processing (ICCP)*. 17-19 October 2024. Cluj-Napoca, Romania: IEEE Xplorer, pp. 1–8. DOI: 10.1109/ICCP63557.2024.10793008.

Verma, S., Pant, M. and Snasel, V. (2021). A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access*, 9, pp. 57757–57791. DOI: 10.1109/ACCESS.2021.3070634.

Wang, J. and Zhao, Y. (2021). Time Series K-Nearest Neighbors Classifier Based on Fast Dynamic Time Warping. In: *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. 28-30 June 2021. Dalian, China: IEEE Xplorer. DOI: 10.1109/ICAICA52286.2021.9497898.

Xu, Q., Xu, Z. and Ma, T. (2020). A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition: Variants, Challenges and Future Directions. *IEEE Access*, 8, pp. 41588–41614. DOI: 10.1109/ACCESS.2020.2973670.

Zhang, Q. and Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computations*, 11 (6), pp. 712–731. DOI: 10.1109/TEVC.2007.892759.

Zheng, Y., Yu, Q., Zhou, Y. et al. (2022). Global burden and trends of sexually transmitted infections from 1990 to 2019: an observational trend study. *The Lancet Infectious Diseases*, 22 (4), pp. 541–551. DOI: 10.1016/S1473-3099(21)00448-5.

Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. *TIK Report*, 103, pp. 1–21. DOI: 10.3929/ETHZ-A-004284029.

# Bibliography

AIO Photos (2025). *Agile Methodology Vs Waterfall Model*. Available at: https://www.aiophotoz.com/photos/agile-methodology-vs-waterfall-model.html [Accessed on 25th May 2025].

Aladdin, A. M. and Rashid, T. A. (2023). A New Lagrangian Problem Crossover—A Systematic Review and Meta-Analysis of Crossover Standards. *Systems*, 11 (3), p. 144. DOI: 10.3390/systems11030144.

Alanazi, A. (2022). Using machine learning for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, 30 (1), p. 100924. DOI: 10.1016/j.imu.2022.100924.

Alduailij, M., Khan, Q. W., Tahir, M. et al. (2022). Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method. *Symmetry*, 14 (6), p. 1095. DOI: 10.3390/sym14061095.

Amazon Web Services (2025). *Amazon Web Services*. Available at: https://aws.amazon.com/ [Accessed on 22nd Aug. 2025].

Anderson, S. (2024). Expanding data literacy to include data preparation: building a sound marketing analytics foundation. *Journal of Marketing Analytics*, 12 (2), pp. 227–234. DOI: 10.1057/s41270-024-00293-3.

Athalye, A., Carlini, N. and Wagner, D. (July 2018). Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In: *Proceedings of the 35th International Conference on Machine Learning*. Stockholmsmässan, Stockholm, Sweden: PMLR, pp. 274–283.

Awotunde, J. B., Adeniyi, A. E., Ajagbe, S. A. et al. (2022). 'Swarm Intelligence and Evolutionary Algorithms in Processing Healthcare Data'. In: *Connected e-Health: Integrated IoT and Cloud Computing*. Ed. by S. Mishra, A. González-Briones, A. K. Bhoi et al. Cham, Germany: Springer International Publishing, pp. 105–124. DOI: 10.1007/978-3-030-97929-4_5.

Bagheri, A., Patrignani, A., Ghanbarian, B. et al. (2025). A hybrid time series and physics-informed machine learning framework to predict soil water content. *Engineering Applications of Artificial Intelligence*, 144 (2025), p. 110105. DOI: 10.1016/j.engappai.2025.110105.

Barrasdas, A., Tejeda-Gil, A. and Canton-Croda, R.-M. (2022). Real-Time Big Data Architecture for Processing Cryptocurrency and Social Media Data: A Clustering Approach Based on k-Means. *Algorithms*, 15, p. 140. DOI: 10.3390/a15050140.

Bayram, F., Ahmed, B. S. and Kassler, A. (2022). From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 245 (2022), p. 108632. DOI: 10.1016/j.knosys.2022.108632.

Bhosale, Y. H. and Patnaik, K. S. (2023). PulDi-COVID: Chronic obstructive pulmonary (lung) diseases with COVID-19 classification using ensemble deep convolutional neural network from chest X-ray images to minimize severity and mortality rates. *Biomedical Signal Processing and Control*, 81 (2023), p. 104445. DOI: 10.1016/j.bspc.2022.104445.

Blasco, T., Sanchez, J. S. and Garcia, V. (2024). A survey on uncertainty quantification in deep learning for financial time series prediction. *Neurocomputing*, 576 (2024), p. 127339. DOI: 10.1016/j.neucom.2024.127339.

Carter, J. G., Iturbe, L. O., Duprey, J.-L. et al. (2022). Ultrarapid detection of SARS-CoV-2 RNA using a reverse transcription–free exponential amplification reaction, RTF-EXPAR. *Applied Physical Sciences*, 118 (35). DOI: 10.1073/pnas.2100347118.

Caruso, G., Giammanco, A., Virruso, R. et al. (2021). Current and Future Trends in the Laboratory Diagnosis of Sexually Transmitted Infections. *International Journal of Environmental Research and Public Health*, 18 (3), p. 1038. DOI: 10.3390/ijerph18031038.

Costa, B. and Georgieva, P. (2023). Explainable Artificial Intelligence in Healthcare Applications: A Systematic Review. In: *2023 International Scientific Conference on Computer Science (COMSCI)*. 18-20 September 2023. Sozopol, Bulgaria: IEEE Xplorer, pp. 1–8. DOI: 10.1109/COMSCI59259.2023.10315829.

Dasari, K. B. and Devarakonda, N. (2022). TCP/UDP-Based Exploitation DDoS Attacks Detection Using AI Classification Algorithms with Common Uncorrelated Feature Subset Selected by Pearson, Spearman and Kendall Correlation Methods. *Revue d'Intelligence Artificielle*, 36 (1), pp. 61–71. DOI: 10.18280/ria.360107.

Deb, K., Pratap, A., Agarwal, S. et al. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, pp. 182–197. DOI: 10.1109/4235.996017.

Du, K.-L., Jiang, B., Lu, J. et al. (2024). Exploring Kernel Machines and Support Vector Machines: Principles, Techniques, and Future Directions. *Mathematics*, 12 (24), p. 3935. DOI: 10.3390/math12243935.

Dumitrescu, E. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, 297 (2022), pp. 1178–1192. DOI: 10.1016/j.ejor.2021.06.053.

Efimov, V. (2024). *Demystifying Confidence Intervals with Examples*.

Emmerich, M., Deutz, A., Wang, H. et al., eds. (2023). *Evolutionary Multi-Criterion Optimization - 12th International Conference*. Leiden, Netherlands, 20-24 March 2023. New York, NY: Springer.

Endalamaw, A., Khatri, R. B., Erku, D. et al. (2024). Barriers and strategies for primary health care workforce development: synthesis of evidence. *BMC Primary Care*, 25 (99). DOI: 10.1186/s12875-024-02336-1.

Fadhil, S., Zaher, H., Ragaa, N. et al. (2023). A Modified Differential Evolution Algorithm Based on Improving A New Mutation Strategy and Self-Adaptation Crossover. *MethodsX*, 11 (2023), p. 102276. DOI: 10.1016/j.mex.2023.102276.

Fang, J., Li, R., Zhang, T. et al. (2025). Time series identification, classification, and display methods for periodic flow patterns. *Flow Measurement and Instrumentation*, 106 (2025), p. 102977. DOI: 10.1016/j.flowmeasinst.2025.102977.

Folgado, D., Barandas, M., Famiglini, L. et al. (2023). Explainability meets uncertainty quantification: Insights from feature-based model fusion on multimodal time series. *Information Fusion*, 100 (2023), p. 101955. DOI: 10.1016/j.inffus.2023.101955.

Gani, R., Isty, M. N., Rimi, R. A. et al. (2024). Impact of AI in Healthcare Services: Analysis Using Medical Synthetic Data. In: *Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*. 8-9 March 2024. Dhaka, Bangladesh: IEEE Xplorer, pp. 1–6. DOI: 10.1109/iCACCESS61735.2024.10499613.

Geyer, P., Singh, M. M. and Chen, X. (2024). Explainable AI for engineering design: A unified approach of systems engineering and component-based deep learning demonstrated by energy-efficient building design. *Advanced Engineering Informatics*, 62 (2024), p. 102843. DOI: 10.1016/j.aei.2024.102843.

Google (2025). *Google Cloud*. Available at: https://cloud.google.com/ [Accessed on 22nd Aug. 2025].

Hernandez, M., Epelde, G., Alberdi, A. et al. (2022). Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493 (2022), pp. 28–45. DOI: 10.1016/j.neucom.2022.04.053.

Hu, L., Zhang, F., Qin, M. et al. (2022). A Dynamic Pyramid Tilling Method for Traffic Data Stream Based on Flink. *IEEE Transactions on Intelligent Transportation Systems*, 23 (7), pp. 6679–6688. DOI: 10.1109/TITS.2021.3060576.

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110 (2021), pp. 457–506. DOI: 10.1007/s10994-021-05946-3.

Ibrahim, M. (2024). A Deep Dive Into Learning Curves in Machine Learning. *Domain Agnostic*. [blog] 13 March. Available at: https://wandb.ai/mostafaibrahim17/ml-articles/reports/A-Deep-Dive-Into-Learning-Curves-in-Machine-Learning--Vmlldzo0NjA1ODY0 [Accessed on 20th Apr. 2024].

Irawan, B., Fahmi, F. and Zamzami, E. M. (2024). Optimizing K-Nearest Neighbor Values Using The Elbow Method. In: *IEEE International Conference on Informatics and Computing (ICIC)*. 24-25 October 2024. Medan, Indonesia: IEEE Xplorer. DOI: 10.1109/ICIC64337.2024.10957541.

Irfan, D., Gunawan, T. S. and Wanayumini (2023). Comparison of SGD, RMSprop, and Adam Optimisation in Animal Classification Using CNNs. In: *2nd International Conference on Information Science and Technology Innovation (ICoSTEC)*. 25 Feb-

ruary 2023. Yogyakarta, Indonesia: Universitas Respati Yogyakarta. DOI: 10.35842/icostec.v2i1.32.

König, T., Ramananda, A. M., Wagner, F. et al. (2024). A LSTM-GAN Algorithm for Synthetic Data Generation of Time Series Data for Condition Monitoring. *Procedia Computer Science*, 246 (2024), pp. 1508–1517. DOI: 10.1016/j.procs.2024.09.602.

Lamba, R., Gulati, T. and Jain, A. (2021). A Hybrid Feature Selection Approach for Parkinson's Detection Based on Mutual Information Gain and Recursive Feature Elimination. *Arabian Journal for Science and Engineering*, 47 (2021), pp. 10263–10276. DOI: https://doi.org/10.1007/s13369-021-06544-0.

Lee, S., Choi, C., Do, H. et al. (2025). Batch active learning for time-series classification with multi-mode exploration. *Information Sciences*, 711 (2025), p. 122109. DOI: 10.1016/j.ins.2025.122109.

Li, G. and Zhang, C. (2022). HTAP Databases: What is New and What is Next. In: *Proceedings of the 2022 International Conference on Management of Data*. 11 June 2022. Philadelphia, USA: Association for Computing Machinery, pp. 2483–2488. DOI: 10.1145/3514221.3522565.

Linear Diagnostics (2025). *Next generation of near-patient diagnostics*. Available at: https://www.lineardiagnostics.com/ [Accessed on 3rd June 2025].

Liu, L., Fei, T., Zhu, Z. et al. (Aug. 2023). A Survey of Evolutionary Algorithms. In: *4th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. Hangzhou, China: IEEE Xplorer, pp. 22–27. DOI: 10.1109/ICBAIE59714.2023.10281260.

Loglisci, C., Impedovo, A., Calders, T. et al. (2024). Heuristic approaches for non-exhaustive pattern-based change detection in dynamic networks. *Journal of Intelligent Information Systems*, 62 (2024), pp. 1455–1492. DOI: 10.1007/s10844-024-00866-9.

Martínez-Agüero, S., Soguero-Ruiz, C., Alonso-Moral, J. M. et al. (2022). Interpretable clinical time-series modeling with intelligent feature selection for early prediction of antimicrobial multidrug resistance. *Future Generation Computer Systems*, 133 (2022), pp. 68–83. DOI: 10.1016/j.future.2022.02.021.

Microsoft (2025). *Microsoft Azure*. Available at: https://azure.microsoft.com/en-us/get-started/azure-portal/ [Accessed on 22nd Aug. 2025].

Monter-Pozos, A. and González-Estrada, E. (2024). On testing the skew normal distribution by using Shapiro–Wilk test. *Journal of Computational and Applied Mathematics*, 440 (2024), p. 115649. DOI: 10.1016/j.cam.2023.115649.

Nascimento, G. F. M., Wurtz, F., Kuo-Peng, P. et al. (2021). Outlier Detection in Buildings' Power Consumption Data Using Forecast Error. *Energies*, 14 (24), p. 8325. DOI: 10.3390/en14248325.

NHS UK (2025). *Chlamydia*. Available at: https://www.nhs.uk/conditions/chlamydia/ [Accessed on 14th May 2025].

Noroozi, M., Mohammadi, H., Efatinasab, E. et al. (2022). Golden Search Optimisation Algorithm. *IEEE Access*, 10 (2022), pp. 37515–37532. DOI: 10.1109/ACCESS.2022.3162853.

Nouis, S. C., Uren, V. and Jariwala, S. (2025). Evaluating accountability, transparency, and bias in AI-assisted healthcare decision- making: a qualitative study of healthcare professionals' perspectives in the UK. *BMC Medical Ethics*, 26 (1). DOI: 10.1186/s12910-025-01243-z.

Ogundokun, R. O., Misra, S., Douglas, M. et al. (2022). Medical Internet-of-Things Based Breast Cancer Diagnosis Using Hyperparameter-Optimized Neural Networks. *Future Internet*, 14 (5), p. 153. DOI: 10.3390/fi14050153.

Rahman, T., Chowdhury, M. and Khandakar, A. (2022). *COVID-19 Radiography Database*. Available at: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database [Accessed on 21st July 2025].

Ranja, F., Nababan, E. B. and Candra, A. (2023). Synthetic Data Generation Using Time-Generative Adversarial Network (Time-GAN) to Predict Cash ATM. In: *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. 4-5 October 2023. Bandung, Indonesia: IEEE Xplorer, pp. 418–423. DOI: 10.1109/IC3INA60834.2023.10285809.

Roberts, A. (2025). *MSc Notebook Files*. Available at: https://github.com/al-roberts-computing/msc-notebook-files [Accessed on 28th Sept. 2025].

Sahoo, R. K., Sahoo, K. C., Negi, S. et al. (2025). Health professionals' perspectives on the use of Artificial Intelligence in healthcare: A systematic review. *Patient Engineering and Counseling*, 134 (2025), p. 108680. DOI: 10.1016/j.pec.2025.108680.

Syed, A. B., Hasan, R., Chowdhury, N. I. et al. (2025). A systematic review of time series algorithms and analytics in predictive maintenance. *Decision Analytics Journal*, 15 (15), p. 100573. DOI: 10.1016/j.dajour.2025.100573.

Tanveer, H. and Latif, S. (2024). Decoding Stock Market Predictions: Insights from Explainable AI Using Layer-wise Relevance Propagation. In: *2024 26th International Multi-Topic Conference (INMIC)*. 30-31 December 2024. Karachi, Pakistan: IEEE Xplorer, pp. 1–6. DOI: 10.1109/INMIC64792.2024.11004356.

Triggle, N. (2025). Major breast cancer screening AI trial to begin. *BBC News*. 4 February. Available at: https://www.bbc.co.uk/news/articles/cly7gx2gx3eo [Accessed on 1st June 2025].

Tuddenham, S., Hamill, M. M. and Ghanem, K. G. (2022). Diagnosis and Treatment of Sexually Transmitted Infections. *JAMA*, 327 (2), pp. 161–172. DOI: 10.1001/jama.2021.23487.

UK, N. (2025). *Gonorrhoea*. Available at: https://www.nhs.uk/conditions/gonorrhoea/ [Accessed on 14th May 2025].

UK Health Security Agency (2025). *Sexually transmitted infections and screening for chlamydia in England: 2024 report*. Available at: https://www.gov.uk/government/statistics/sexually-transmitted-infections-stis-annual-data-tables/sexually-transmitted-infections-and-screening-for-chlamydia-in-england-2024-report [Accessed on 14th May 2025].

Ungureanu, I. L., Portase, R.-L., Dinsoreanu, M. et al. (2024). Hybrid Statistical and AI Methods for Synthetic Time Series Generation in Practical Applications. In: *2024 IEEE 20th International Conference on Intelligent Computer Communication and Processing (ICCP)*. 17-19 October 2024. Cluj-Napoca, Romania: IEEE Xplorer, pp. 1–8. DOI: 10.1109/ICCP63557.2024.10793008.

Vatter, J., Mayer, R. and Jacobsen, H.-a. (2023). The Evolution of Distributed Systems for Graph Neural Networks and Their Origin in Graph Processing and Deep Learning: A Survey. *ACM Computing Surveys*, 56 (1), pp. 1–37. DOI: 10.1145/3597428.

Verma, S., Pant, M. and Snasel, V. (2021). A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access*, 9, pp. 57757–57791. DOI: 10.1109/ACCESS.2021.3070634.

Wang, J. and Zhao, Y. (2021). Time Series K-Nearest Neighbors Classifier Based on Fast Dynamic Time Warping. In: *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. 28-30 June 2021. Dalian, China: IEEE Xplorer. DOI: 10.1109/ICAICA52286.2021.9497898.

Xu, Q., Xu, Z. and Ma, T. (2020). A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition: Variants, Challenges and Future Directions. *IEEE Access*, 8, pp. 41588–41614. DOI: 10.1109/ACCESS.2020.2973670.

Yuan, D., Zhang, L., Dong, J. et al. (2024). Snowy Dove: An open-source toolkit for preprocessing of Chinese Gaofen series data. *PLoS One*, 19 (11), pp. 1–15. DOI: https://doi.org/10.1371/journal.pone.0313584.

Zhang, C., Li, G., Zhang, J. et al. (2024). HTAP Databases: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 36 (11), pp. 6410–6429. DOI: 10.1109/TKDE.2024.3389693.

Zhang, Q. and Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computations*, 11 (6), pp. 712–731. DOI: 10.1109/TEVC.2007.892759.

Zheng, Y., Yu, Q., Zhou, Y. et al. (2022). Global burden and trends of sexually transmitted infections from 1990 to 2019: an observational trend study. *The Lancet Infectious Diseases*, 22 (4), pp. 541–551. DOI: 10.1016/S1473-3099(21)00448-5.

Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. *TIK Report*, 103, pp. 1–21. DOI: 10.3929/ETHZ-A-004284029.

# Appendix A Dataset Snapshots

| | | Type of Sample | V | V | V | V | | EC | EC | EC | EC | EC | EC | EC | EC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time Axis | | Sample Number | 1-320-129 | | | | | 1-320-132-1 | | | | 1-320-132-2 | | | |
| Mins | Seconds | | | | | | | | | | | | | | |
| 0 | 0 | | 3.01129 | 2.885547 | 2.875749 | 2.934538 | | 2.238667 | 2.195359 | 2.008804 | 2.012135 | 2.613958 | 2.703352 | 2.999678 | 3.011266 |
| 0.5 | 30 | | 3.09784 | 2.967198 | 2.970464 | 3.022721 | | 2.350267 | 2.305294 | 2.100416 | 2.107079 | 2.76957 | 2.877174 | 3.198332 | 3.198332 |
| 1 | 60 | | 3.141931 | 3.009657 | 3.02762 | 3.068446 | | 2.43022 | 2.396906 | 2.180368 | 2.190362 | 2.913594 | 3.02782 | 3.360566 | 3.3556 |
| 1.5 | 120 | | 3.186023 | 3.058647 | 3.088042 | 3.11417 | | 2.560142 | 2.543485 | 2.301963 | 2.313622 | 3.148668 | 3.272827 | 3.598951 | 3.579086 |
| 2 | 150 | | 3.228482 | 3.106005 | 3.158262 | 3.166427 | | 2.766686 | 2.775014 | 2.495181 | 2.508506 | 3.380432 | 3.493002 | 3.762841 | 3.734698 |
| 2.5 | 189 | | 3.251344 | 3.132133 | 3.203986 | 3.190922 | | 2.941582 | 2.97323 | 2.701725 | 2.716716 | 3.521145 | 3.63206 | 3.875411 | 3.840647 |
| 3 | 228 | | 3.27094 | 3.154996 | 3.236647 | 3.213784 | | 3.041522 | 3.088161 | 2.868292 | 2.884949 | 3.643648 | 3.757874 | 3.983016 | 3.943285 |
| 3.5 | 267 | | 3.290536 | 3.172959 | 3.266041 | 3.235014 | | 3.12314 | 3.18477 | 2.979892 | 2.994883 | 3.764496 | 3.873756 | 4.080687 | 4.039301 |
| 4 | 306 | | 3.310133 | 3.194188 | 3.297068 | 3.256243 | | 3.206424 | 3.276382 | 3.074836 | 3.093158 | 3.875411 | 3.986327 | 4.176704 | 4.12704 |
| 4.5 | 345 | | 3.332995 | 3.21705 | 3.326463 | 3.277472 | | 3.279714 | 3.361332 | 3.168114 | 3.181439 | 3.98136 | 4.09062 | 4.266098 | 4.214779 |
| 5 | 384 | | 3.352591 | 3.23828 | 3.349325 | 3.295435 | | 3.349672 | 3.447947 | 3.253063 | 3.271385 | 4.083998 | 4.194914 | 4.352181 | 4.294241 |
| 5.5 | 423 | | 3.37382 | 3.259509 | 3.377086 | 3.318298 | | 3.41963 | 3.526234 | 3.338012 | 3.356335 | 4.18167 | 4.29093 | 4.438265 | 4.373702 |
| 6 | 462 | | 3.393417 | 3.280738 | 3.404848 | 3.342793 | | 3.482926 | 3.606186 | 3.416299 | 3.441284 | 4.274375 | 4.380324 | 4.511105 | 4.448198 |
| 6.5 | 501 | | 3.416279 | 3.301967 | 3.430976 | 3.362389 | | 3.551219 | 3.67781 | 3.499583 | 3.517905 | 4.358803 | 4.468063 | 4.5856 | 4.517727 |

Figure A.1: A snapshot of the original CT dataset.

| sample_id | sample_type | overall_result | gender | duplicate |
|---|---|---|---|---|
| 1-320-129 | V | positive | F | no |
| 1-320-132 | EC | positive | F | no |
| 1-320-133 | HV | positive | F | no |
| 1-320-135 | HV | positive | F | no |
| 1-320-136 | HV | positive | F | no |
| 1-320-137 | EC | positive | F | no |
| 1-320-138 | HV | positive | F | no |
| 1-320-196 | HV | positive | F | no |
| 1-320-197 | HV | positive | F | no |
| 1-320-198 | EC | positive | F | no |
| 1-320-199 | HV | positive | F | no |
| 1-320-200 | HV | positive | F | no |
| 1-320-201 | HV | positive | F | no |
| 1-320-202 | HV | positive | F | no |
| 1-320-203 | HV | positive | F | no |
| 1-320-205 | HV | positive | F | no |
| 1-320-206 | V | positive | F | no |
| 1-320-246 | V | positive | F | no |
| 1-320-59 | M | positive | M | no |
| 1-320-121 | M | positive | M | no |
| 1-320-143 | M | positive | M | no |
| 1-320-144 | M | positive | M | no |
| 1-320-146 | M | positive | M | no |
| 1-320-147 | M | positive | M | no |
| 1-320-150 | M | positive | M | no |
| 1-320-152 | M | positive | M | no |
| 1-320-156 | M | positive | M | no |
| 1-320-158 | M | positive | M | no |

Figure A.2: A snapshot of the additional CT dataset information.

| Time (mins) | A2 | A3 | A5 | A7 | A8 | A9 | A10 | A12 | B1 | B2 | B3 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 34638.973 | 29757.746 | 27771.76 | 31440.324 | 33098.164 | 32426.812 | 37602.48 | 29292.395 | 10075.057 | 14099.583 | 23776 |
| 2 | 34383.457 | 29281.486 | 27626.613 | 31230.99 | 33214.445 | 32434.96 | 37579.93 | 29291.04 | 10093.724 | 15420.409 | 23679 |
| 3 | 33659.113 | 28691.936 | 27536.602 | 30853.24 | 34666.5 | 32584.76 | 37586.7 | 29295.074 | 10648.04 | 19854.574 | 23701 |
| 4 | 33432.145 | 28565.412 | 27596.457 | 30064.596 | 35794.016 | 32350.611 | 37505.965 | 29453.002 | 12057.501 | 21209.684 | 23978. |
| 5 | 33472.84 | 28522.031 | 27179.41 | 29898.494 | 35985.582 | 32416.16 | 37277.945 | 30010.871 | 12497.723 | 21049.426 | 24145 |
| 6 | 33724.562 | 28447.979 | 27073.24 | 30038.293 | 36271.414 | 32993.66 | 37352.523 | 30004.797 | 13684.104 | 20886.533 | 24823. |
| 7 | 33780.473 | 28560.162 | 27325.637 | 30197.398 | 36325.074 | 33293.32 | 37886.99 | 29556.176 | 16047.214 | 20911.71 | 25456. |
| 8 | 33909.066 | 29164.535 | 27615.635 | 30248.207 | 36743.99 | 33846.51 | 38091.75 | 29725.408 | 19666.03 | 21205.24 | 25908. |
| 9 | 34612.727 | 29564.666 | 27319.752 | 30470.148 | 36818.434 | 34320.51 | 38312.32 | 30926.176 | 19786.545 | 21420.988 | 26065. |
| 10 | 34800.98 | 29602.562 | 26893.074 | 30625.668 | 37042.766 | 34414.348 | 38389.59 | 31184.824 | 19312.55 | 22350.453 | 26470. |
| 11 | 34853.79 | 29968.518 | 26961.404 | 31186.912 | 37052.99 | 34820.63 | 38761.918 | 31486.367 | 19038.963 | 22443.957 | 26664. |
| 12 | 35364.594 | 30779.463 | 27583.951 | 31069.012 | 37281.285 | 34777.457 | 38815.18 | 31783.24 | 18937.852 | 22695.795 | 26682. |
| 13 | 35465.305 | 31173.43 | 27818.33 | 30961.28 | 36932.492 | 34705.594 | 38864.395 | 31808.379 | 18907.514 | 22238.021 | 27125 |
| 14 | 35798.13 | 30765.057 | 28370.28 | 31084.154 | 36855.082 | 34932.98 | 38321.926 | 31856.059 | 19089.137 | 23370.166 | 27115 |
| 15 | 36194.383 | 30562.635 | 28282.135 | 31022.69 | 37515.965 | 34972.938 | 38345.914 | 32128.248 | 19179.777 | 29824.031 | 27112. |
| 16 | 36795.08 | 30922.068 | 28033.094 | 31025.305 | 37466.324 | 34936.72 | 39294.914 | 33644.56 | 19579.498 | 32901.62 | 27055. |
| 17 | 36691.965 | 31938.738 | 27943.145 | 31451.7 | 37164.62 | 34978.355 | 39379.45 | 34166.137 | 19412.95 | 33557.004 | 27424. |
| 18 | 36610.082 | 31968.244 | 28207.04 | 32138.9 | 37106.61 | 35632.324 | 39058.844 | 34558.42 | 19388.506 | 33912.08 | 2781 |
| 19 | 37113.6 | 31622.578 | 28772.111 | 32552.945 | 37749.53 | 35692.223 | 39037.03 | 34511.043 | 19908.771 | 33885.312 | 27889. |
| 20 | 37460.625 | 31680.223 | 29057.842 | 32416.031 | 38115.273 | 35597.26 | 39758.98 | 34587.1 | 19990.682 | 34823.418 | 28485 |
| 21 | 37601.508 | 32716.053 | 28855.443 | 32008.633 | 38614.734 | 35900.074 | 40360.777 | 35675.06 | 20641.479 | 35012.6 | 28825. |

Figure A.3: A snapshot of the original COVID-19 dataset.

| Time (min) | 1450 Copies/uL | | | 725 Copies/uL | | | 72.5 Copies/uL | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.25 | 16368.045 | 9663.707 | 11759.481 | 5199.029 | 4488.901 | 4671.888 | 8968.779 | 3134.61 | 5199.206 |
| 0.5 | 16248.674 | 9655.151 | 11735.7 | 4916.828 | 4328.003 | 4343.35 | 9323.482 | 3125.417 | 5141.153 |
| 0.75 | 16365.661 | 9754.741 | 11728.559 | 4546.627 | 4207.605 | 4087.741 | 9600.963 | 3481.459 | 4771.488 |
| 1 | 17166.691 | 9741.088 | 11196.713 | 4109.122 | 4218.261 | 3865.515 | 9512.777 | 3302.216 | 4303.159 |
| 1.25 | 17567.613 | 9690.267 | 10953.167 | 3817.162 | 4163.784 | 3394.191 | 8799.017 | 3017.482 | 4074.528 |
| 1.5 | 17173.19 | 9460.537 | 11093.968 | 3468.181 | 3909.574 | 3440.231 | 8613.091 | 2691.229 | 4006.127 |
| 1.75 | 17272.553 | 9516.629 | 11762.746 | 3383.068 | 3871.459 | 4065.839 | 8842.538 | 2050.903 | 4096.655 |
| 2 | 18385.488 | 10108.494 | 11858.778 | 3380.18 | 4083.47 | 4182.136 | 8800.98 | 2046.462 | 3970.551 |
| 2.25 | 18836.412 | 10691.067 | 12180.813 | 3595.778 | 4637.344 | 4430.752 | 8540.52 | 2784.974 | 3752.142 |
| 2.5 | 20031.861 | 10992.322 | 13494.644 | 4455.661 | 5255.38 | 5077.248 | 7875.588 | 2887.32 | 3278.879 |
| 2.75 | 20688.06 | 12333.044 | 14025.792 | 4625.6 | 5686.283 | 5569.729 | 7707.94 | 2505.369 | 3375.71 |
| 3 | 21417.867 | 13798.367 | 14764.607 | 4847.033 | 6149.418 | 6384.068 | 7702.777 | 2032.529 | 4227.059 |
| 3.25 | 21658.445 | 15406.497 | 16300.864 | 5138.115 | 6489.237 | 7379.928 | 7777.645 | 1912.136 | 4345.416 |

Figure A.4: A snapshot of the original PNAS dataset.

# Appendix B Literature Review Meta-analyses

Please refer to the glossary for the metric abbreviations contained within the following table.

Table B.1: A summary of key sources and their contributions.

| Citation | Dataset | Contribution | Key Metrics | Results |
|---|---|---|---|---|
| Bagheri et al. (2025) | Soil moisture readings, gathered from several locations across the US. | Integrating physics-based rules into neural network predictions to help improve predictions and reduce uncertainty. | RMSE, NRMSE, MAPE | Reduced epistemic and aleatoric uncertainties through the use of physics-informed machine learning. |
| Blasco, Sanchez and Garcia (2024) | No data used. | A systematic survey of the importance uncertainty quantification has on financial forecasting, including its practical potential. | No metrics shown. | A well-balanced review of several uncertainty quantification techniques for predictive analytics regarding time series stock market data. |
| Costa and Georgieva (2023) | Synthetic healthcare dataset of physiological measurements. | Investigates the importance of statistical techniques (e.g., T-test and ANOVA) for evaluating AI efficiency and accuracy, and other variables of an AI healthcare system. | Sum squares and mean squares. | Successfully illustrated the importance of using hypothesis testing to validate explainable AI findings, such as multiple linear regression. |

| Continuation of Table B.1 | | | | |
|---|---|---|---|---|
| Fang et al. (2025) | Sensor readings and images relating to flow patterns in rocket engines. | Investigated how a combination of image recognition and time series processing could be applied to predict and describe flow patterns in rocket engines. | Accuracy, precision, recall, F1-score, Log Loss | Successful results with over 88% accuracy. Image recognition outperformed the time series analysis aspect (as it reached almost 100%). |
| Folgado et al. (2023) | Human physiological data with numerous features. | Leveraged a novel multimodal ensemble method (based on model uncertainty) to create an overall model with reduced explanation complexity. | Accuracy and F1-score | The final model had a slight reduction in evaluation performance compared to the model that did not leverage their approach. But explainability metrics and uncertainty in the former was significantly reduced. |
| Gani et al. (2024) | No data used. | A systematic review of examining healthcare professional attitudes and concerns towards integrating AI into healthcare. | No metrics shown. | An overview of several advantages and disadvantages of applying AI in healthcare, from a higher level, less technical standpoint. |
| Geyer, Singh and Chen (2024) | Several building-related measurements and sensor readings (e.g., heating levels, lightning availability). | Utilising system decomposition to build component-based machine learning frameworks for improved AI explainability and interpretability. | $R^2$, MAPE, RMSE, SA | Improved generalisation (compared to monolithic models), explainability and interpretability of data-driven black-box machine learning. |

| Continuation of Table B.1 | | | | |
|---|---|---|---|---|
| Hernandez et al. (2022) | No data used. | Performed a systematic review of current literature on synthetic data generation. | Resemblance, Utility and Privacy. | A broad overview of several techniques and approaches for creating synthetic data to help with class imbalances and privacy preservation. |
| König et al. (2024) | Vibration and acoustic sensor data for machines. | Illustrated how a hybrid neural network (LSTM-GAN) can be leveraged to generate usable synthetic data. | RMSE, kurtosis values, summary statistics | Increased synthetic data-supplemented model accuracy from 65% to 100% for a limited number of samples. |
| Lee et al. (2025) | Several different datasets. | Presented a new active learning framework for improved training performance (particularly for classification) for neural networks. | Macro-average F1-score | Outperformed other benchmarked active learning methods by an average F1-score of $\approx 5-7\%$, as well as a significant reduction in model training time. |
| Martínez-Agüero et al. (2022) | Intensive Care Unit patient data at University Hospital of Fuenlabrada in Madrid. | Attempted to leverage temporal-specialised neural networks (RNNs) to predict to what degree ICU patients could be infected with drug-resistant bacteria. | Accuracy, specificity and sensitivity | Most evaluation metrics ranging between 55% and 65%. But explainability and interpretability are promising. |

| Continuation of Table B.1 | | | | |
|---|---|---|---|---|
| Ranja, Nababan and Candra (2023) | Cash machine money availability over time. | Utilised synthetic data generation by TimeGAN to help mitigate class imbalances in a time series. | MAE, MSE, $R^2$ | Although the synthetic data looked authentic and superimposable, it became apparently discernible from the real data upon using principal component analysis. |
| Sahoo et al. (2025) | No data used. | A systematic review of current ethical issues with utilising artificial intelligence in clinical outputs, particularly centred around model explainability. | No metrics. | Presents an explainability-focused review which identifies key challenges and provides direction for tackling them. |
| Syed et al. (2025) | No data used. | Systematically reviewing and summarising over fifty sources relating to time series for use cases involving predictive maintenance. | RMSE, NRMSE, MAE, Accuracy, Recall, Precision, F1-score | A clear, comprehensive and in-depth review of useful time-series algorithms, along with key preprocessing and data augmentation steps specifically for time-series scenarios. |

| Continuation of Table B.1 | | | | |
|---|---|---|---|---|
| Tanveer and Latif (2024) | Pakistan Stock Exchange | Leveraging a Conv1D-BiLSTM model to predict whether the Pakistan Stock Exchange was in a state or stability or instability. This was accompanied by explainability evaluation methods such as SHAP, LIME and LRP. | Accuracy | A fair accuracy of around 80% with brief justifications for explainability. But the use of only the accuracy measurement limits the significance of model results. |
| Ungureanu et al. (2024) | Household appliance data | Proposed time series decomposition with residual replacement and time series decomposition with seasonality and residual replacement as more explainable ML models, especially for univariate time series. | Accuracy | Increased accuracy on average of about 10% (with new values reaching around 86% to 90% accuracy). |
| Wang and Zhao (2021) | Multiple datasets from UCR Time Series Classification Archive | Proposed Amerced Dynamic Time Warping (ADTW) for improved and more balanced time-series classification that penalises extra warping steps. | Accuracy | Increased accuracy by an average of 1% to 5%. Produced more intuitive alignments. |
| End of Table. | | | | |

# Appendix C Dataset Statistical Types and Data Types

Table C.1: Attributes of the long-column formatted CT dataset.

| **Attribute** | **Description** | **Statistical Type** | **Data Type** |
|---|---|---|---|
| Mins | The time at which the data point was recorded in minutes. | Ratio | Decimal |
| Seconds | The time at which the data point was recorded in seconds. | Ratio | Decimal |
| Value | The value of the data point. | Ratio | Decimal |
| Replicate | The number representing which replicate of a specific sample the data point belongs to. | Nominal | Integer |
| Sample ID | The unique sample that the data point belongs to. | Nominal | String |

Table C.2: Attributes of the additional information for the CT dataset.

| **Attribute** | **Description** | **Statistical Type** | **Data Type** |
|---|---|---|---|
| Sample ID | The unique sample that the data point belongs to. | Nominal | String |
| Sample Type | The region from where the sample originated from (refer to Section 3.2). | Nominal | String |
| Result | The outcome of the sample, namely whether it tested positive or negative. | Nominal | String |
| Gender | The biological gender of the patient, from which the sample originated. | Nominal | String |
| Duplicate | A denotation of whether the sample is a duplicate (refer to Section 3.2). | Nominal | String |

Table C.3: Attributes of the long-column formatted COVID-19 dataset.

| **Attribute** | **Description** | **Statistical Type** | **Data Type** |
|---|---|---|---|
| Mins | The time at which the data point was recorded in minutes. | Ratio | Integer |
| Sample ID | The unique sample that the data point belongs to. | Nominal | String |
| Plate | The plate to which the sample was tested on. Can be considered negligible. | Nominal | String |
| Result | The outcome of the sample, namely whether it tested positive or negative. | Nominal | String |
| Replicate | The number representing which replicate of a specific sample the data point belongs to. | Nominal | Integer |
| Value | The value of the data point. | Ratio | Decimal |

Table C.4: Attributes of the long-column formatted PNAS dataset.

| **Attribute** | **Description** | **Statistical Type** | **Data Type** |
|---|---|---|---|
| Mins | The time at which the data point was recorded in minutes. | Ratio | Decimal |
| Sample ID | The unique sample that the data point belongs to. | Nominal | String |
| Value | The value of the data point. | Ratio | Decimal |
| Concentration | The specified concentration of the COVID-19 RNA added to the reaction. | Ratio | Decimal |
| Replicate | The number representing which replicate of a specific sample the data point belongs to. | Nominal | Integer |

# Appendix D Extra Exploratory Data Analysis

## D.1 Dataset 1: Chlamydia

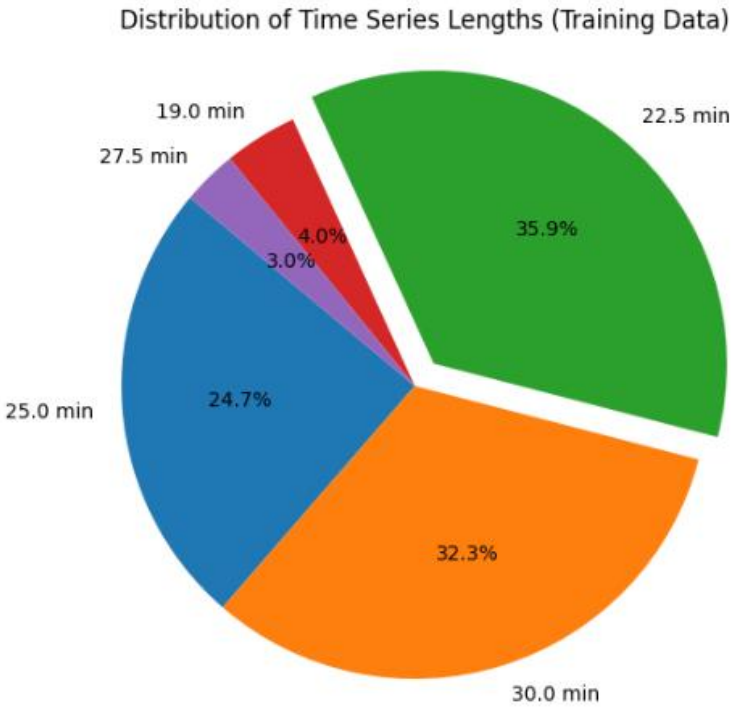


Figure D.1: Distribution of varying Chlamydia time-series lengths.

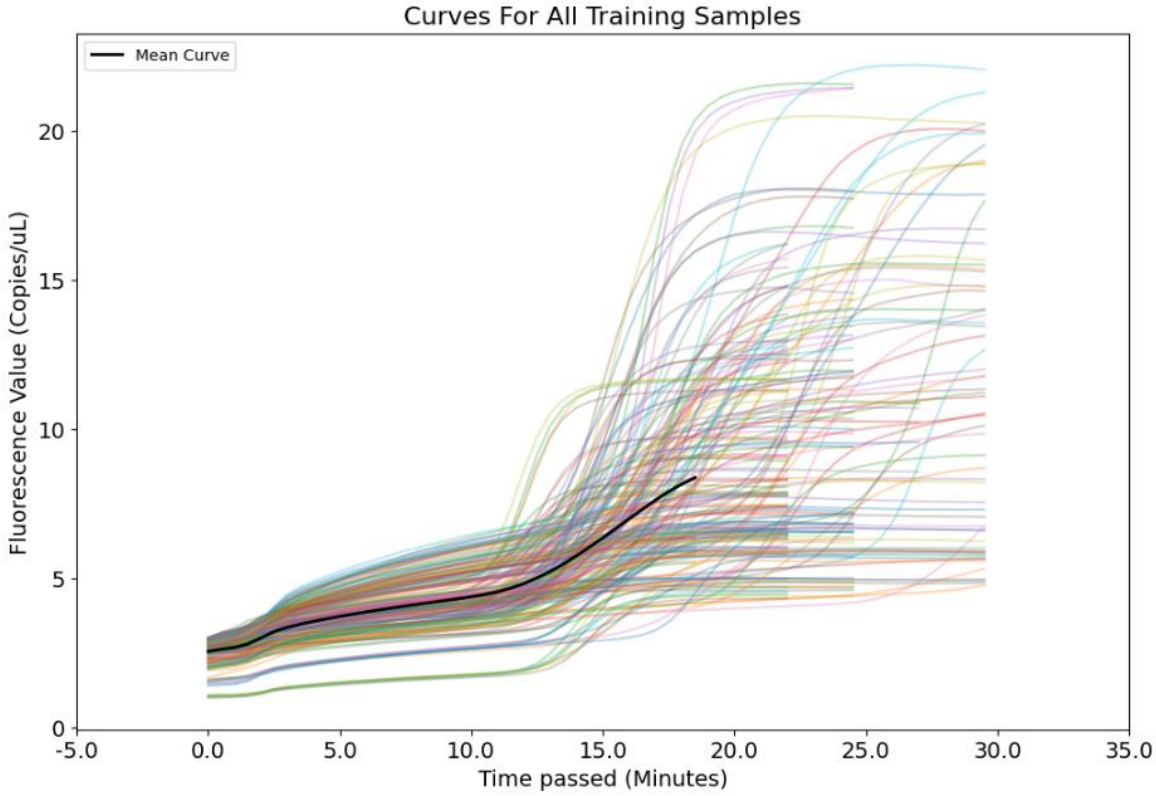


Figure D.2: Chlyamydia curves without data scaling.

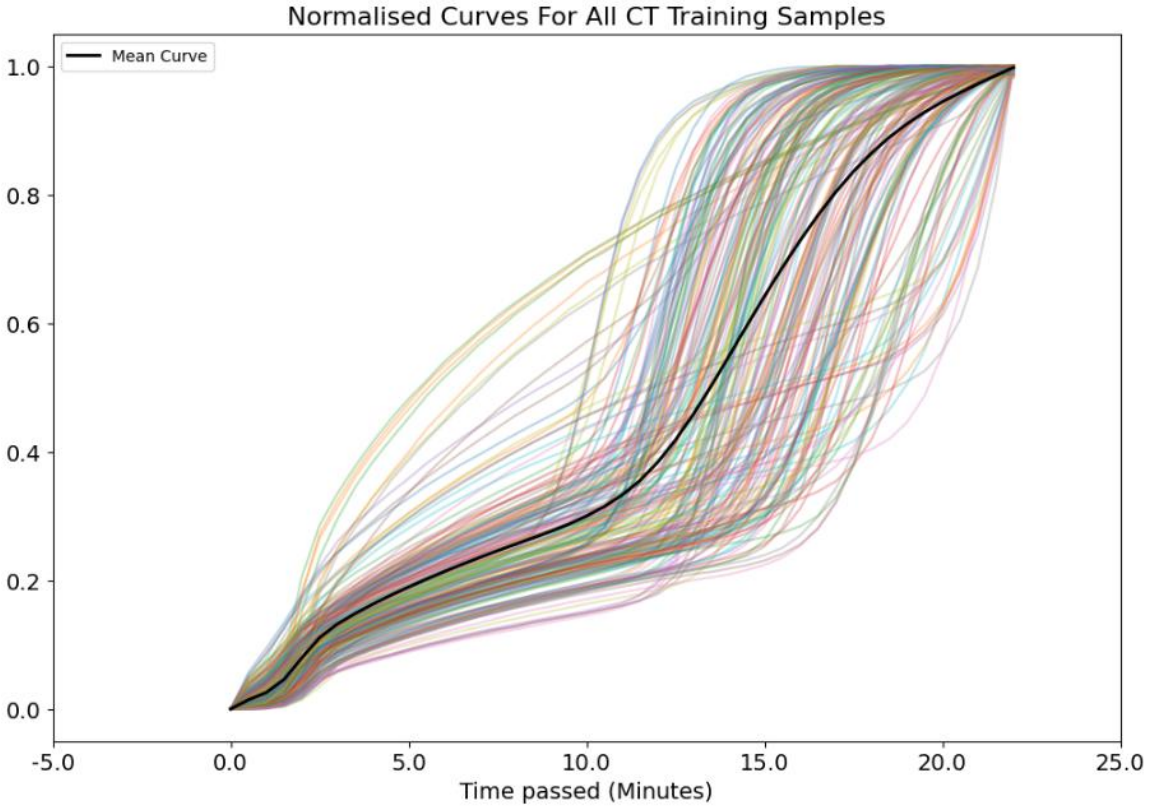


Figure D.3: Normalised Chlamydia curves (without outlier removal).

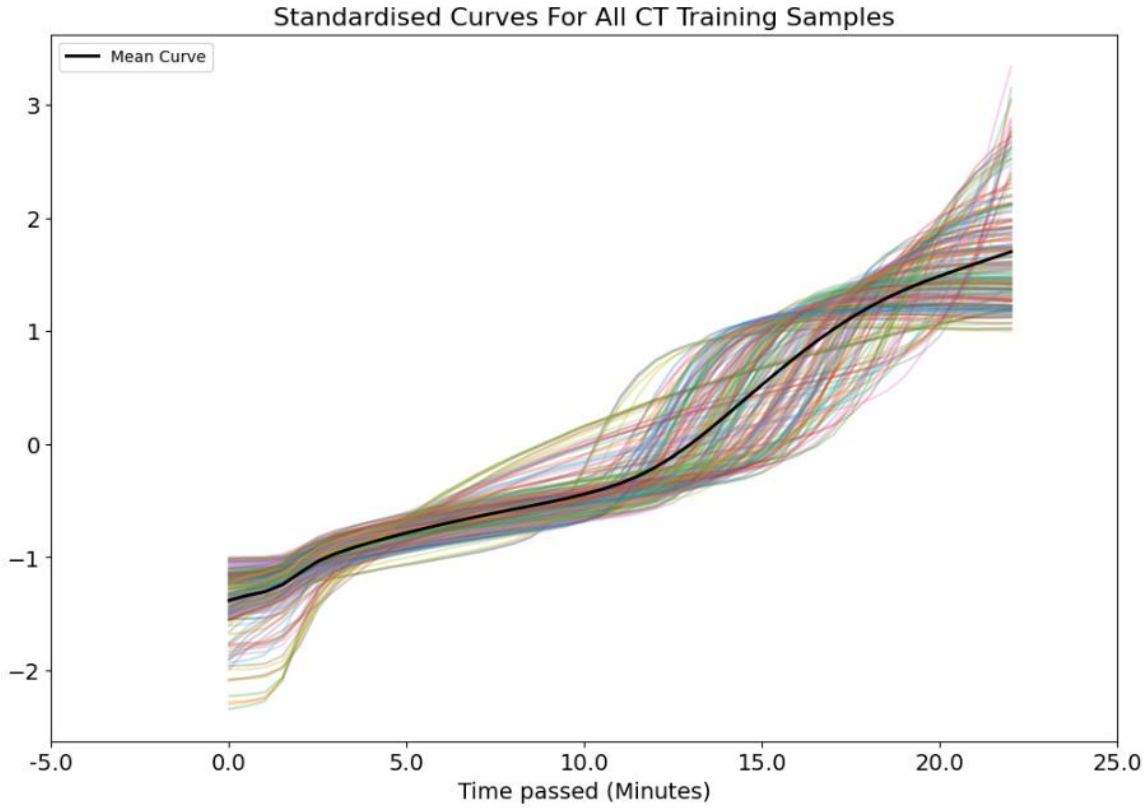


Figure D.4: Standardised Chlamydia curves (without outlier removal).

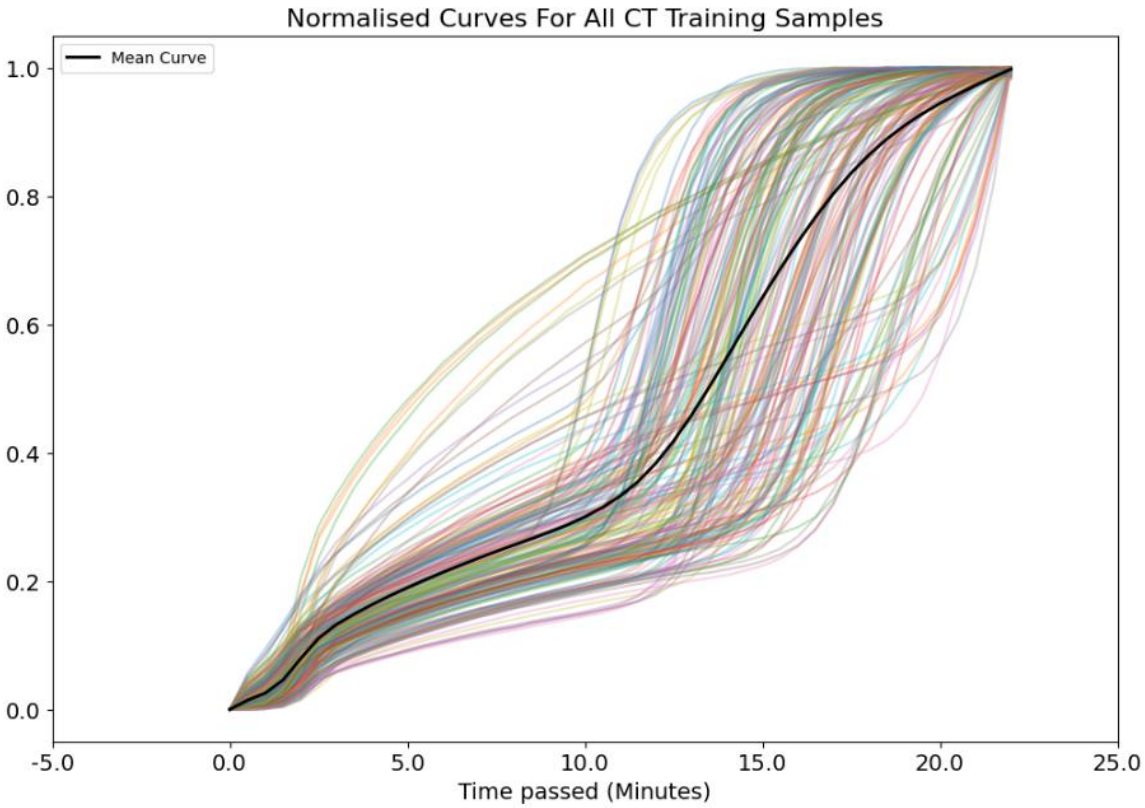


Figure D.5: Normalised Chlamydia curves (with outlier removal).

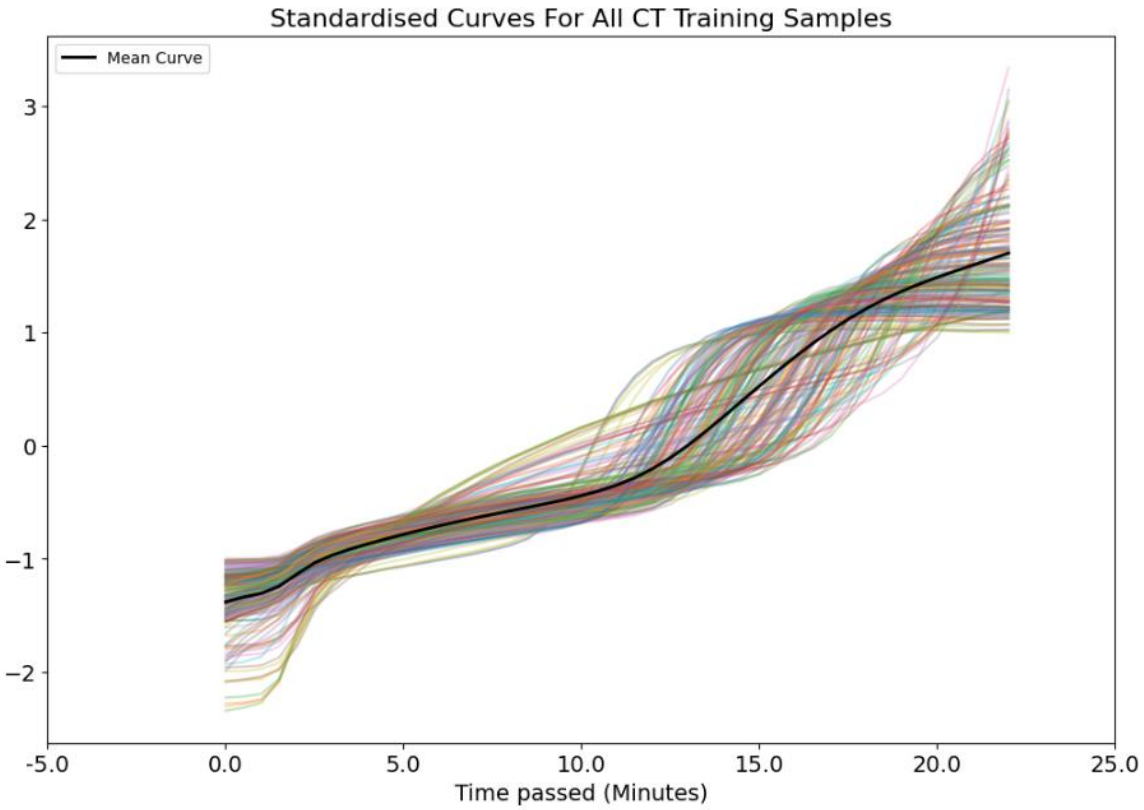


Figure D.6: Standardised Chlamydia curves (with outlier removal).

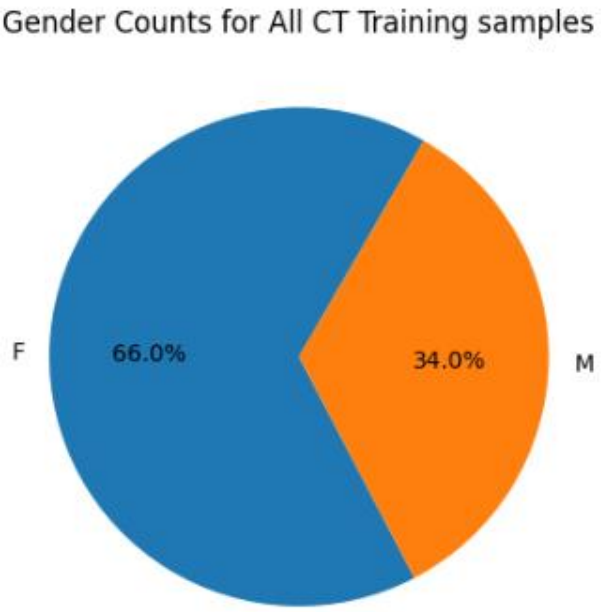


Figure D.7: Chlamydia gender distribution.

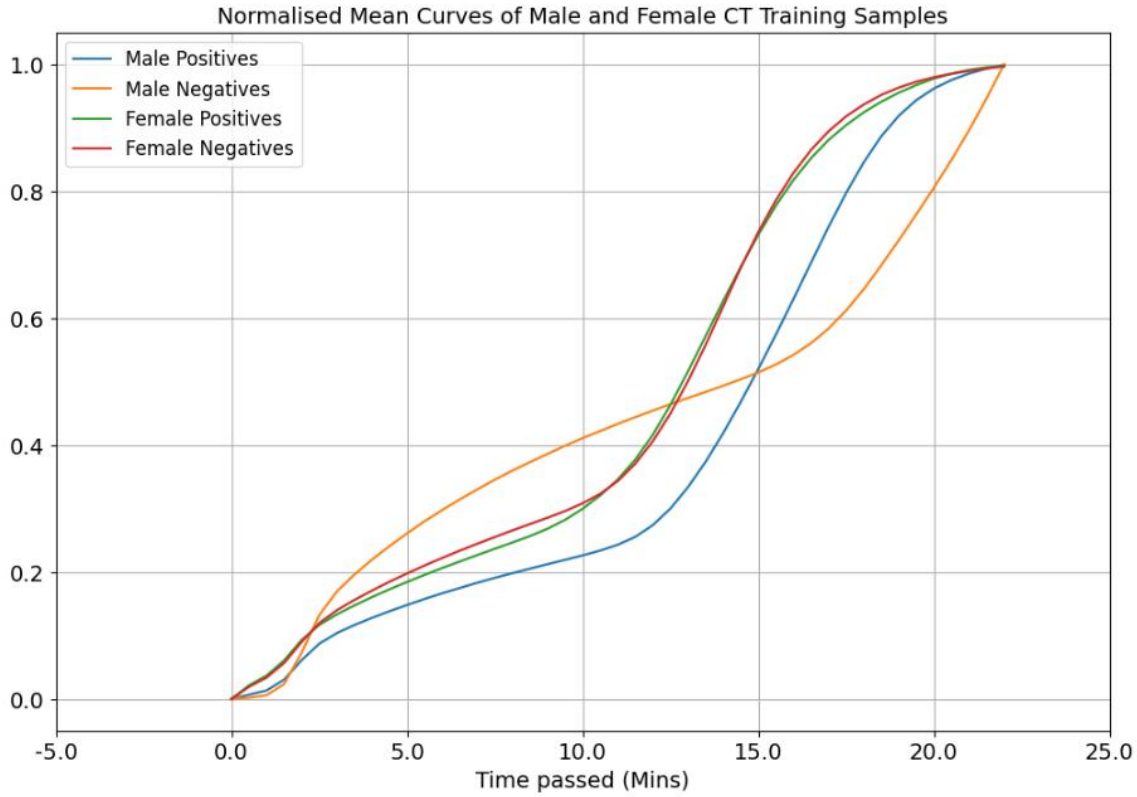


Figure D.8: Chlamydia normalised mean curves by gender.

## D.2 Dataset 2: COVID-19

Figure D.9 shows the COVID-19 curves without data scaling, which shows how scale can distort trends and patterns. Figure D.14 displays the distribution of positive cases versus negative cases for the COVID-19 dataset. It is clear there are far more negative cases compared to positive ones. Furthermore, Figure D.15 shows the mean curves for the positive samples and negative samples (for the training set). Finally, Figure D.16 illustrates the dynamic time warping region between the mean positive samples curve and mean negative samples curve.

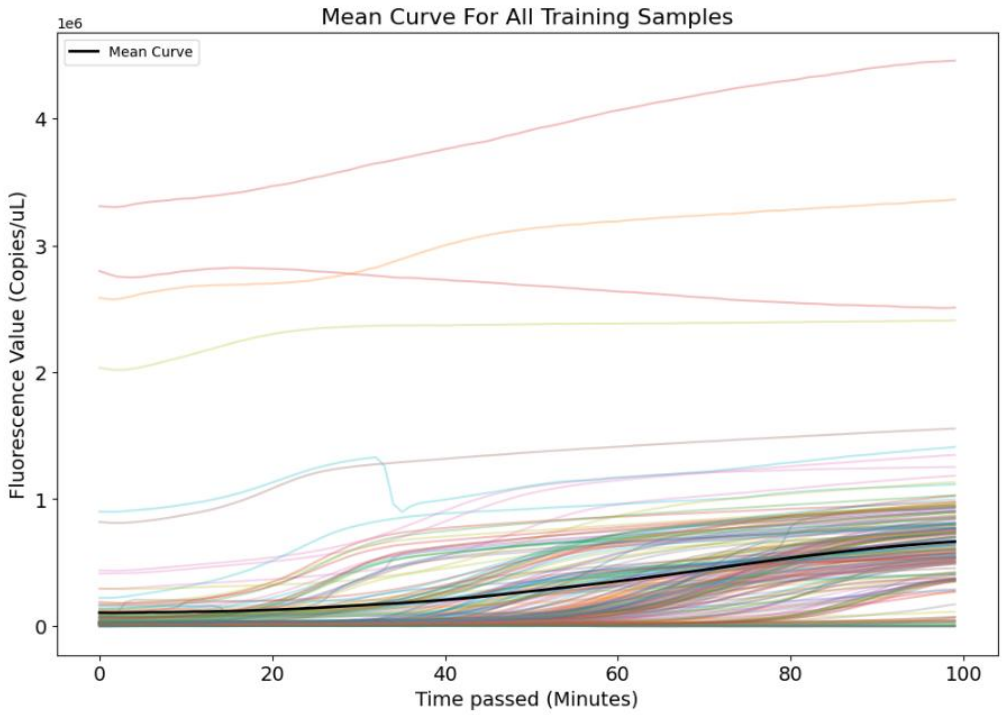


Figure D.9: COVID-19 curves without data scaling.

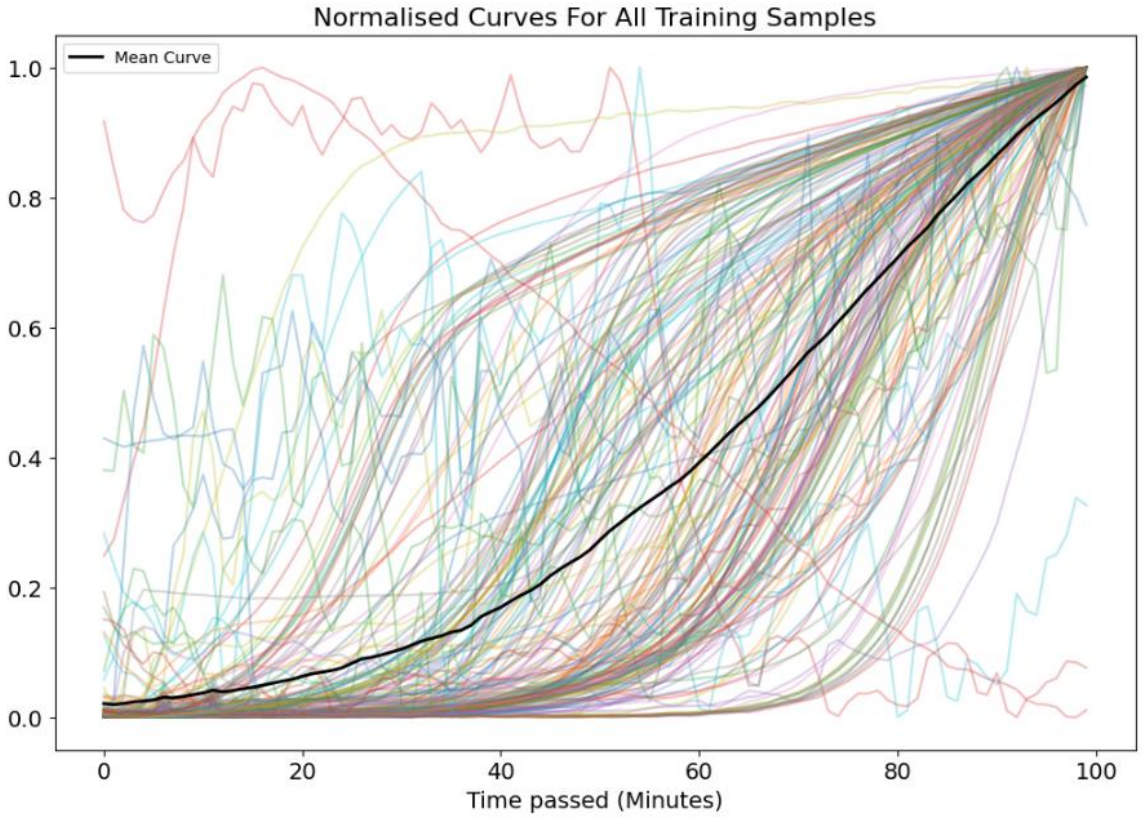


Figure D.10: Normalised COVID-19 curves (without outlier removal).

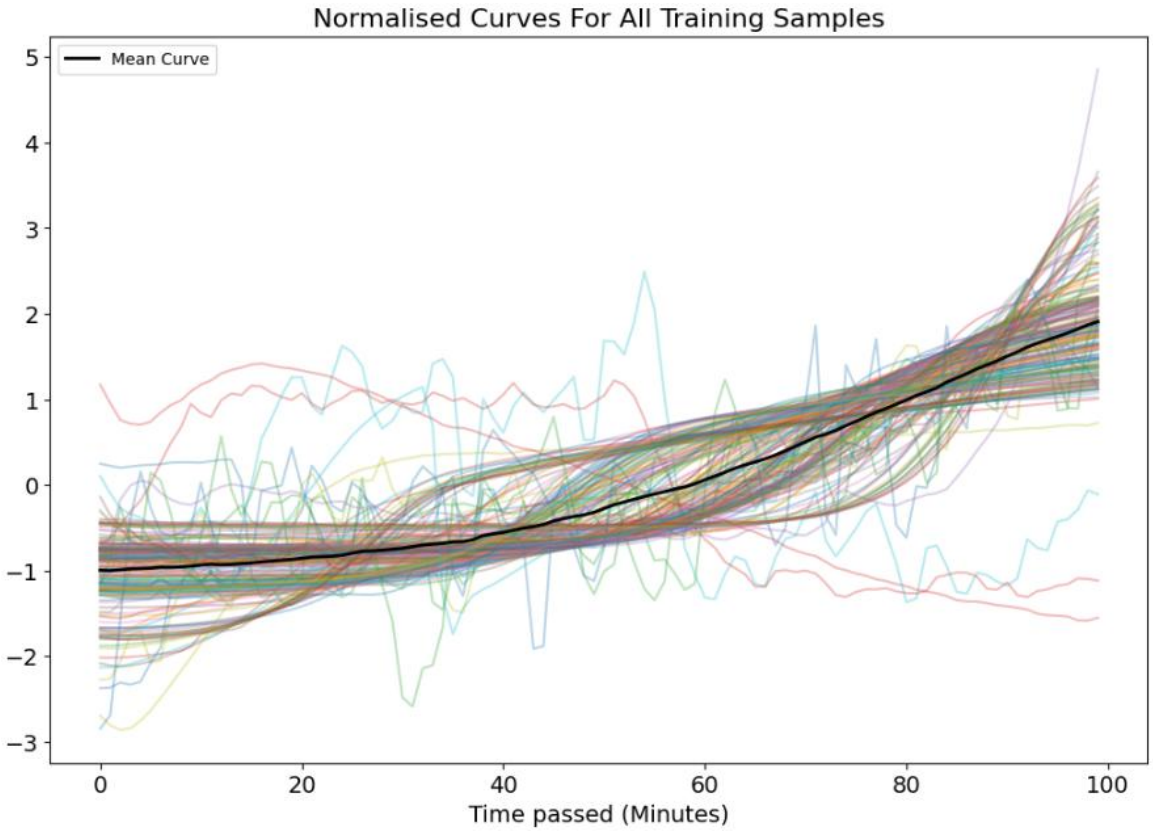


Figure D.11: Standardised COVID-19 curves (without outlier removal).

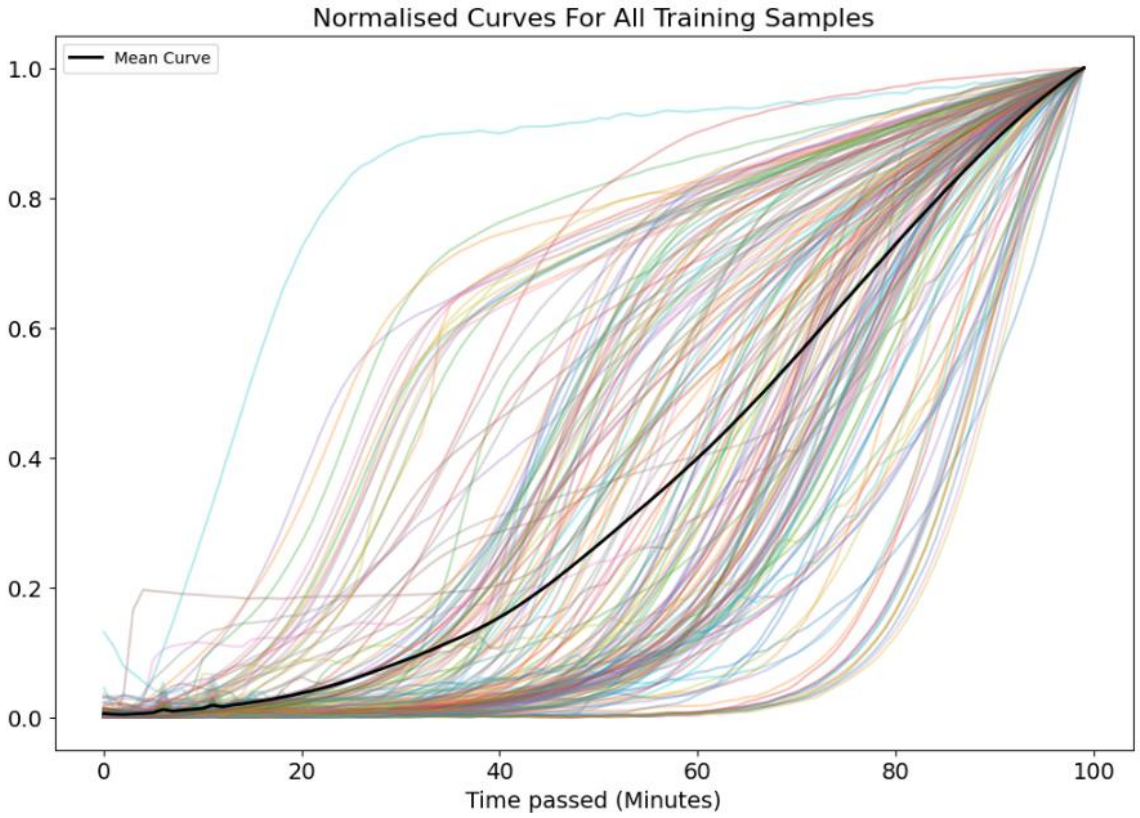


Figure D.12: Normalised COVID-19 curves (with outlier removal).

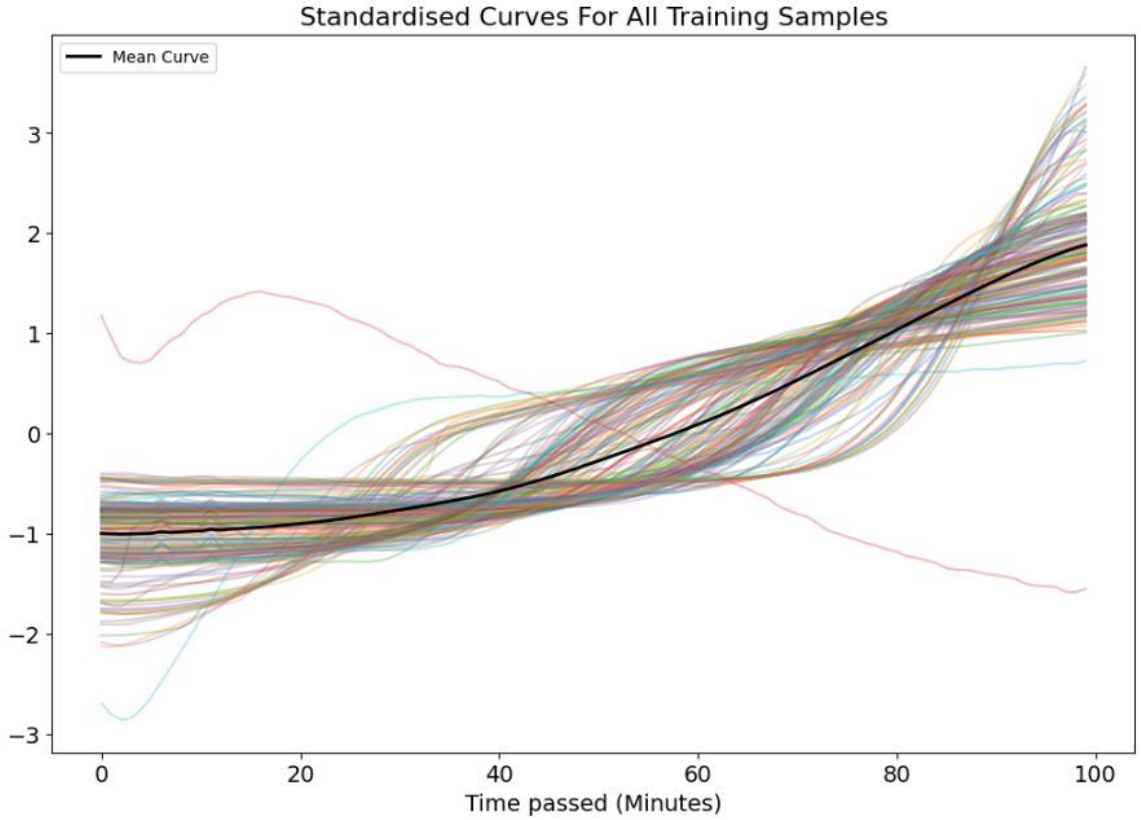


Figure D.13: Standardised COVID-19 curves (with outlier removal).

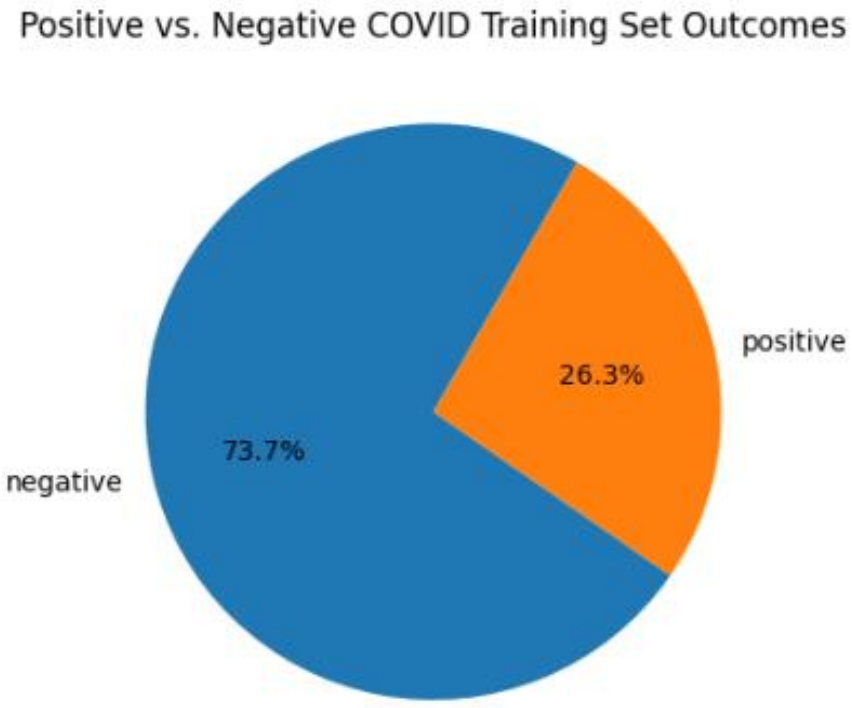


Figure D.14: COVID-19 pie chart of the sample outcome distribution.

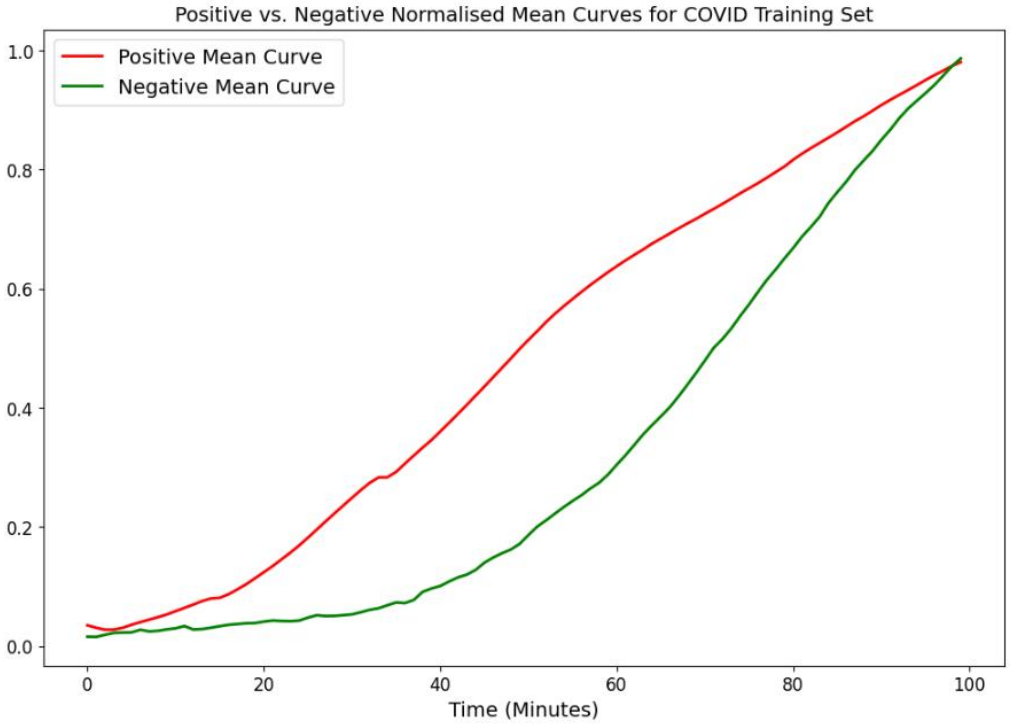


Figure D.15: COVID-19 mean curves for positive and negative samples.

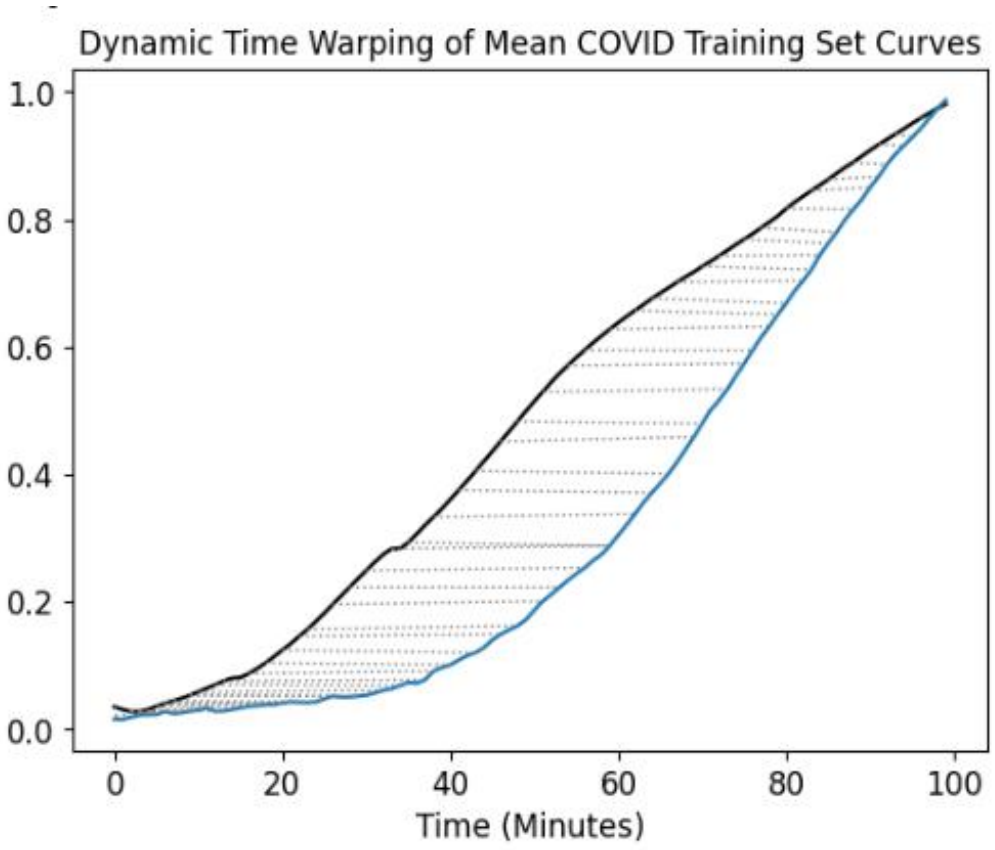


Figure D.16: COVID-19 dynamic time warping region between the mean curves for positive and negative samples.

## D.3 Dataset 3: PNAS

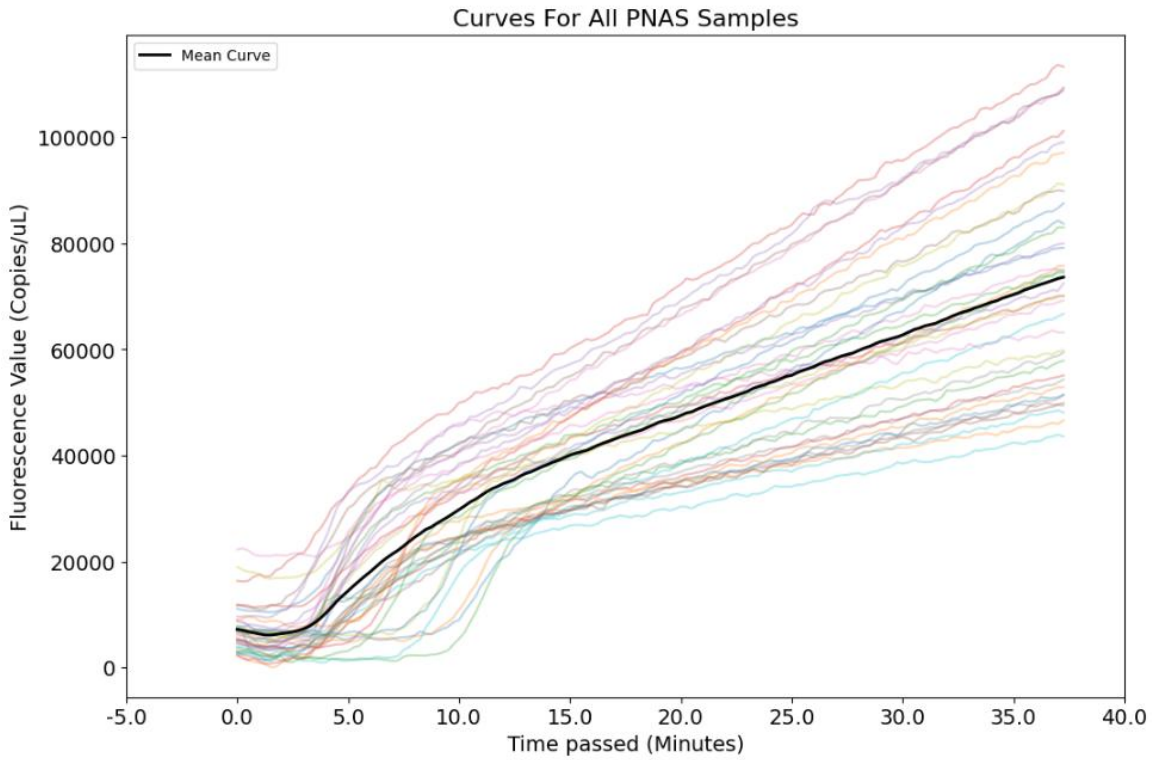


Figure D.17: PNAS curves without data scaling.

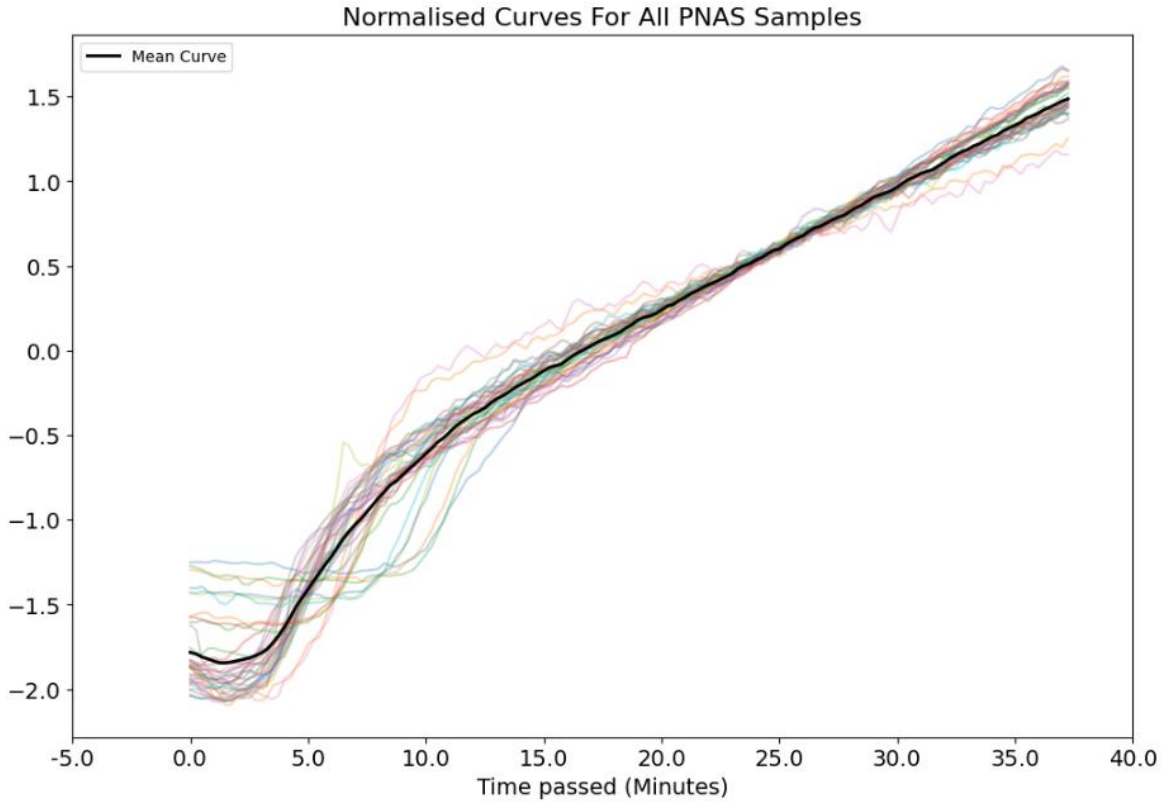


Figure D.18: Normalised PNAS curves.

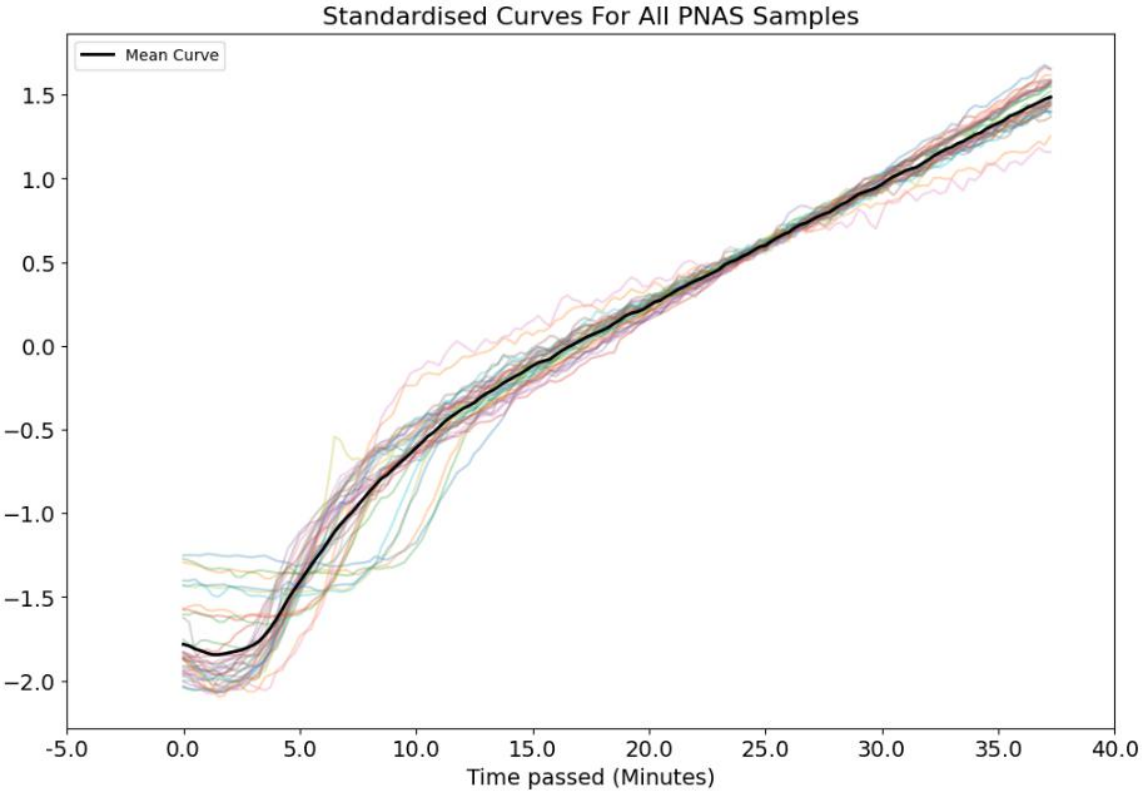


Figure D.19: Standardised PNAS curves.

# Appendix E Extra Model Results

## E.1 DTW and K-Nearest Neighbour

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.93 | 0.90 | 45 |
| 1 | 0.94 | 0.88 | 0.91 | 51 |
| accuracy | | | 0.91 | 96 |
| macro avg | 0.91 | 0.91 | 0.91 | 96 |
| weighted avg | 0.91 | 0.91 | 0.91 | 96 |

Figure E.1: Optimised results for utilising DTW and k-nearest neighbour on the COVID-19 training data.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.85 | 0.87 | 0.86 | 82 |
| 1 | 0.87 | 0.85 | 0.86 | 86 |
| accuracy | | | 0.86 | 168 |
| macro avg | 0.86 | 0.86 | 0.86 | 168 |
| weighted avg | 0.86 | 0.86 | 0.86 | 168 |

Figure E.2: Optimised results for utilising DTW and k-nearest neighbour on the chlamydia training data.

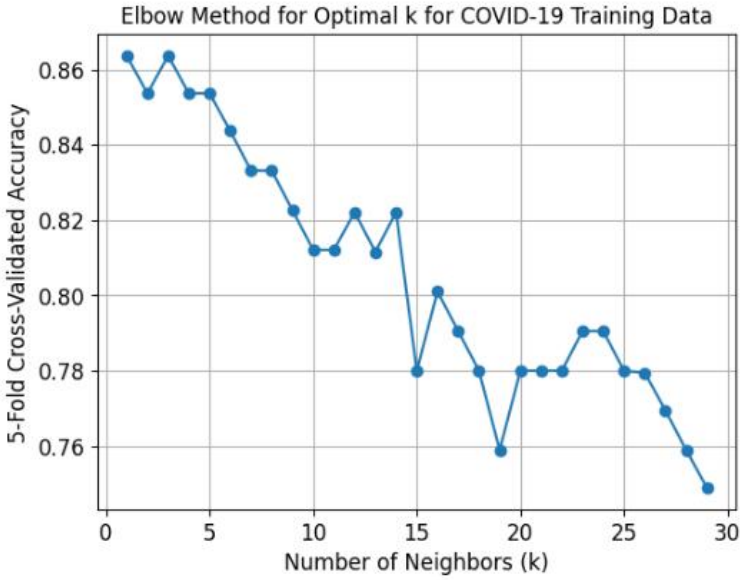


Figure E.3: Applying the Elbow method for optimising 'k' for the COVID-19 training data.

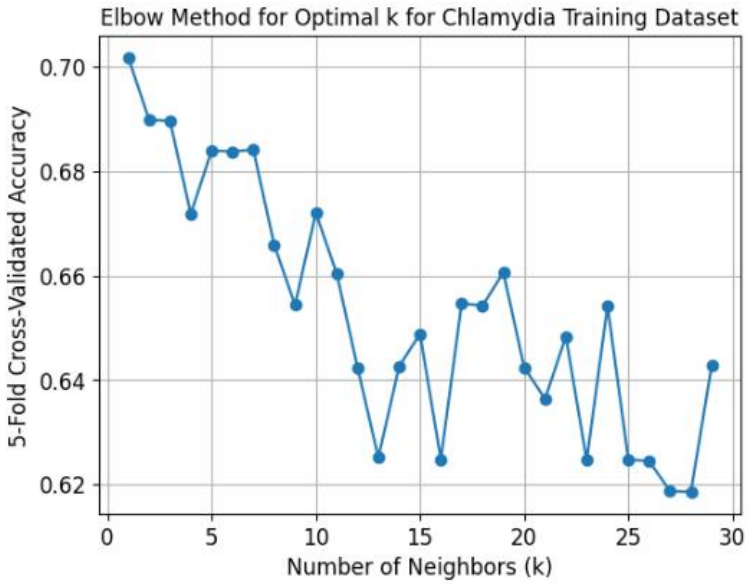


Figure E.4: Applying the Elbow method for optimising 'k' for the chlamydia training data.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 7 |
| 1 | 1.00 | 1.00 | 1.00 | 7 |
| accuracy | | | 1.00 | 14 |
| macro avg | 1.00 | 1.00 | 1.00 | 14 |
| weighted avg | 1.00 | 1.00 | 1.00 | 14 |

Figure E.5: Optimised results for utilising DTW and k-nearest neighbour on the COVID-19 validation data.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.67 | 0.67 | 0.67 | 12 |
| 1 | 0.69 | 0.69 | 0.69 | 13 |
| accuracy | | | 0.68 | 25 |
| macro avg | 0.68 | 0.68 | 0.68 | 25 |
| weighted avg | 0.68 | 0.68 | 0.68 | 25 |

Figure E.6: Optimised results for utilising DTW and k-nearest neighbour on the chlamydia validation data.

## E.2 Correlation Analysis

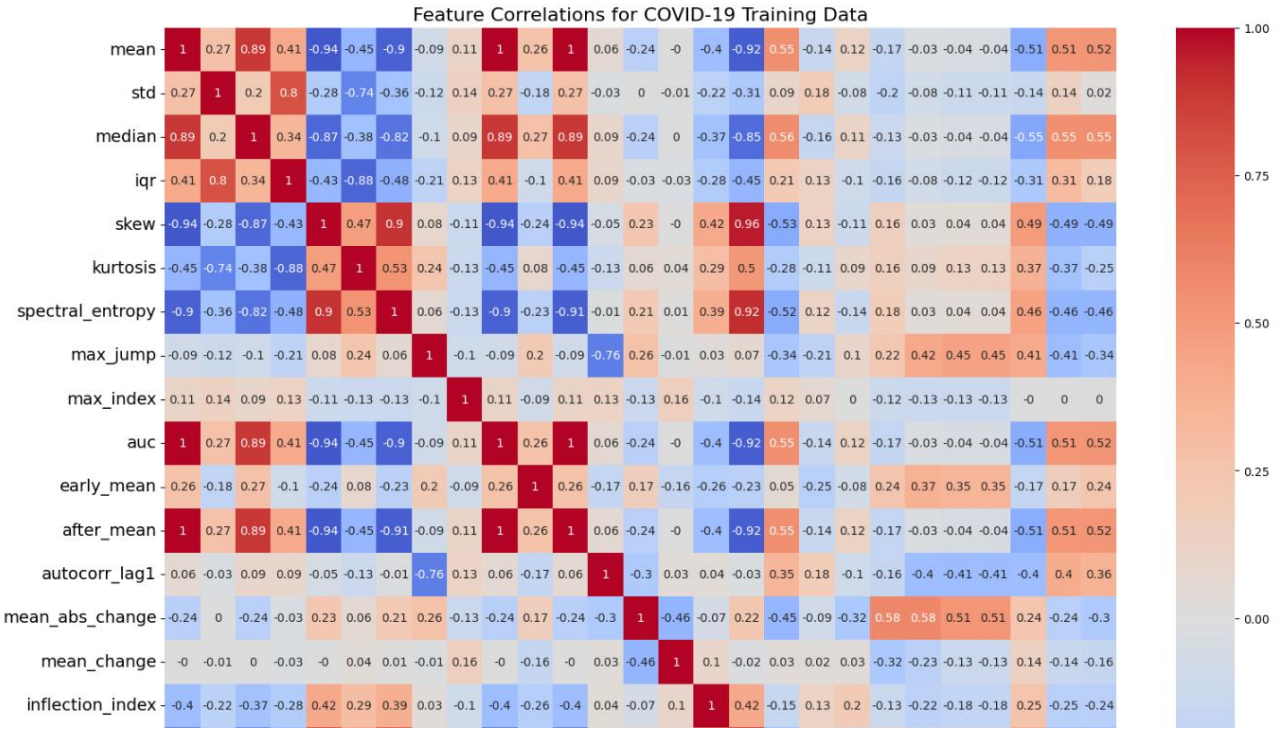


Figure E.7: Kendall Tau correlation analysis for the COVID-19 training set. part 1.

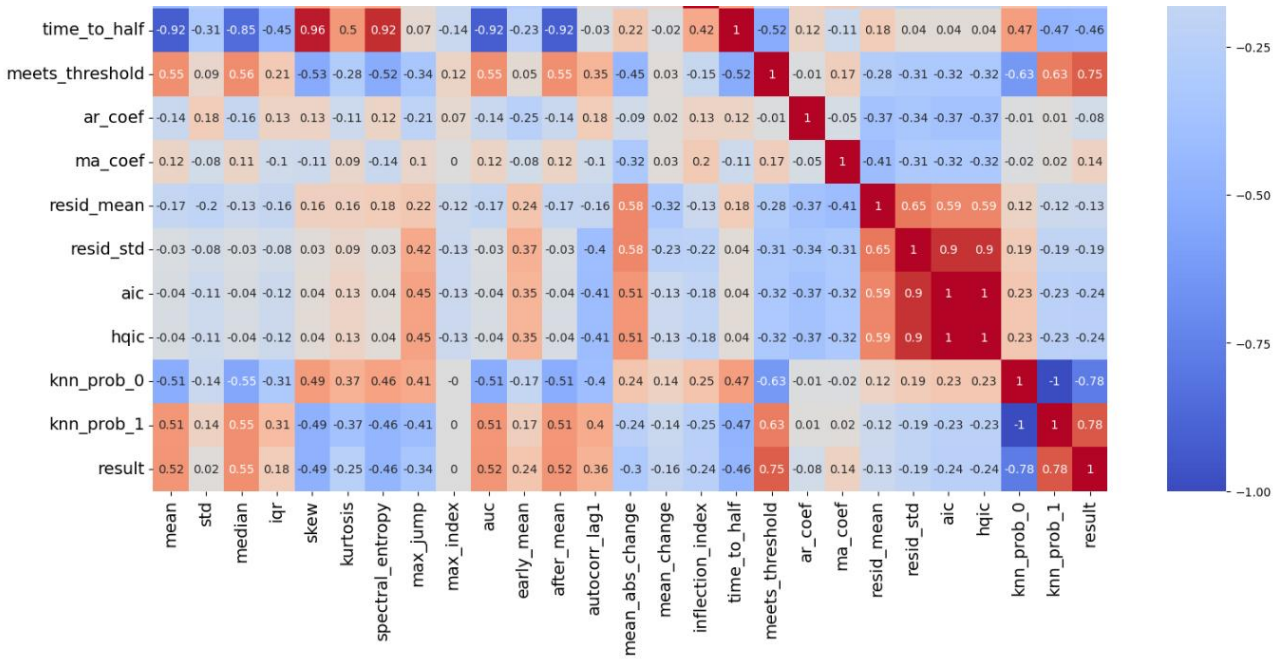


Figure E.8: Kendall Tau correlation analysis for the COVID-19 training set, part 2.

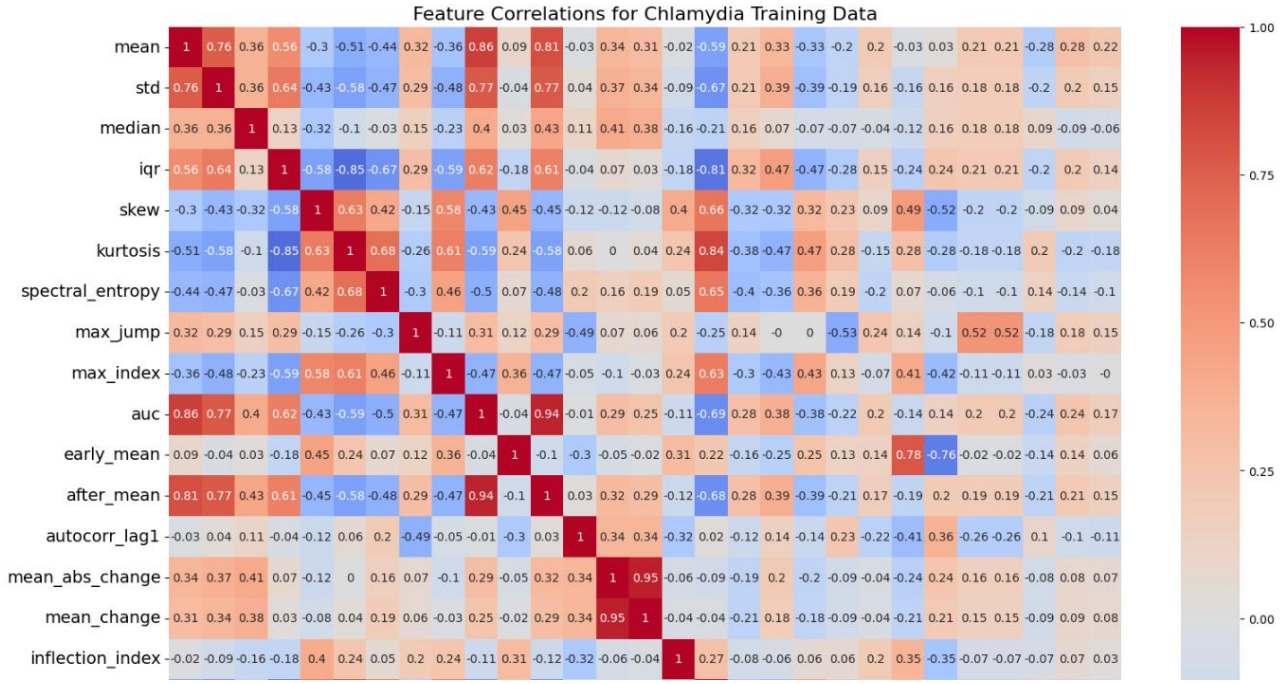


Figure E.9: Kendall Tau correlation analysis for the chlamydia training set, part 1.

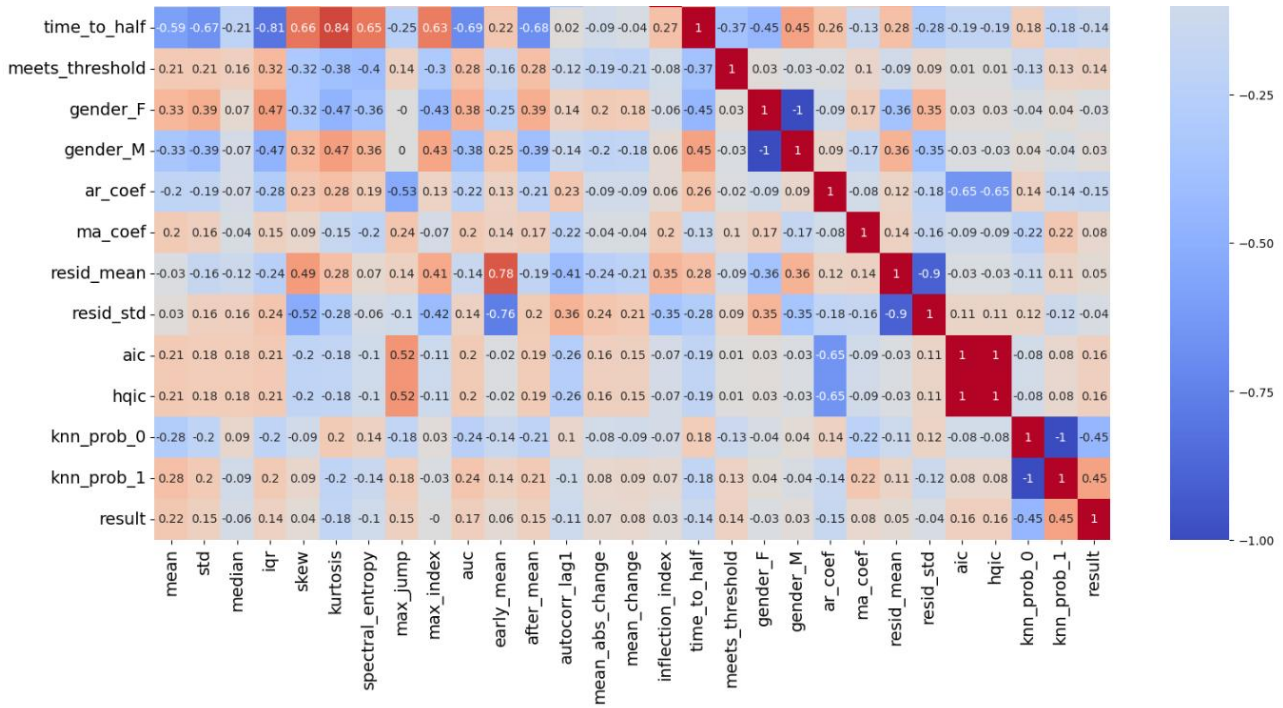


Figure E.10: Kendall Tau correlation analysis for the chlamydia training set, part 2.

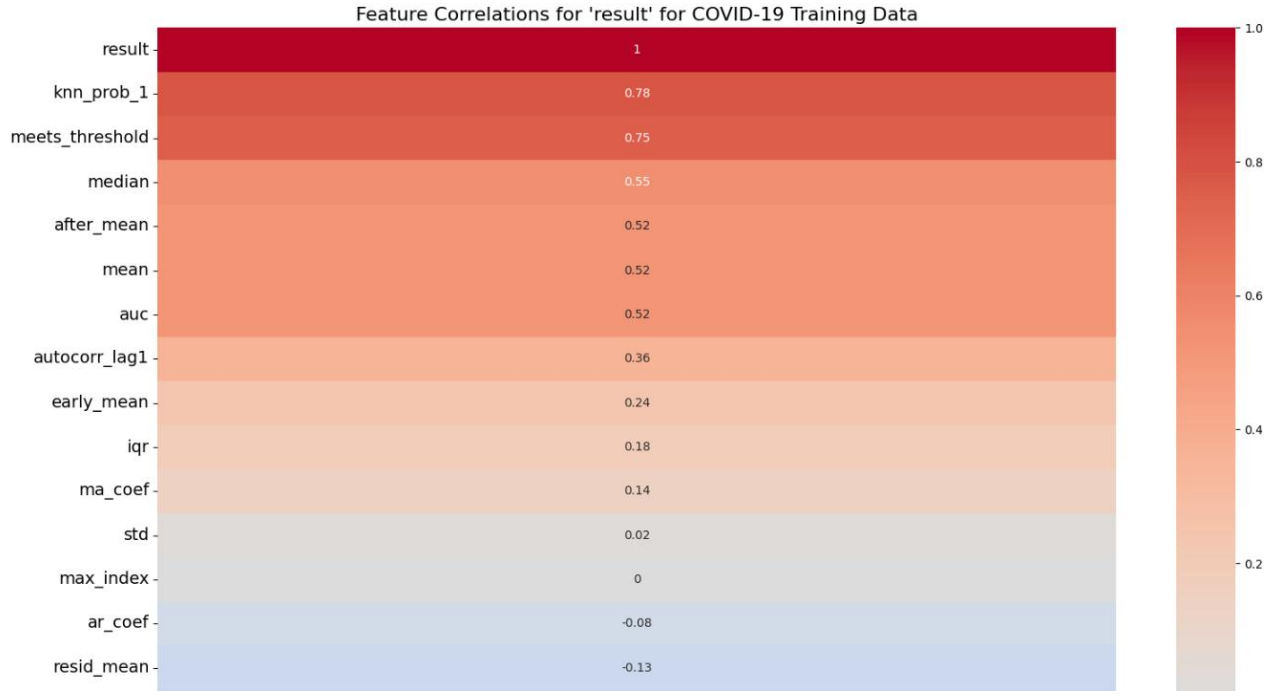


Figure E.11: Kendall Tau correlation analysis for 'result' for the COVID-19 training set. part 1.

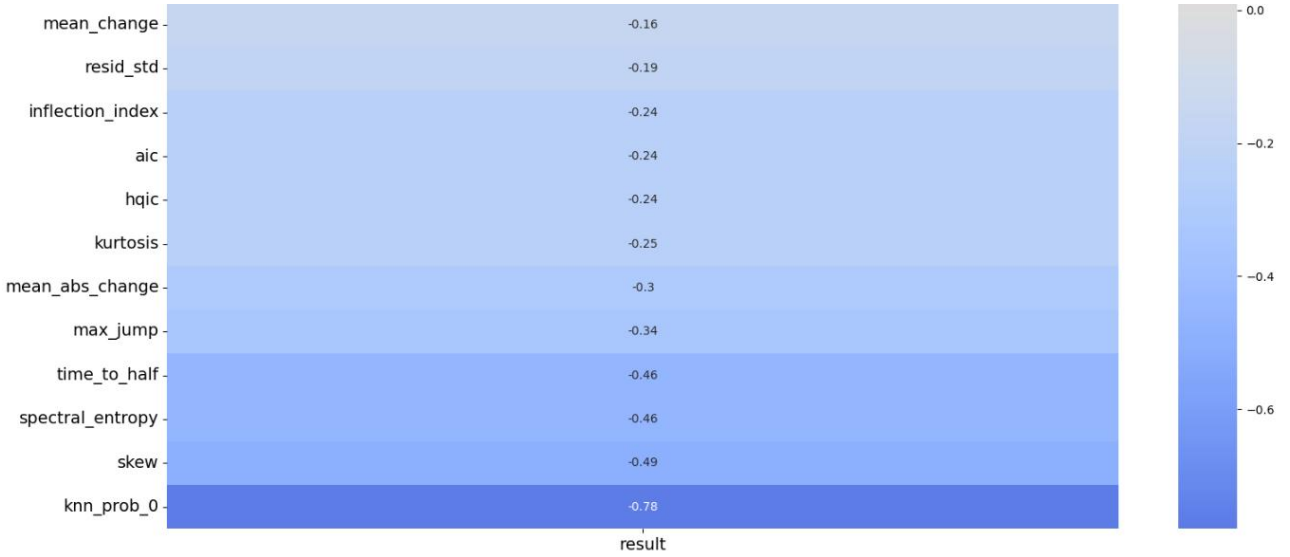


Figure E.12: Kendall Tau correlation analysis for 'result' for the COVID-19 training set, part 2.

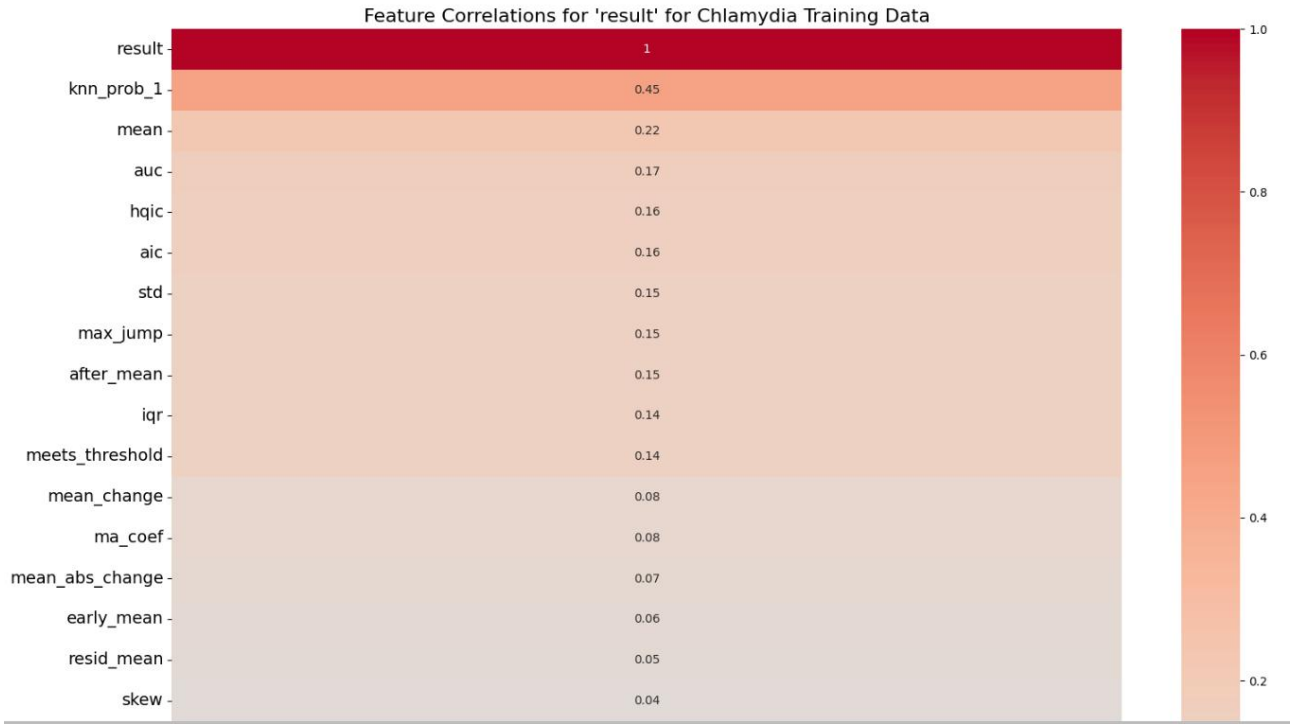


Figure E.13: Kendall Tau correlation analysis for 'result' for the chlamydia training set, part 1.

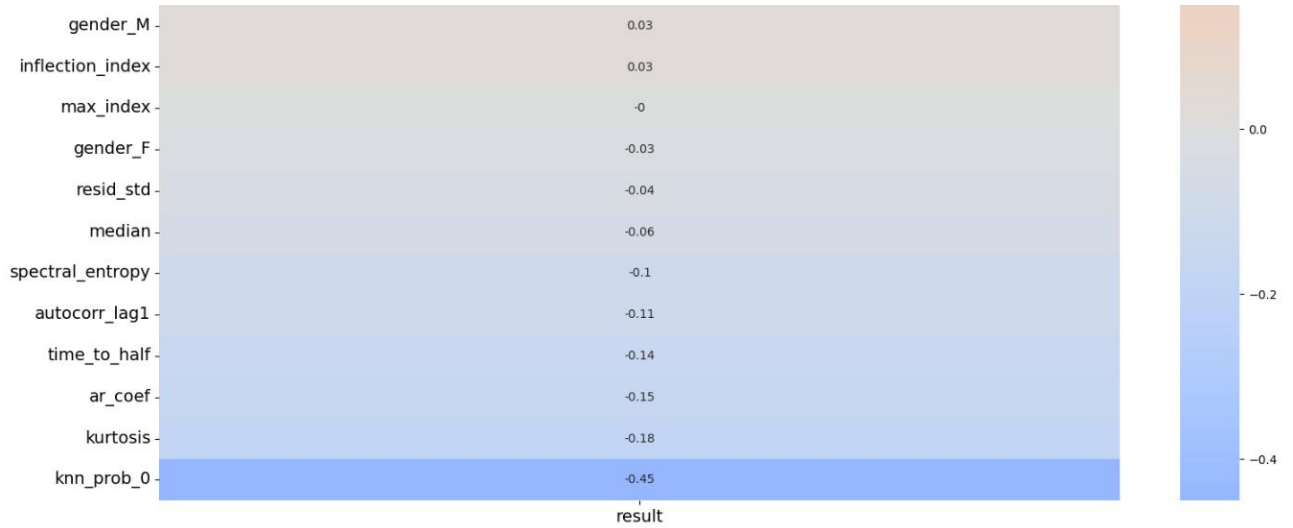


Figure E.14: Kendall Tau correlation analysis for 'result' for the chlamydia training set, part 2.

## E.3 Model Features

Table E.1: All attributes of the final training data before model development.

| **Attribute** | **Description** |
|---|---|
| Mean | The mean value of each time series. |
| Median | The median value of each time series. |
| Standard deviation | The standard deviation of each time series. |
| Interquartile range | The interquartile range of each time series |
| Skew | A statistic representing the asymmetry of the distribution of data points for each time series. |
| Kurtosis | A statistic representing the tailedness of the distribution of data points for each time series. |
| Spectral entropy | A measure of the complexity or disorder in each time series. |
| Max jump | The single biggest difference between any two adjacent data points within each time series. |
| Max index | The maximum index the time series has reached. This is not important if all the time series lengths are the same. |
| Area under curve | The total area underneath each time series |
| Early mean | The mean value of a set of data points before a specific time index, for each time series. |
| After mean | The mean value of a set of data points after and including a specific time index, for each time series. |
| Lag-1 autocorrelation | A measure of how strongly a time series is correlated with itself one step earlier. This measures how "erratic" the time series progresses. |
| Mean change | The after mean minus the early mean. |
| Mean absolute change | The absolute difference between the early mean and after mean. |
| Inflection index | The index of when the curve stops curving in one direction and starts curving in the other. |
| Max growth | The single biggest difference between any two adjacent data points within each time series. |

| Continuation of Table E.1 | |
|---|---|
| Time to half | The index at the middle of each time series. |
| DSTT (meets) threshold | The classification prediction by DSTT whether the current time series is positive or negative. |
| Gender | The gender of the patient. |
| Autoregressive coefficient | A measure, when using ARIMA, of how much previous values influence the current value, for each time series. |
| Moving average coefficient | A measure, when using ARIMA, of how much past errors (residuals) influence the current value, for each time series. |
| Residual mean | The mean average prediction error when applying ARIMA to each time series. |
| Residual standard deviation | The standard deviation of the prediction error when applying ARIMA to each time series. |
| Akaike Information Criterion (AIC) | A model selection criterion that measures how well ARIMA has adapted to the current time series. |
| Hannan–Quinn Information Criterion (HQIC) | This operates similar to AIC, but penalises more complexity in the ARIMA model. |
| DTW and k-nearest neighbour predictions | The classification prediction by the DTW and k-nearest neighbour method whether the current time series is positive or negative. |
| Result | The ground truth — the actual class the specific time series belongs to. |
| End of Table | |

## E.4 Hyperparameter Tuning

Table E.2: Hyperparameters to be tuned for decision tree.

| **Hyperparameter** | **Description** | **Setting** |
|---|---|---|
| Max depth | The maximum number of levels the tree can grow. | 3, 5, 7, 10, None |
| Minimum samples split | The minimum number of samples required to split an internel node. | 2, 5, 10 |
| Minimum samples leaf | The minimum number of samples required to be in a leaf node. | 1, 2, 4 |
| Criterion | The metric used to evaluate the quality of a split in the tree. | Gini or Entropy loss |

Table E.3: Hyperparameters to be tuned for random forest.

| **Hyperparameter** | **Description** | **Setting** |
|---|---|---|
| Number of estimators | The number of decision trees in the ensemble, which vote and declare their prediction for the sample. | 50, 100, 200 |
| Max depth | The maximum number of levels the tree can grow. | 3, 5, 7, 10, None |
| Minimum samples split | The minimum number of samples required to split an internel node. | 2, 5, 10 |
| Minimum samples leaf | The minimum number of samples required to be in a leaf node. | 1, 2, 4 |
| Max features | How many features each tree is allowed to consider when making a split. | Square root, log base 2, 0.5, 0.8 |

Table E.4: Hyperparameters to be tuned for XGBoost.

| **Hyperparameter** | **Description** | **Setting** |
|---|---|---|
| Number of estimators | The number of decision trees in the ensemble, which vote and declare their prediction for the sample. | 50, 100, 200 |
| Max depth | The maximum number of levels the tree can grow. | 3, 5, 7, 10, None |
| Learning rate | Controls how much each new tree contributes to the overall model. | 0.01, 0.05, 0.1, 0.2 |
| Subsampling | Controls how much of the training data is randomly sampled for each boosting round. | 0.6, 0.8, 1.0 |
| Columns to sample by tree | Controls how many features are randomly chosen for each tree in the ensemble. | 0.6, 0.8, 1.0 |
| Gamma | A regularization hyperparameter that controls minimum loss reduction required to make a split. | 0, 1, 5 |
| Minimum child weight | A regularization hyperparameter that controls the minimum sum of instance weights (or sample count) required to create a new child node. | 1, 3, 5 |

Table E.5: Hyperparameters to be tuned for k-nearest neighbour.

| **Hyperparameter** | **Description** | **Setting** |
|---|---|---|
| Number of neighbours | How many neighbours to consider when assigning a majority label between them. Also known as the 'k' value. | 1 to 21 |
| Weights | Determines how much each neighbour has when making a prediction. | Uniform (all equal voting power), distanced-based (where closer ones are more important) |
| Metric | The function to use to calculate distances between points. | Euclidean, Manhattan, and Minkowski |
| 'p' | Controls the shape of the Minkowski distance metric. | 1, 2 |

Table E.6: Hyperparameters to be tuned for ridge classifier.

| **Hyperparameter** | **Description** | **Setting** |
|---|---|---|
| Alpha | Controls the strength of the L2 regularisation. | 0.01, 0.1, 1, 10, 100 |

Table E.7: Hyperparameters to be tuned for logistic regression.

| **Hyperparameter** | **Description** | **Setting** |
|---|---|---|
| 'C' | Controls the regularisation strength. | 0.01, 0.1, 1, 10, 100 |
| Penalty | The type of regulariser to apply. | L2 regularisation |
| Solver | Determines how the model is trained. | 'liblinear, 'saga' |
| Class weight | Adjusts the importance of each class during training. | None, balanced |

Table E.8: Hyperparameters to be tuned for support vector classifier.

| **Hyperparameter** | **Description** | **Setting** |
|---|---|---|
| 'C' | Controls the regularisation strength. | 0.01, 0.1, 1, 10, 100 |
| Kernel | Defines how input data is transformed into a higher-dimensional space that is linearly separable. | Linear, radial basis function |
| Gamma | Determines the reach of each training point's influence in the transformed feature space. | 'scale', 'auto' |
| Class weight | Adjusts the importance of each class during training. | None, balanced |

## E.5 Model Results for Feature Selection

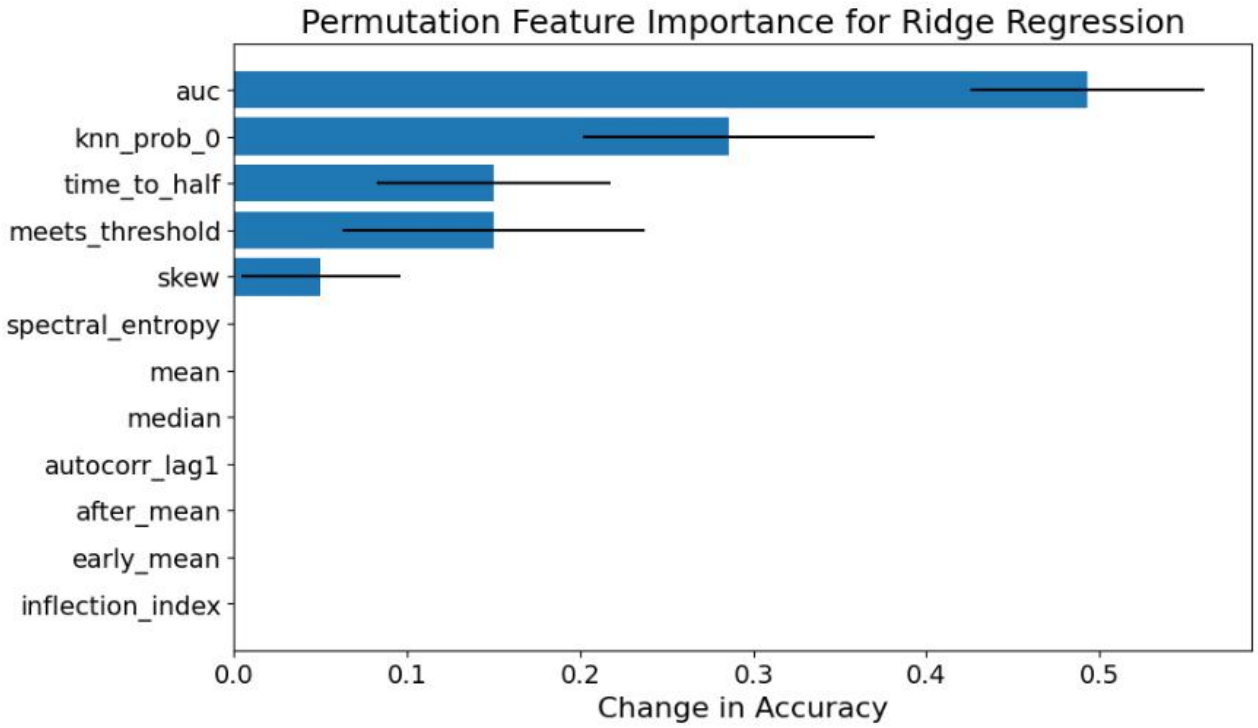


Figure E.15: Permutation importance for ridge classifier for initial feature inputs for the COVID-19 dataset.

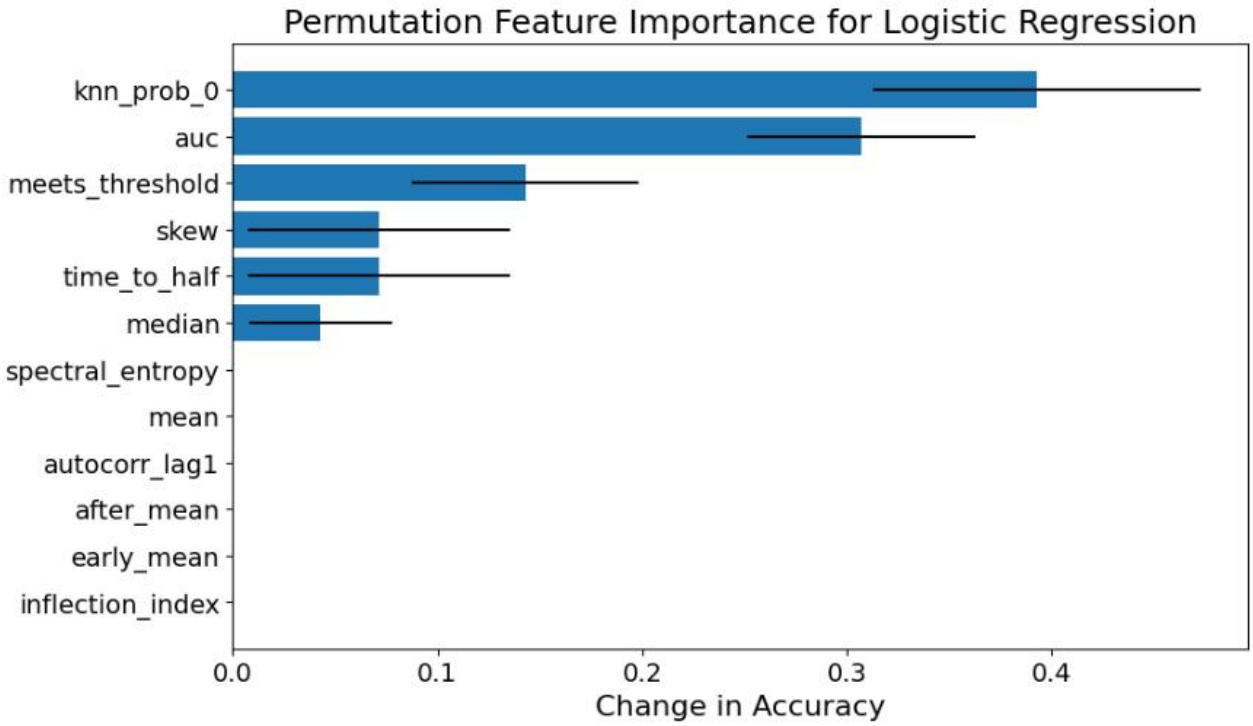


Figure E.16: Permutation importance for logistic regression for initial feature inputs for the COVID-19 dataset.

## E.6 Final Model Results

### E.6.1 COVID-19

```
Best parameters: {'criterion': 'gini', 'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 10}
Best cross-validation score: 0.9273684210526316

Classification report for training data:
              precision    recall  f1-score   support

           0       0.96      0.90      0.92        48
           1       0.90      0.96      0.93        48

    accuracy                           0.93        96
   macro avg       0.93      0.93      0.93        96
weighted avg       0.93      0.93      0.93        96

Training ROC-AUC score: 0.9270833333333334

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.78      1.00      0.88         7
           1       1.00      0.71      0.83         7

    accuracy                           0.86        14
   macro avg       0.89      0.86      0.85        14
weighted avg       0.89      0.86      0.85        14

Validation ROC-AUC score: 0.8571428571428572

Classification report for testing data:

              precision    recall  f1-score   support

           0       1.00      0.86      0.92        14
           1       0.88      1.00      0.93        14

    accuracy                           0.93        28
   macro avg       0.94      0.93      0.93        28
weighted avg       0.94      0.93      0.93        28

Testing ROC-AUC score: 0.9285714285714286
```

Figure E.17: Final results for decision tree for the COVID-19 dataset.

```
Best parameters: {'max_depth': 5, 'max_features': 0.8, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 50}
Best cross-validation score: 0.9268421052631579

Classification report for training data:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        48
           1       0.96      0.96      0.96        48

    accuracy                           0.96        96
   macro avg       0.96      0.96      0.96        96
weighted avg       0.96      0.96      0.96        96

Training ROC-AUC score: 0.9583333333333335

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.78      1.00      0.88         7
           1       1.00      0.71      0.83         7

    accuracy                           0.86        14
   macro avg       0.89      0.86      0.85        14
weighted avg       0.89      0.86      0.85        14

Validation ROC-AUC score: 0.8571428571428572

Classification report for testing data:

              precision    recall  f1-score   support

           0       1.00      0.86      0.92        14
           1       0.88      1.00      0.93        14

    accuracy                           0.93        28
   macro avg       0.94      0.93      0.93        28
weighted avg       0.94      0.93      0.93        28

Testing ROC-AUC score: 0.9285714285714286
```

Figure E.18: Final results for random forest for the COVID-19 dataset.

```
Best parameters: {'colsample_bytree': 1.0, 'gamma': 0, 'learning_rate': 0.2, 'max_depth': 3, 'min_child_weight': 1, 'n_estimators': 500, 'subsample': 0.8}
Best cross-validation score: 0.9273684210526316

Classification report for training data:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        48
           1       1.00      0.98      0.99        48

    accuracy                           0.99        96
   macro avg       0.99      0.99      0.99        96
weighted avg       0.99      0.99      0.99        96

Training ROC-AUC score: 0.9895833333333333

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.88      1.00      0.93         7
           1       1.00      0.86      0.92         7

    accuracy                           0.93        14
   macro avg       0.94      0.93      0.93        14
weighted avg       0.94      0.93      0.93        14

Validation ROC-AUC score: 0.9285714285714286

Classification report for testing data:

              precision    recall  f1-score   support

           0       1.00      0.86      0.92        14
           1       0.88      1.00      0.93        14

    accuracy                           0.93        28
   macro avg       0.94      0.93      0.93        28
weighted avg       0.94      0.93      0.93        28

Testing ROC-AUC score: 0.9285714285714286
```

Figure E.19: Final results for XGBoost for the COVID-19 dataset.

```
Best parameters: {'metric': 'manhattan', 'n_neighbors': 4, 'p': 1, 'weights': 'distance'}
Best cross-validation score: 0.8536842105263158

Classification report for training data:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        48
           1       1.00      1.00      1.00        48

    accuracy                           1.00        96
   macro avg       1.00      1.00      1.00        96
weighted avg       1.00      1.00      1.00        96

Training ROC-AUC score: 1.0

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.88      1.00      0.93         7
           1       1.00      0.86      0.92         7

    accuracy                           0.93        14
   macro avg       0.94      0.93      0.93        14
weighted avg       0.94      0.93      0.93        14

Validation ROC-AUC score: 0.9285714285714286

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.80      0.86      0.83        14
           1       0.85      0.79      0.81        14

    accuracy                           0.82        28
   macro avg       0.82      0.82      0.82        28
weighted avg       0.82      0.82      0.82        28

Testing ROC-AUC score: 0.8214285714285714
```

Figure E.20: Final results for k-nearest neighbour for the COVID-19 dataset.

```
Best parameters: {'alpha': 10}
Best cross-validation score: 0.9268421052631579

Classification report for training data:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94        48
           1       0.94      0.94      0.94        48

    accuracy                           0.94        96
   macro avg       0.94      0.94      0.94        96
weighted avg       0.94      0.94      0.94        96

Training ROC-AUC score: 0.9375

Classification report for validation data:

              precision    recall  f1-score   support

           0       1.00      0.86      0.92         7
           1       0.88      1.00      0.93         7

    accuracy                           0.93        14
   macro avg       0.94      0.93      0.93        14
weighted avg       0.94      0.93      0.93        14

Validation ROC-AUC score: 0.9285714285714286

Classification report for testing data:

              precision    recall  f1-score   support

           0       1.00      0.86      0.92        14
           1       0.88      1.00      0.93        14

    accuracy                           0.93        28
   macro avg       0.94      0.93      0.93        28
weighted avg       0.94      0.93      0.93        28

Testing ROC-AUC score: 0.9285714285714286
```

Figure E.21: Final results for ridge classifier for the COVID-19 dataset.

```
Best parameters: {'C': 100, 'class_weight': None, 'penalty': 'l2', 'solver': 'liblinear'}
Best cross-validation score: 0.884736842105263

Classification report for training data:
              precision    recall  f1-score   support

           0       0.90      0.92      0.91        48
           1       0.91      0.90      0.91        48

    accuracy                           0.91        96
   macro avg       0.91      0.91      0.91        96
weighted avg       0.91      0.91      0.91        96

Training ROC-AUC score: 0.90625

Classification report for validation data:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00         7
           1       1.00      1.00      1.00         7

    accuracy                           1.00        14
   macro avg       1.00      1.00      1.00        14
weighted avg       1.00      1.00      1.00        14

Validation ROC-AUC score: 1.0

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.92      0.86      0.89        14
           1       0.87      0.93      0.90        14

    accuracy                           0.89        28
   macro avg       0.89      0.89      0.89        28
weighted avg       0.89      0.89      0.89        28

Testing ROC-AUC score: 0.8928571428571429
```

Figure E.22: Final results for logistic regression for the COVID-19 dataset.

```
Best parameters: {'C': 0.1, 'class_weight': None, 'gamma': 'scale', 'kernel': 'linear'}
Best cross-validation score: 0.9268421052631579

Classification report for training data:
              precision    recall  f1-score   support

           0       0.94      0.96      0.95        48
           1       0.96      0.94      0.95        48

    accuracy                           0.95        96
   macro avg       0.95      0.95      0.95        96
weighted avg       0.95      0.95      0.95        96

Training ROC-AUC score: 0.9479166666666667

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.78      1.00      0.88         7
           1       1.00      0.71      0.83         7

    accuracy                           0.86        14
   macro avg       0.89      0.86      0.85        14
weighted avg       0.89      0.86      0.85        14

Validation ROC-AUC score: 0.8571428571428572

Classification report for testing data:

              precision    recall  f1-score   support

           0       1.00      0.86      0.92        14
           1       0.88      1.00      0.93        14

    accuracy                           0.93        28
   macro avg       0.94      0.93      0.93        28
weighted avg       0.94      0.93      0.93        28

Testing ROC-AUC score: 0.9285714285714286
```

Figure E.23: Final results for support vector classifier for the COVID-19 dataset.

```
Classification report for training data:
              precision    recall  f1-score   support

           0       0.94      0.96      0.95        48
           1       0.96      0.94      0.95        48

    accuracy                           0.95        96
   macro avg       0.95      0.95      0.95        96
weighted avg       0.95      0.95      0.95        96

Training ROC-AUC score: 0.9479166666666667

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.88      1.00      0.93         7
           1       1.00      0.86      0.92         7

    accuracy                           0.93        14
   macro avg       0.94      0.93      0.93        14
weighted avg       0.94      0.93      0.93        14

Validation ROC-AUC score: 0.9285714285714286

Classification report for validation data:

              precision    recall  f1-score   support

           0       1.00      0.86      0.92        14
           1       0.88      1.00      0.93        14

    accuracy                           0.93        28
   macro avg       0.94      0.93      0.93        28
weighted avg       0.94      0.93      0.93        28

Validation ROC-AUC score: 0.9285714285714286
```

Figure E.24: Final results for voting ensemble classifier for the COVID-19 dataset.

```
Classification report for training data:
              precision    recall  f1-score   support

           0       0.91      0.90      0.91        48
           1       0.90      0.92      0.91        48

    accuracy                           0.91        96
   macro avg       0.91      0.91      0.91        96
weighted avg       0.91      0.91      0.91        96

Training ROC-AUC score: 0.90625

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.88      1.00      0.93         7
           1       1.00      0.86      0.92         7

    accuracy                           0.93        14
   macro avg       0.94      0.93      0.93        14
weighted avg       0.94      0.93      0.93        14

Validation ROC-AUC score: 0.9285714285714286

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.91      0.71      0.80        14
           1       0.76      0.93      0.84        14

    accuracy                           0.82        28
   macro avg       0.84      0.82      0.82        28
weighted avg       0.84      0.82      0.82        28

Validation ROC-AUC score: 0.8214285714285715
```

Figure E.25: Final results for the DTW and k-nearest neighbour method for the COVID-19 dataset.

Table E.9: Model results for LSTM (hidden size 64) for the COVID-19 dataset.

| **Evaluation metric** | **Result (3 s.f.)** |
|---|---|
| Accuracy | 78.6% |
| Log loss | 0.501 |
| Recall (sensitivity) | 73.8% |
| Precision | 79.2% |
| Specificity | 87.8% |
| F1-score | 74.4% |
| AUC-ROC | 81.8% |

Table E.10: Model results for LSTM (hidden size 128) for the COVID-19 dataset.

| **Evaluation metric** | **Result (3 s.f.)** |
|---|---|
| Accuracy | 82.1% |
| Log loss | 0.411 |
| Recall (sensitivity) | 81.0% |
| Precision | 82.1% |
| Specificity | 87.8% |
| F1-score | 80.4% |
| AUC-ROC | 85.4% |

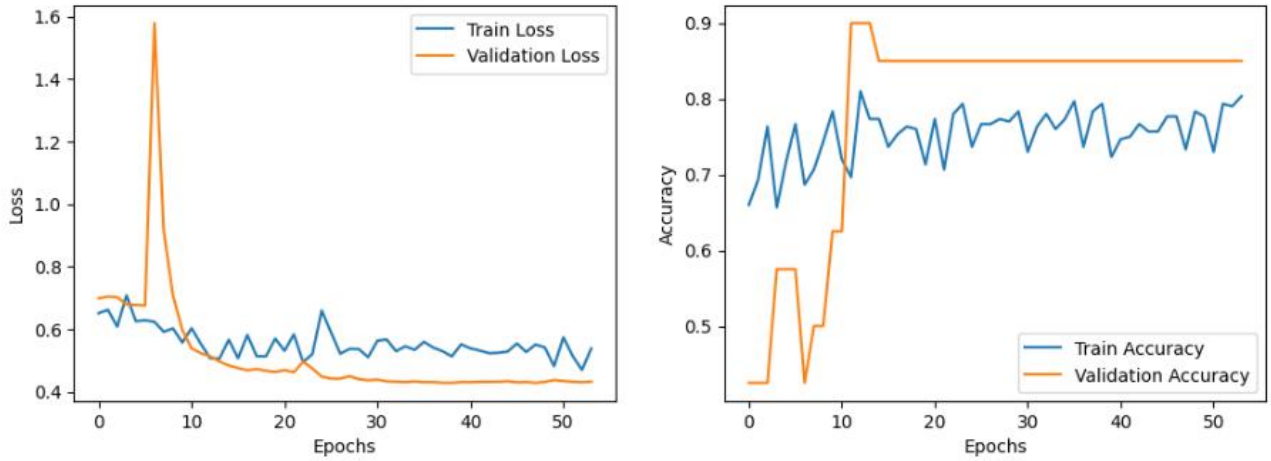


Figure E.26: LSTM learning curves for the COVID-19 dataset, with LSTM hidden size 64.

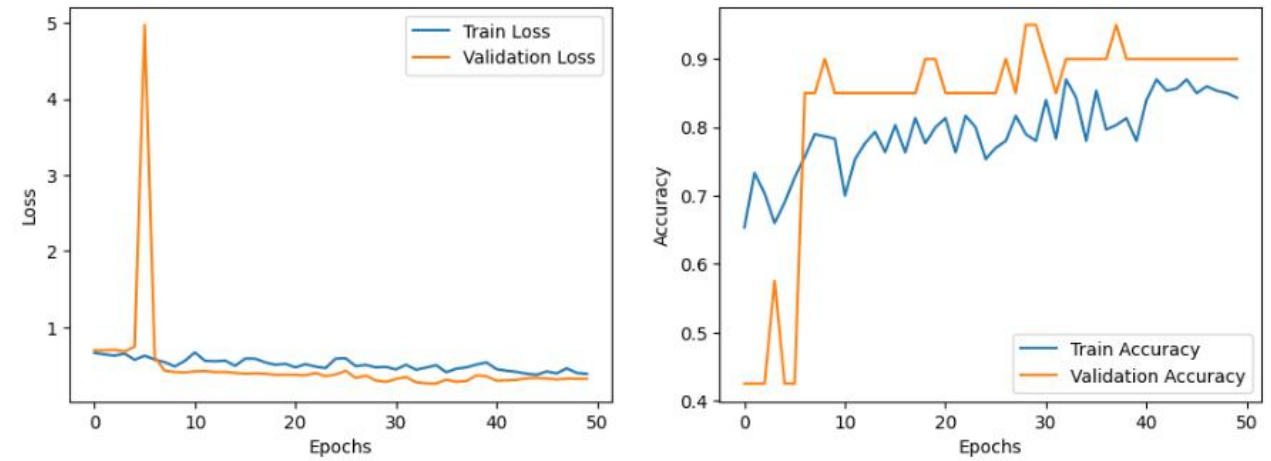


Figure E.27: LSTM learning curves for the COVID-19 dataset, with LSTM hidden size 128.

### E.6.2 Chlamydia

```
Best parameters: {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2}
Best cross-validation score: 0.7798573975044564

Classification report for training data:
              precision    recall  f1-score   support

           0       0.87      0.89      0.88        84
           1       0.89      0.87      0.88        84

    accuracy                           0.88       168
   macro avg       0.88      0.88      0.88       168
weighted avg       0.88      0.88      0.88       168

Training ROC-AUC score: 0.8809523809523809

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.69      0.92      0.79        12
           1       0.89      0.62      0.73        13

    accuracy                           0.76        25
   macro avg       0.79      0.77      0.76        25
weighted avg       0.79      0.76      0.76        25

Validation ROC-AUC score: 0.7660256410256411

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.83      0.96      0.89        25
           1       0.95      0.79      0.86        24

    accuracy                           0.88        49
   macro avg       0.89      0.88      0.88        49
weighted avg       0.89      0.88      0.88        49

Testing ROC-AUC score: 0.8758333333333332
```

Figure E.28: Final results for decision tree for the chlamydia dataset.

```
Best parameters: {'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Best cross-validation score: 0.8383244206773618

Classification report for training data:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        84
           1       1.00      1.00      1.00        84

    accuracy                           1.00       168
   macro avg       1.00      1.00      1.00       168
weighted avg       1.00      1.00      1.00       168

Training ROC-AUC score: 1.0

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.79      0.92      0.85        12
           1       0.91      0.77      0.83        13

    accuracy                           0.84        25
   macro avg       0.85      0.84      0.84        25
weighted avg       0.85      0.84      0.84        25

Validation ROC-AUC score: 0.8429487179487178

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.85      0.88      0.86        25
           1       0.87      0.83      0.85        24

    accuracy                           0.86        49
   macro avg       0.86      0.86      0.86        49
weighted avg       0.86      0.86      0.86        49

Testing ROC-AUC score: 0.8566666666666668
```

Figure E.29: Final results for random forest for the chlamydia dataset.

```
Best parameters: {'colsample_bytree': 1.0, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 7, 'min_child_weight': 3, 'n_estimators': 500, 'subsample': 1.0}
Best cross-validation score: 0.8267379679144385

Classification report for training data:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        84
           1       1.00      1.00      1.00        84

    accuracy                           1.00       168
   macro avg       1.00      1.00      1.00       168
weighted avg       1.00      1.00      1.00       168

Training ROC-AUC score: 1.0

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.65      0.92      0.76        12
           1       0.88      0.54      0.67        13

    accuracy                           0.72        25
   macro avg       0.76      0.73      0.71        25
weighted avg       0.77      0.72      0.71        25

Validation ROC-AUC score: 0.7275641025641025

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.74      0.80      0.77        25
           1       0.77      0.71      0.74        24

    accuracy                           0.76        49
   macro avg       0.76      0.75      0.75        49
weighted avg       0.76      0.76      0.75        49

Testing ROC-AUC score: 0.7541666666666668
```

Figure E.30: Final results for XGBoost for the chlamydia dataset.

```
Best parameters: {'metric': 'euclidean', 'n_neighbors': 11, 'p': 1, 'weights': 'distance'}
Best cross-validation score: 0.785204991087344

Classification report for training data:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        84
           1       1.00      1.00      1.00        84

    accuracy                           1.00       168
   macro avg       1.00      1.00      1.00       168
weighted avg       1.00      1.00      1.00       168

Training ROC-AUC score: 1.0

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.73      0.92      0.81        12
           1       0.90      0.69      0.78        13

    accuracy                           0.80        25
   macro avg       0.82      0.80      0.80        25
weighted avg       0.82      0.80      0.80        25

Validation ROC-AUC score: 0.8044871794871794

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.79      0.92      0.85        25
           1       0.90      0.75      0.82        24

    accuracy                           0.84        49
   macro avg       0.85      0.83      0.84        49
weighted avg       0.85      0.84      0.84        49

Testing ROC-AUC score: 0.8350000000000001
```

Figure E.31: Final results for k-nearest neighbour for the chlamydia dataset.

```
Best parameters: {'alpha': 1}
Best cross-validation score: 0.6547237076648841

Classification report for training data:
              precision    recall  f1-score   support

           0       0.69      0.61      0.65        84
           1       0.65      0.73      0.69        84

    accuracy                           0.67       168
   macro avg       0.67      0.67      0.67       168
weighted avg       0.67      0.67      0.67       168

Training ROC-AUC score: 0.6666666666666667

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.64      0.75      0.69        12
           1       0.73      0.62      0.67        13

    accuracy                           0.68        25
   macro avg       0.69      0.68      0.68        25
weighted avg       0.69      0.68      0.68        25

Validation ROC-AUC score: 0.6826923076923077

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.64      0.72      0.68        25
           1       0.67      0.58      0.62        24

    accuracy                           0.65        49
   macro avg       0.65      0.65      0.65        49
weighted avg       0.65      0.65      0.65        49

Testing ROC-AUC score: 0.6516666666666667
```

Figure E.32: Final results for ridge classifier for the chlamydia dataset.

```
Best parameters: {'C': 10, 'class_weight': None, 'penalty': 'l2', 'solver': 'liblinear'}
Best cross-validation score: 0.6549019607843137

Classification report for training data:
              precision    recall  f1-score   support

           0       0.68      0.63      0.65        84
           1       0.66      0.70      0.68        84

    accuracy                           0.67       168
   macro avg       0.67      0.67      0.67       168
weighted avg       0.67      0.67      0.67       168

Training ROC-AUC score: 0.6666666666666667

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.64      0.75      0.69        12
           1       0.73      0.62      0.67        13

    accuracy                           0.68        25
   macro avg       0.69      0.68      0.68        25
weighted avg       0.69      0.68      0.68        25

Validation ROC-AUC score: 0.6826923076923077

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.64      0.72      0.68        25
           1       0.67      0.58      0.62        24

    accuracy                           0.65        49
   macro avg       0.65      0.65      0.65        49
weighted avg       0.65      0.65      0.65        49

Testing ROC-AUC score: 0.6516666666666667
```

Figure E.33: Final results for logistic regression for the chlamydia dataset.

```
Best parameters: {'C': 100, 'class_weight': None, 'gamma': 'auto', 'kernel': 'rbf'}
Best cross-validation score: 0.7616755793226381

Classification report for training data:
              precision    recall  f1-score   support

           0       0.84      0.95      0.89        84
           1       0.95      0.82      0.88        84

    accuracy                           0.89       168
   macro avg       0.89      0.89      0.89       168
weighted avg       0.89      0.89      0.89       168

Training ROC-AUC score: 0.8869047619047619

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.71      0.83      0.77        12
           1       0.82      0.69      0.75        13

    accuracy                           0.76        25
   macro avg       0.77      0.76      0.76        25
weighted avg       0.77      0.76      0.76        25

Validation ROC-AUC score: 0.7628205128205129

Classification report for testing data:

              precision    recall  f1-score   support

           0       0.74      0.80      0.77        25
           1       0.77      0.71      0.74        24

    accuracy                           0.76        49
   macro avg       0.76      0.75      0.75        49
weighted avg       0.76      0.76      0.75        49

Testing ROC-AUC score: 0.7541666666666668
```

Figure E.34: Final results for support vector classifier for the chlamydia dataset.

```
Classification report for training data:
              precision    recall  f1-score   support

           0       0.77      0.89      0.82        84
           1       0.87      0.73      0.79        84

    accuracy                           0.81       168
   macro avg       0.82      0.81      0.81       168
weighted avg       0.82      0.81      0.81       168

Training ROC-AUC score: 0.8095238095238096

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.69      0.92      0.79        12
           1       0.89      0.62      0.73        13

    accuracy                           0.76        25
   macro avg       0.79      0.77      0.76        25
weighted avg       0.79      0.76      0.76        25

Validation ROC-AUC score: 0.7660256410256411

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.69      0.88      0.77        25
           1       0.82      0.58      0.68        24

    accuracy                           0.73        49
   macro avg       0.76      0.73      0.73        49
weighted avg       0.75      0.73      0.73        49

Validation ROC-AUC score: 0.7316666666666668
```

Figure E.35: Final results for voting ensemble classifier for the chlamydia dataset.

```
Classification report for training data:
              precision    recall  f1-score   support

           0       0.87      0.85      0.86        84
           1       0.85      0.87      0.86        84

    accuracy                           0.86       168
   macro avg       0.86      0.86      0.86       168
weighted avg       0.86      0.86      0.86       168

Training ROC-AUC score: 0.8571428571428572

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.67      0.67      0.67        12
           1       0.69      0.69      0.69        13

    accuracy                           0.68        25
   macro avg       0.68      0.68      0.68        25
weighted avg       0.68      0.68      0.68        25

Validation ROC-AUC score: 0.6794871794871796

Classification report for validation data:

              precision    recall  f1-score   support

           0       0.78      0.56      0.65        25
           1       0.65      0.83      0.73        24

    accuracy                           0.69        49
   macro avg       0.71      0.70      0.69        49
weighted avg       0.71      0.69      0.69        49

Validation ROC-AUC score: 0.6966666666666668
```

Figure E.36: Final results for the DTW and k-nearest neighbour method for the chlamydia dataset.

Table E.11: Model results for LSTM (hidden size 64) for the chlamydia dataset.

| **Evaluation metric** | **Result (3 s.f.)** |
|---|---|
| Accuracy | 69.4% |
| Log loss | 0.546 |
| Recall (sensitivity) | 83.3% |
| Precision | 65.2% |
| Specificity | 51.7% |
| F1-score | 72.4% |
| AUC-ROC | 76.4% |

Table E.12: Model results for LSTM (hidden size 128) for the chlamydia dataset.

| **Evaluation metric** | **Result (3 s.f.)** |
|---|---|
| Accuracy | 71.4% |
| Log loss | 0.582 |
| Recall (sensitivity) | 100% |
| Precision | 63.2% |
| Specificity | 40.8% |
| F1-score | 76.3% |
| AUC-ROC | 79.1% |

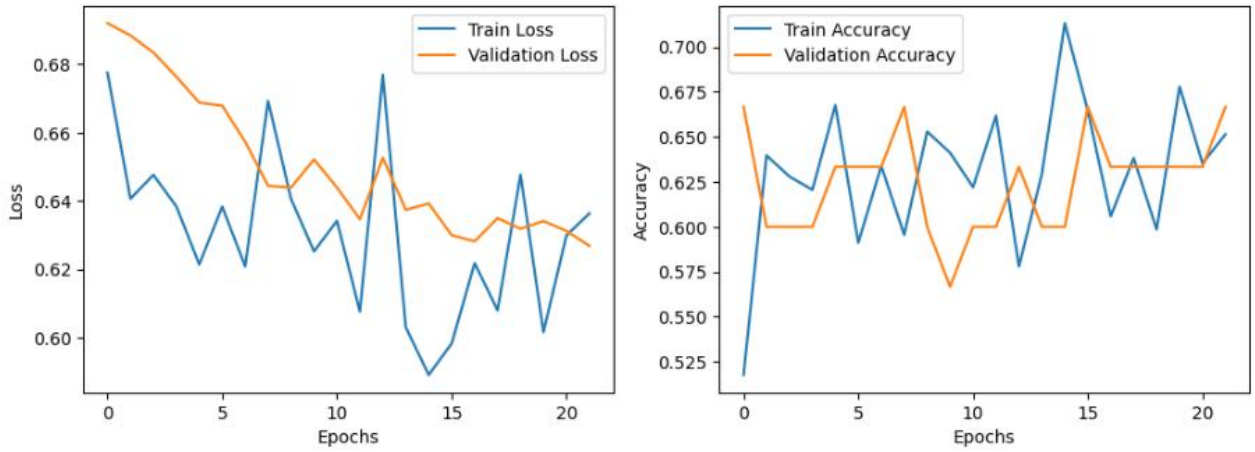


Figure E.37: LSTM learning curves for the chlamydia dataset, with LSTM hidden size 64.

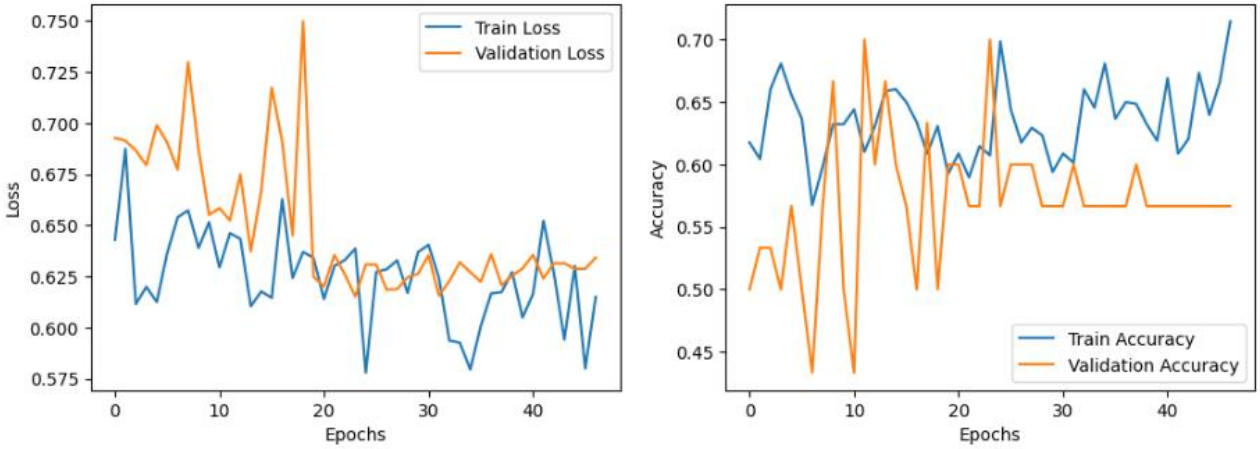


Figure E.38: LSTM learning curves for the chlamydia dataset, with LSTM hidden size 128.

## E.7 Explainability and Interpretability

### E.7.1 Ridge Classification

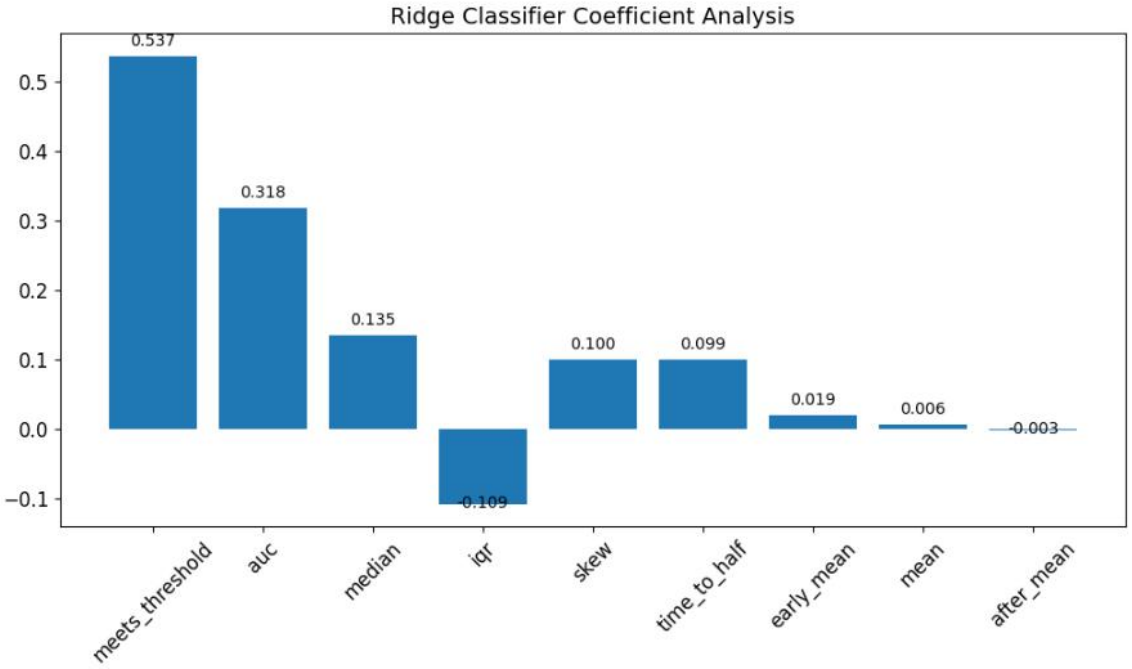


Figure E.39: Ridge classifier coefficient analysis.

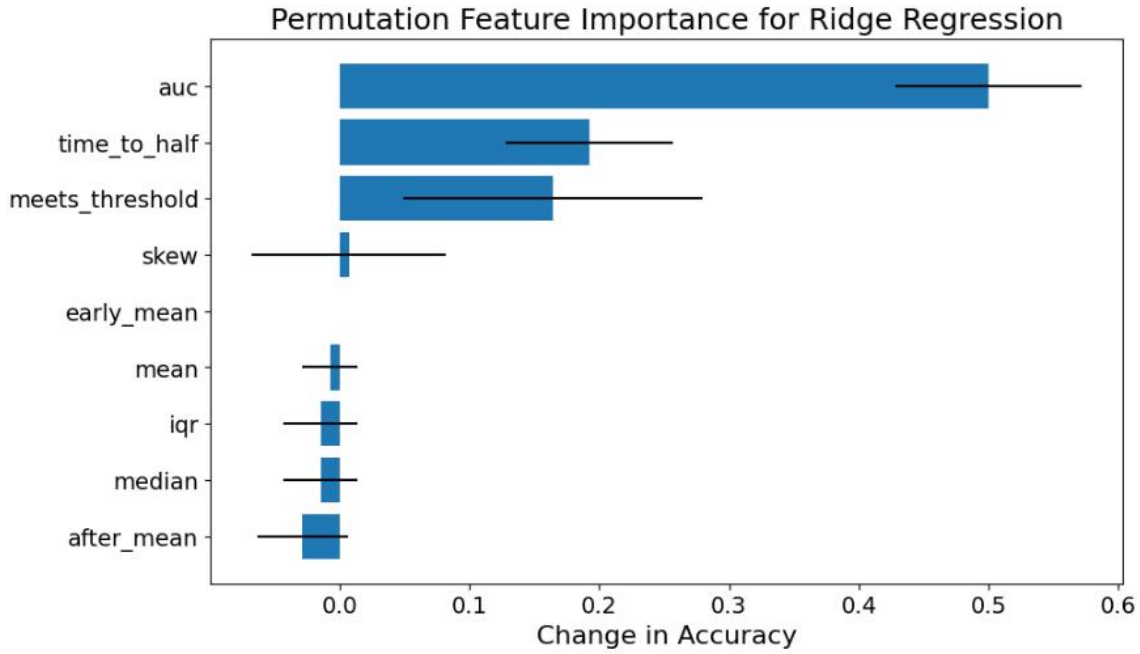


Figure E.40: Ridge classifier permutation importance.

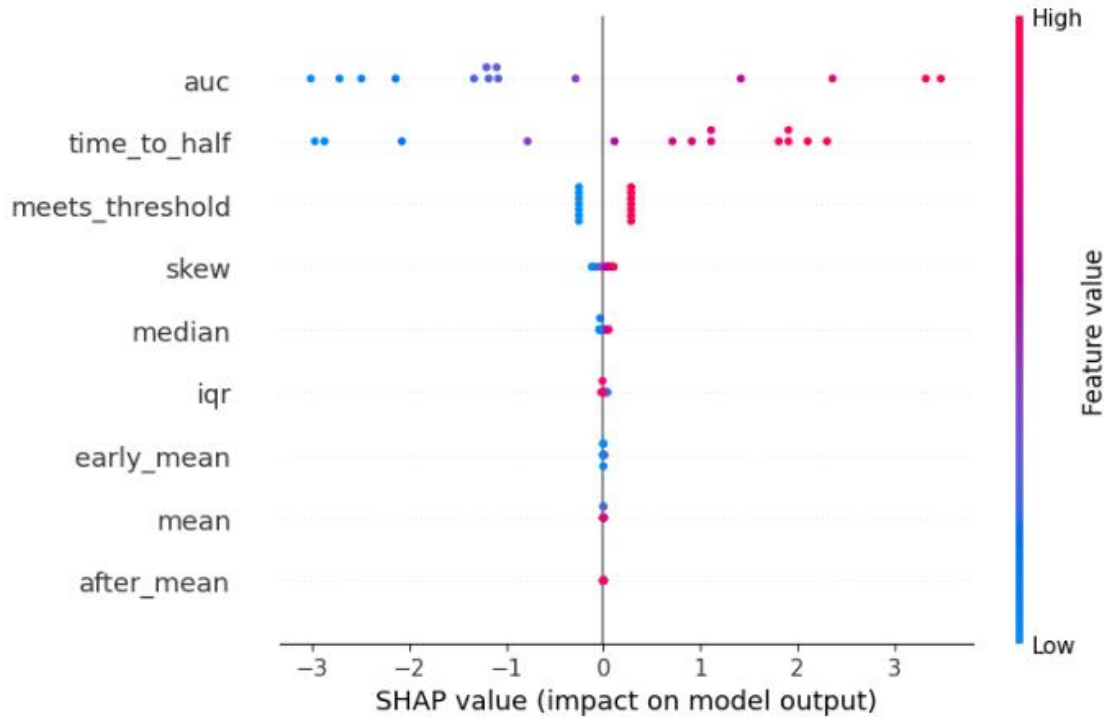


Figure E.41: Ridge classifier SHAP values on the COVID-19 validation set.

### E.7.2 Random Forest

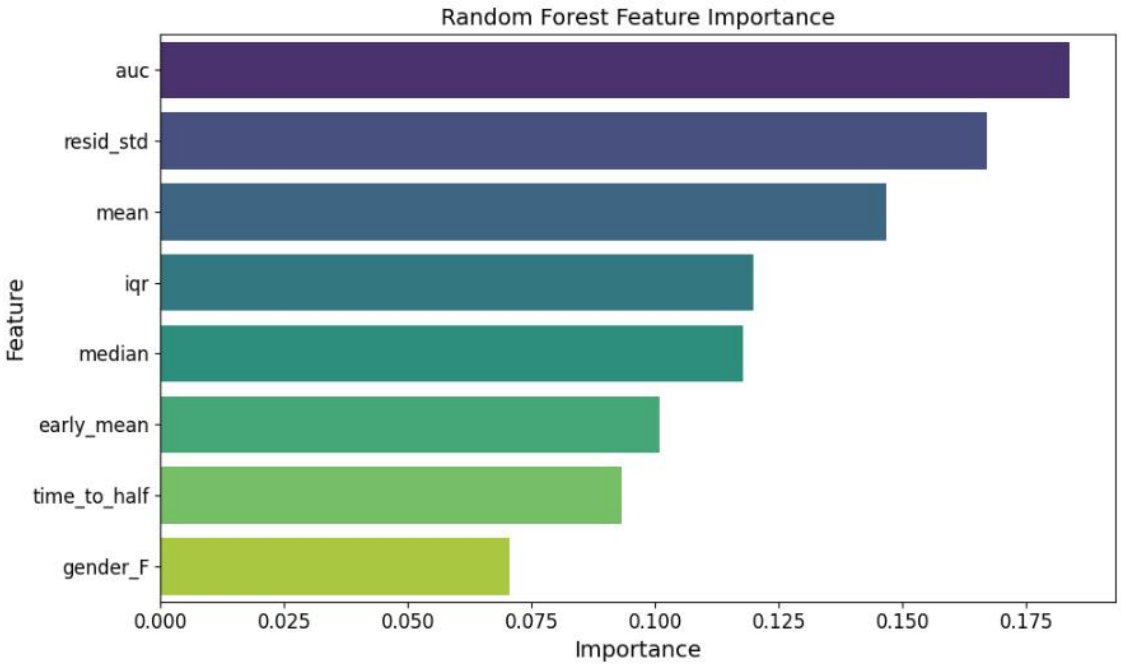


Figure E.42: Random forest features importances (based on Gini index) for the chlamydia data.

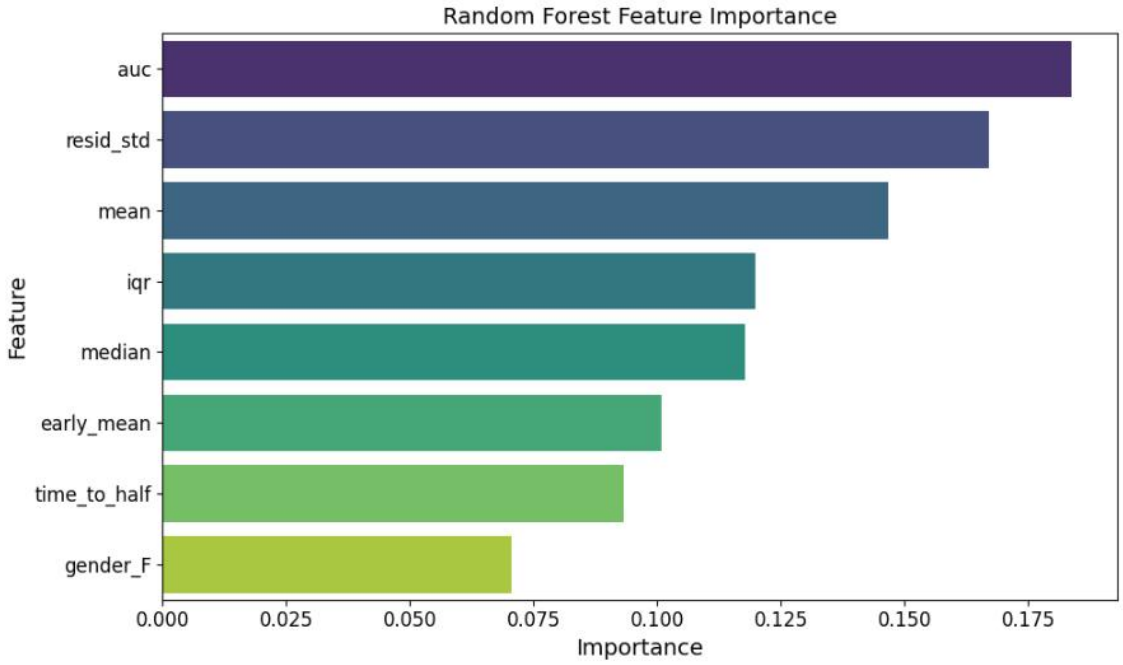


Figure E.43: Random forest permutation importance for the chlamydia validation set.


Figure E.44: Random forest LIME result for first test sample in the chlamydia testing set.

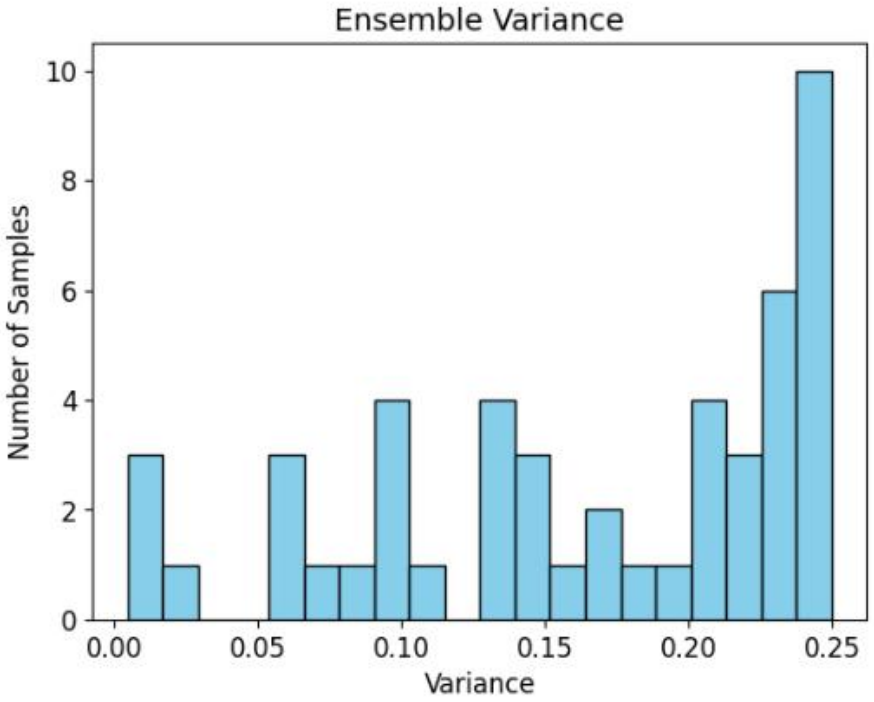


Figure E.45: Random forest ensemble variance (for epistemic uncertainty for the chlamydia testing data.

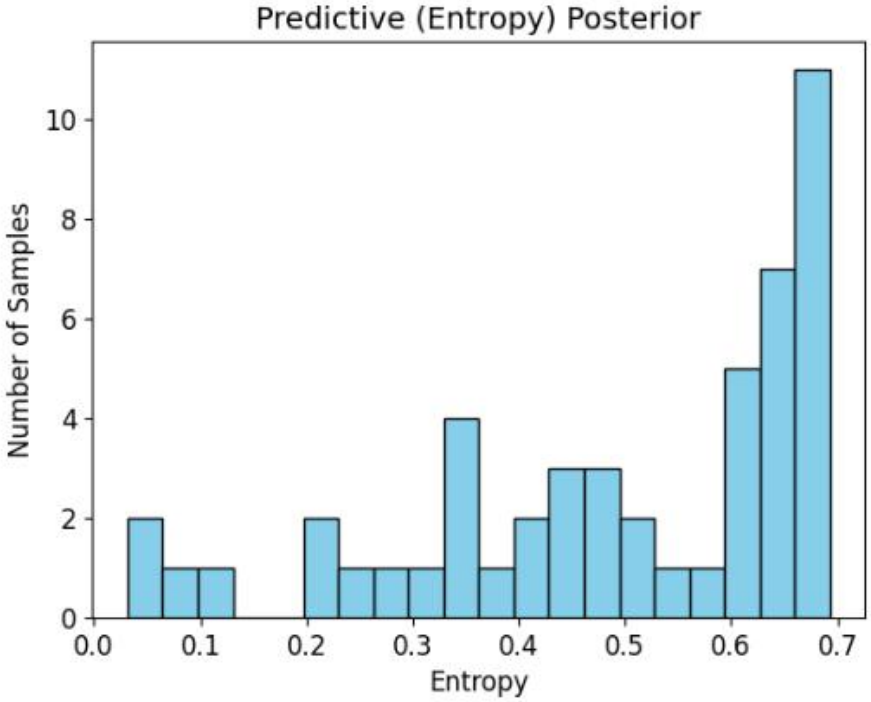


Figure E.46: Random forest predictive (Entropy) posterior for the chlamydia testing set.

# Appendix F Code

All code (IPython Notebook files) can be found on GitHub through Roberts (2025). The data will not be available here due to proprietary reasons.

# Appendix G Gantt Chart of the Project Timeline

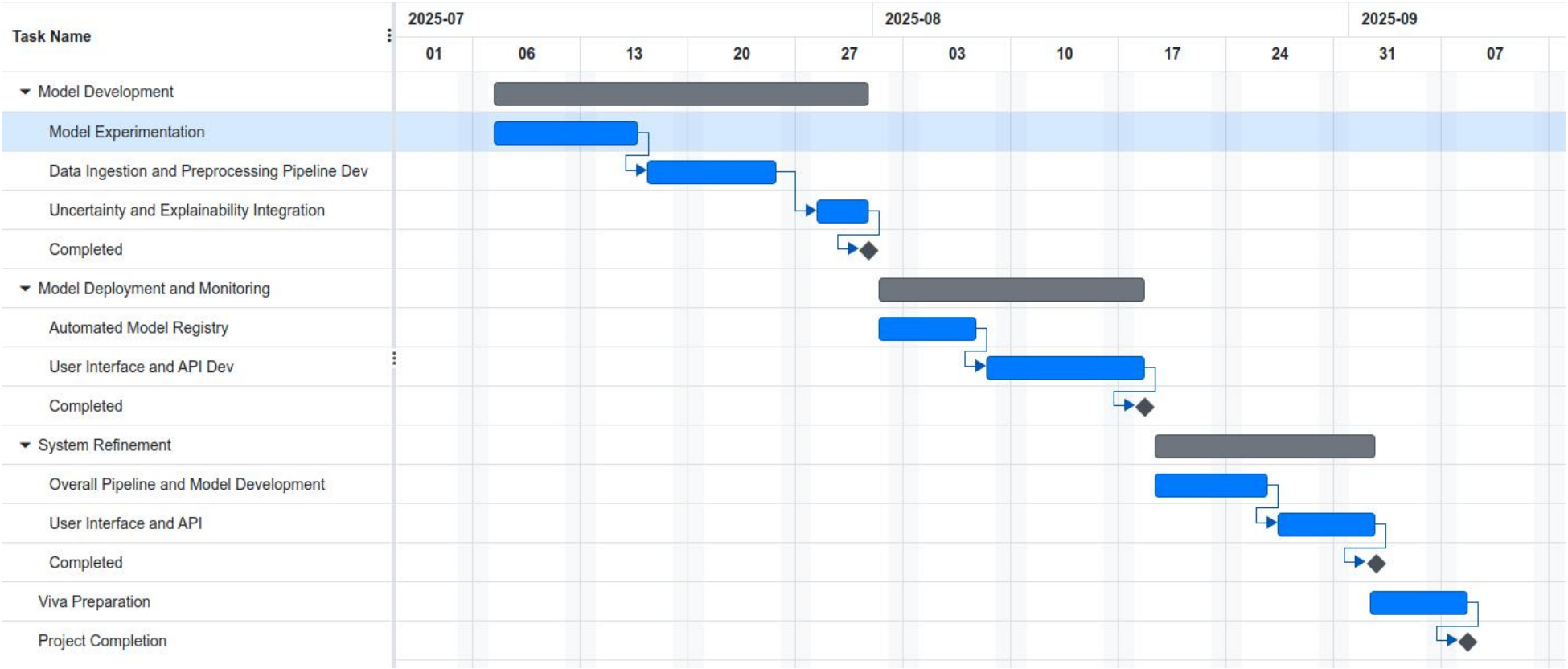


Figure G.1: Originally proposed project timeline.

| Task Name | Start | End | Duration | Progress % |
|---|---|---|---|---|
| ▾ Model Development | 2025-07-07 | 2025-07-31 | 19 days | 25 |
| Model Experimentation | 2025-07-07 | 2025-07-16 | 8 days | 60 |
| Data Ingestion and Preprocessing Pipeline Dev | 2025-07-17 | 2025-07-25 | 7 days | 0 |
| Uncertainty and Explainability Integration | 2025-07-28 | 2025-07-31 | 4 days | 0 |
| Completed | 2025-07-31 | 2025-07-31 | 0 days | 0 |
| ▾ Model Deployment and Monitoring | 2025-08-01 | 2025-08-18 | 12 days | 0 |
| Automated Model Registry | 2025-08-01 | 2025-08-07 | 5 days | 0 |
| User Interface and API Dev | 2025-08-08 | 2025-08-18 | 7 days | 0 |
| Completed | 2025-08-18 | 2025-08-18 | 0 days | 0 |
| ▾ System Refinement | 2025-08-19 | 2025-09-02 | 11 days | 0 |
| Overall Pipeline and Model Development | 2025-08-19 | 2025-08-26 | 6 days | 0 |
| User Interface and API | 2025-08-27 | 2025-09-02 | 5 days | 0 |
| Completed | 2025-09-02 | 2025-09-02 | 0 days | 0 |
| Viva Preparation | 2025-09-02 | 2025-09-08 | 5 days | 0 |
| Project Completion | 2025-09-08 | 2025-09-08 | 0 days | 0 |

Figure G.2: Original project breakdown.

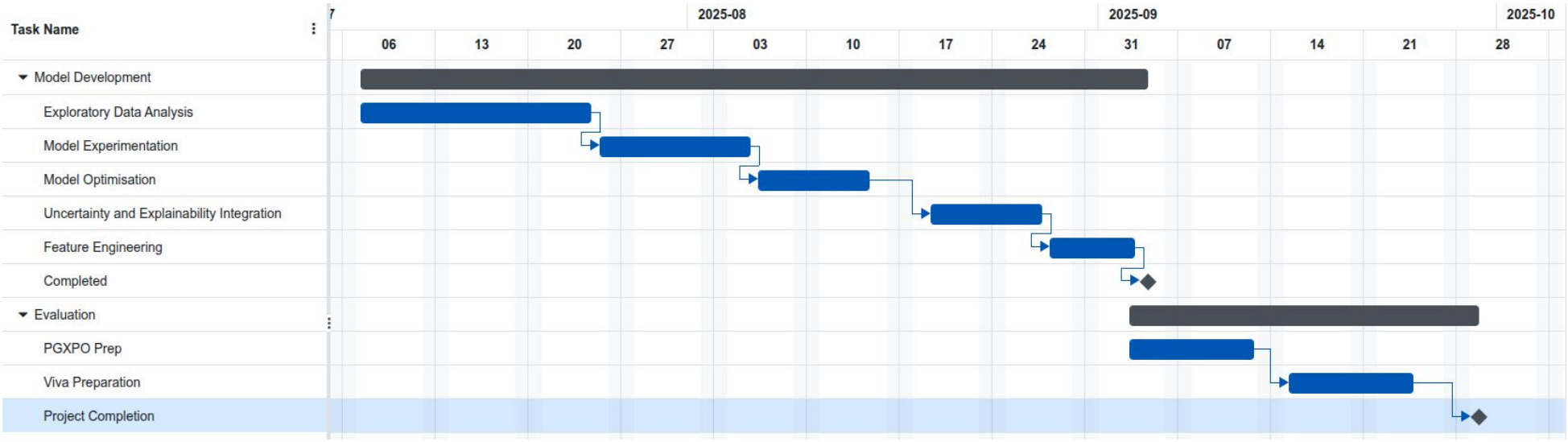


Figure G.3: Project timeline.

| Task Name | Start | End | Duration | Progress % |
|---|---|---|---|---|
| ▾ Model Development | 2025-07-07 | 2025-09-04 | 44 days | 100 |
| Exploratory Data Analysis | 2025-07-07 | 2025-07-24 | 14 days | 100 |
| Model Experimentation | 2025-07-25 | 2025-08-05 | 8 days | 100 |
| Model Optimisation | 2025-08-06 | 2025-08-14 | 7 days | 100 |
| Uncertainty and Explainability Integration | 2025-08-19 | 2025-08-27 | 7 days | 100 |
| Feature Engineering | 2025-08-28 | 2025-09-03 | 5 days | 100 |
| Completed | 2025-09-04 | 2025-09-04 | 0 days | 0 |
| ▾ Evaluation | 2025-09-03 | 2025-09-29 | 19 days | 100 |
| PGXPO Prep | 2025-09-03 | 2025-09-12 | 8 days | 100 |
| Viva Preparation | 2025-09-15 | 2025-09-24 | 8 days | 100 |
| Project Completion | 2025-09-29 | 2025-09-29 | 0 days | 0 |

Figure G.4: Project breakdown.