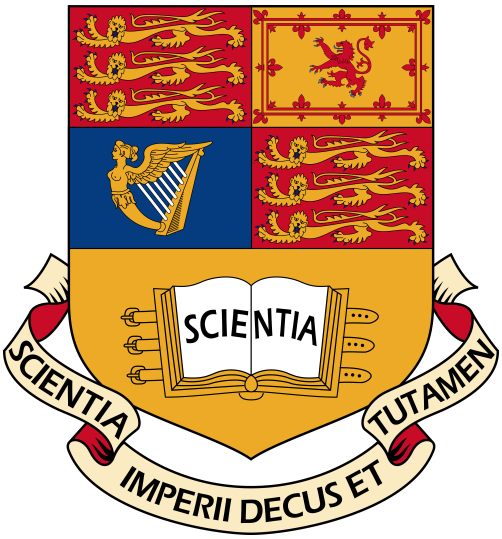
Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2023



Project Title:	**Intelligent Algorithms for DNA Detection**
Student:	**Sungjun (Andy) Han**
CID:	**01708685**
Course:	**EIE4**
Project Supervisor:	**Dr Jesus Rodriguez-Manzano**
Second Marker:	**Professor Timothy Constandinou**

Final Report Plagiarism Statement

I affirm that I have submitted, or will submit, an electronic copy of my final year project report to the provided EEE link.

I affirm that I have submitted, or will submit, an identical electronic copy of my final year project to the provided Blackboard module for Plagiarism checking.

I affirm that I have provided explicit references for all the material in my Final Report that is not authored by me, but is represented as my own work.

I have used ChatGPT v4 as an aid in the preparation of my report. I have used it to improve the quality of my English throughout, however all technical content and references comes from my original text.

Abstract

Loop-Mediated Isothermal Amplification (LAMP) is a widely used technique for amplifying and detecting DNA sequences. It involves plotting the amplification curve of fluorescence intensity against time cycles, allowing the identification of an amplification reaction, which indicates the presence of the target DNA sequence. However, the application of LAMP for testing multiple DNA primers in a single sample, known as "multiplex LAMP," poses a significant challenge due to difficulties in distinguishing among different amplification reactions. To overcome this limitation, Malpartida-Cardenas et al. proposed a multiplex LAMP technique integrated with amplification curve analysis, employing a machine learning model to automatically classify each curve to its corresponding target DNA. This project aims to enhance this approach by integrating a deep learning Transformer-based model, with the goal of improving overall performance. Additionally, the melting curve, which represents the change in fluorescence intensity against temperature after the amplification reaction, was explored as an alternative using similar deep learning techniques. In future investigations, training the developed framework with transfer learning could be explored to enhance the model's performance and adaptability.

Acknowledgements

Firstly, I would like to express my sincere gratitude to Dr. Jesus Rodriguez-Manzano for providing me with this project opportunity and offering invaluable guidance throughout. It has been an immensely enriching experience under their mentorship. Secondly, I extend my appreciation to Mr. Louis Kreitmann for his unwavering support and extensive guidance throughout this challenging yet fascinating project. His expertise and encouragement have been instrumental in its success.

Additionally, I am deeply grateful to all my friends and family for their support and encouragement throughout this project, as well as during my MEng studies. Their belief in me and their constant presence have been a source of strength and inspiration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Motivation

Loop-mediated isothermal amplification (LAMP) is a fundamental tool utilised in the field of Deoxyribonucleic acid (DNA) sequence detection. This method employs a DNA polymerase with robust strand displacement activity, and a suite of four to six unique primers that recognize six to eight regions of the target DNA sequence [1]. Unlike conventional techniques, LAMP initiates at relatively lower temperatures, producing a substantial amount of amplified product via auto-cycling strand displacement DNA synthesis, resulting in a series of stem-loop DNA structures, all the while sustaining a constant temperature [2].

Amidst the escalated demand for SARS-CoV-2 tests and the rising popularity of Polymerase Chain Reaction (PCR), interest in LAMP as an economical diagnostic technique for identifying infectious diseases has increased. PCR is a process that identifies and amplifies a specific DNA region by denaturing the DNA strand into single strands, allowing primers to anneal to each original strand for new strand synthesis, and extending new DNA strands from the primers. This denaturing step demands a thermal cyclical process, wherein the temperature must be repetitively increased to denature the DNA strands into single strands.

Conversely, LAMP provides advantages over PCR due to its isothermal nature, as delineated in Figure 1.1 [3]. The pressing need for DNA sequence detection tools, accentuated by the SARS-CoV-2 pandemic, underscores the essentiality of prompt disease detection, facilitating immediate treatment and mitigating potential viral spread. In comparison to traditional PCR, LAMP offers quicker genetic material analysis and has shown efficacy in detecting COVID-19 [4]. Furthermore, LAMP's utility extends beyond SARS-CoV-2 detection; it possesses the capacity to identify various DNA sequences, thereby broadening its potential applications.

While identifying a single DNA sequence suffices for cases like identifying SARS-CoV-2, many LAMP applications involve the simultaneous detection of multiple DNA sequences within a single reaction, known as multiplex LAMP [5]. Multiplex LAMP enables the differential detection of multiple diseases, reducing the number of LAMP experiments required per patient and subsequently

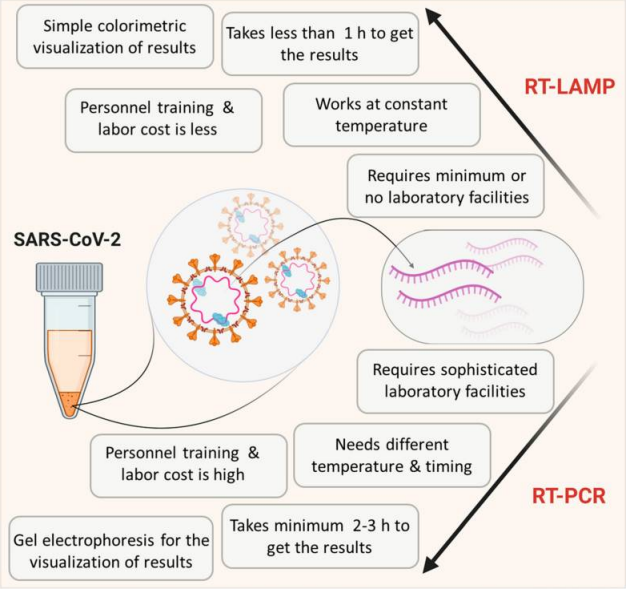


Figure 1.1: Comparison of reverse transcription loop-mediated isothermal amplification (RT-LAMP) with reverse transcription polymerase chain reaction (RT-PCR) by Augustine et al. [3]

lowering costs and saving time. In healthcare settings, it is crucial to provide patients with fast, cost-effective, and easy-to-operate feedback, highlighting the significance of multiplex LAMP.

The current approach for multiplex LAMP involves combining a target DNA sequence with a specific combination of primer sets in a single tube, followed by an isothermal reaction that induces a fluorescence change, resulting in an amplification curve representing the target RNA amplification. Subsequently, the target RNA is extracted for further analysis [6].

However, distinguishing amplification curves based on their corresponding targets poses a challenge due to their similar shapes and features. To overcome this challenge, there is a proposed solution of developing a deep learning-based algorithm. Such an algorithm would effectively classify DNA sequences based on their respective amplification curves, providing significant benefits in the clinical field of disease detection within the context of multiplex LAMP.

Currently, the field of classifying amplification curves in multiplex LAMP heavily relies on machine learning techniques like k-nearest neighbors (k-NN) [7]. However, there is a potential for significant advancements by incorporating state-of-the-art deep learning models, such as Long Short-Term Memory (LSTM) or transformer frameworks, which have proven effective in processing time-series data.

Additionally, it is worth exploring the use of a new model that leverages the melting curve for the classification task. Although the melting curve introduces an extra step compared to the original amplification process and loses the isothermal advantage of LAMP experiments, the additional characteristics it provides could potentially enhance the accuracy of DNA sequence classification.

The development of a deep learning algorithm capable of efficiently and accurately classifying amplification curves and melting curves would be highly motivating and could have a profound impact in the field.

1.1.1 Project Aims

As mentioned in the section above, the report aims to develop a state-of-the-art deep learning framework that will excel previous works in classifying amplification curves to the target DNA sequence.

In order the fulfill the project aims, the following steps are to be taken:

- Pre-process and refine the given multiplex LAMP data to remove outliers and standardise the curves.
- Re-create and define standards using baseline models from previous works.
- Use an LSTM model before the transformer model as an alternative baseline model used to compare frameworks for time-series data.
- Design state-of-the-art transformer model to analyse and classify the amplification curve.
- Optimise hyperparameters to achieve the highest accuracy model.
- Attempt melt curve data as an alternative to amplification curve data.
- Benchmark performance between previous models and the new transformer model

Additional details on the steps to be taken will be further discussed in Section 3, Requirements Capture.

1.1.2 Report Structure

The report is structured as follows:

1. Introduction
    - Highlights the significance of multiplex LAMP and the need for deep learning models in analyzing and classifying amplification curves.
2. Background Material
    - Illustrates the biological and machine learning background knowledge needed to execute this project.
3. Requirements Capture
    - Contains the technical requirements and the optimal performance expected from the final product.
4. Analysis and Design
    - Discusses the high level overview, along with an exploration of the three pivotal steps involved in executing the project.

5. Implementation

    - Demonstrates how the model was implemented and the logical design of the final model.

6. Testing and Results

    - Assesses the transformer model by analyzing its amplification curve and melting curve to determine numerical performance measures, including accuracy, precision, and other relevant metrics.

7. Evaluation

    - Compares the performance of the final transformer model with baseline models used in previous studies. Evaluates the model's success based on performance measures and considers the biological implications of the results.

8. Conclusion and Future Work

    - Concludes the project and provides potential development options for future advancements.

Chapter 2

Background Material

Section 2 presents a comprehensive overview of the biological and deep learning foundations required for the project. This section will commence with an introduction to a standard DNA amplification process, encompassing LAMP, qLAMP, and dLAMP techniques. Additionally, it will delve into the fundamental machine learning and deep learning concepts essential for accomplishing the classification objective. Furthermore, the section will provide valuable insights into leveraging multiplex LAMP in conjunction with deep learning classification algorithms to optimize the efficiency of DNA detection.

2.1 DNA

2.1.1 DNA Amplification

DNA amplification is a fundamental and indispensable procedure in the replication of DNA samples during DNA analysis. Its significance lies in its pivotal role in the identification of specific DNA targets within a given sample. By employing this process, numerous copies of a particular DNA sequence can be produced, even when the initial amount of DNA is limited. The amplification process aids in the detection of specific regions of interest within a DNA sample, thereby facilitating the identification of disease-associated DNA sequences [1].

2.1.2 Polymerase Chain Reaction (PCR) and Multiplexing

Polymerase Chain Reaction (PCR) is a highly effective method used for DNA amplification, relying on a series of thermal cycling steps. The PCR process involves three main stages: (1) denaturation, the double-stranded DNA template being separated into single strands; (2) annealing, the primers binding to each original strand to initiate the synthesis of new DNA strands; and (3) extension, where the DNA strands are extended from the primers using a DNA polymerase enzyme [8]. The denaturation step in PCR requires an increase in temperature to separate the DNA strands into single strands, which is why it is referred to as a thermal cycling process [9].

Multiplexing is a technique that allows the detection of multiple DNA or RNA targets simultaneously in a single PCR reaction. In the past, multiplex PCR was typically accomplished by employing single-well methods that utilized fluorescent oligonucleotide probes specific to the targets, or by implementing spatial multiplexing, where the separation of the sample allowed for simultaneous amplification of multiple targets [10]. However, previous approaches have limitations in detecting only three to four targets at once.

To overcome these limitations, Krietmann et al. proposed a novel approach in their paper titled "Next-generation molecular diagnostics: Leveraging digital technologies to enhance multiplexing in real-time PCR" [11]. In their study, the authors employed machine learning algorithms to analyze the amplification and melting curves in order to classify multiple nucleic acid targets from a single reaction. This approach enables the detection of a larger number of DNA or RNA sequences in a single PCR reaction, enhancing multiplexing capabilities.

2.1.3 LAMP and amplification curve

To understand the relationship between infectious diseases and their respective amplification curves in LAMP testing, it is important to have a basic knowledge of the epidemiological aspects of disease detection. A typical LAMP process involves several steps, as illustrated in Figure 2.1. The first step is the preparation of the sample, which is the collection of a DNA sample. Then, primers that will react and amplify respective target DNAs are added to the sample. LAMP is an isothermal nucleic acid amplification diagnostic test, which offers a significant advantage over PCR. Unlike PCR, which requires a thermal cycler to continuously adjust reaction temperatures, LAMP can be conducted at a constant temperature. Once the reaction is completed, the reaction can usually be observed by the human eye via fluorescence [12]. A positive result is indicated by a visible change in color, resulting from the LAMP amplification process [13].

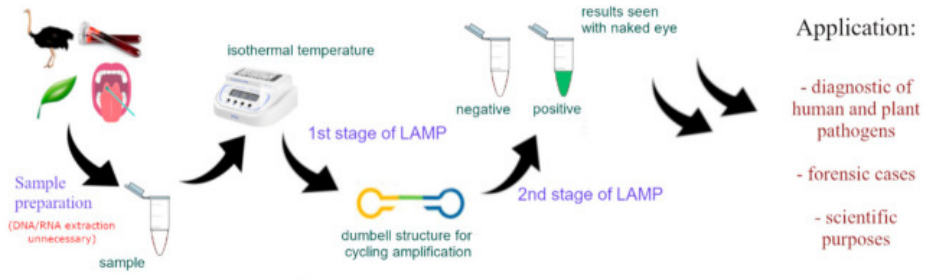


Figure 2.1: LAMP Method Process Diagram from Sampling to Visualisation of Results [13]

However, when utilizing multiplex LAMP to simultaneously detect multiple DNA sequences in a single reaction, the process becomes more intricate. Since any amplification reaction will cause a color change, visually identifying a specific DNA sequence becomes challenging. This is why Amplification Curve Analysis (ACA) is essential for distinguishing the DNA sequence associated with each amplification curve. ACA employs machine learning algorithms to differentiate the amplification curves and classify each curve according to the respective target DNA sequences [14].

Further details on this technique will be discussed in Section 2.2.1.

2.1.4 Melt Curve

Apart from the amplification curve, the melt curve provides valuable information about the target DNA sequence. After the completion of the LAMP reaction, the temperature is incrementally increased, and the fluorescence level is measured at each temperature step. This data allows for the plotting of fluorescence signal against temperature, resulting in the creation of a melt curve. However, it is important to note that the melt curve requires an additional step beyond the amplification reaction, making it a more time-consuming process compared to the amplification curve. Additionally, the incremental temperature changes used in the melt curve analysis eliminate the isothermal advantage of LAMP experiments overall [15].

2.1.5 quantitative LAMP (qLAMP) and digital LAMP (dLAMP)

There are two different types of LAMP data dealt with throughout the project. The first set of data is the qLAMP data. qLAMP is a variation of LAMP that incorporates real-time fluorescence detection to generate an amplification curve. It enables the quantification of the target nucleic acid sequence during the amplification process by utilizing a fluorescent dye that binds to the amplified product. The increase in fluorescence intensity corresponds to the accumulation of the target sequence

The second set of data used was the dLAMP data. dLAMP is a digital version of LAMP that involves the partitioning of the target droplet into small wells or compartments before amplification. Each droplet acts as an independent reaction compartment and contains all the necessary components, including primers. Similar to qLAMP, the droplets undergo isothermal amplification, resulting in the amplification of the target sequence. However, unlike qLAMP, dLAMP generates multiple amplification curves because each droplet or well produces an individual reaction [16].

An advantage of dLAMP over qLAMP is that it allows for quantification based on the frequency of positive droplets. By analyzing the number of droplets that show amplification and comparing it to the total number of droplets, the concentration or abundance of the target sequence can be estimated.

As a larger amount of dLAMP data was available, the original transformer model was developed and optimized using dLAMP data, capturing the ideal characteristics of the amplification curve. In the future, qLAMP data could be utilized to fine-tune the model through transfer learning, enhancing its performance for other practical applications.

2.2 Deep Learning

2.2.1 Baseline Models

The most recent model to be compared as a baseline model throughout the report is a paper published in 2022, "Single-channel digital LAMP multiplexing using amplification curve analysis" by Malpartida-Cardenas et al [7]. In this paper, the authors attempt to use a machine learning based approach to analyse and classify the different amplification curves into the respective targets as illustrated in Figure 2.2. The data used is a 5-plex LAMP data, which consists of five respiratory pathogens: human influenza A virus (IAV), human influenza B virus (IBV), severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), human adenovirus (hAdV), and Klebsiella pneumoniae (KP) [17, 18, 19].

In the paper "Amplification Curve Analysis: Data-Driven Multiplexing Using Real-Time Digital PCR," Moniri et al. introduced a technique called Amplification Curve Analysis (ACA) [14]. The authors demonstrated the application of ACA on polymerase chain reaction (PCR) amplification curves, utilizing five specific features extracted from the curves instead of considering the entire curve. The authors employed a five-variable sigmoid regression model, which generated five distinct features representing the structure of the amplification curve for each amplification event. These features were then utilized in the k-Nearest Neighbor (k-NN) clustering algorithm for further analysis and classification.

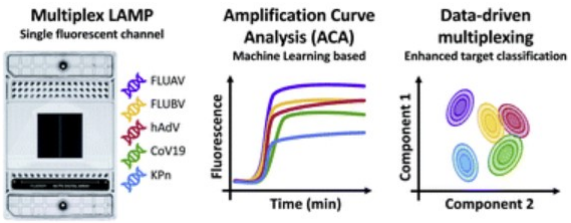


Figure 2.2: Steps for multiplex LAMP by Malpartida-Cardenas et al [7]

A total of 110,880 amplification events were generated, out of which 54,186 were positive amplification reactions. On average, each target received between 6,000 to 14,000 positive amplification events. The researchers employed classification and clustering techniques on the real-time data and used dimensionality reduction to visualize the clusters formed for each target in a 3D space. The clusters can be observed in the Figure 2.3, where each target appears to be assigned to a specific cluster, notably in the case of IBV and hAdV. For classification purposes, the authors chose the k-Nearest Neighbor (k-NN) algorithm with a parameter k set to 10. This choice resulted in a classification accuracy of 91.33% $\pm$ 0.33% (mean $\pm$ std) as reported in the paper.

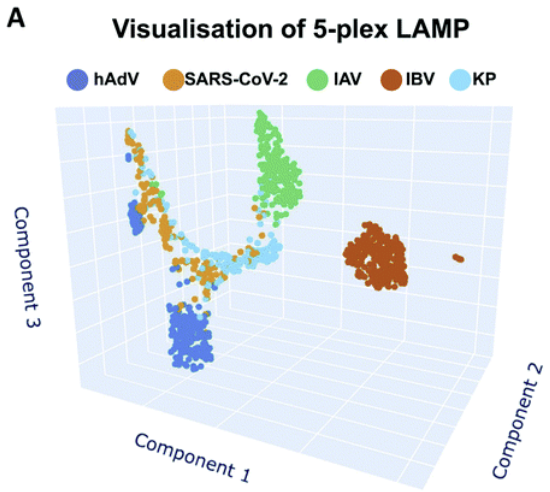


Figure 2.3: Visualisation of the similarity of real-time LAMP amplification curves using the uniform manifold approximation and projection algorithm by Malpartida-Cardenas et al [7]

2.2.2 Transformers

Transformers are a rising deep learning model that is extremely efficient in natural language processing and processing time-series data. As the amplification curve is a plot of cycles over time against the fluorescence intensity level, it can be considered time-series data, and hence using a transformer can be deemed efficient. Unlike traditional recurrent neural networks (RNNs), which are designed to process data sequentially, transformers utilize a self-attention mechanism to compute contextualized representations of each input element. This allows the model to assign varying levels of importance to different elements and features based on their relationships with one another, enabling transformers to capture complex temporal dependencies [20]. Multiple layers of self-attention and feed-forward neural networks can be stacked in transformers, allowing them to effectively process and extract information from time series data. This hierarchical architecture allows the transformer to capture intricate patterns, granting accurate predictions and meaningful insights for time-series data.

There are many features to consider for the transformer to perform classification on sequential data, listed below [20, 21, 22, 23]:

1. Attention Mechanism: Transformers rely on self-attention mechanisms to capture dependencies between different time steps. It must be ensured that the attention mechanism is designed to capture long-range dependencies effectively in the time sequence data.

2. Input Representation: Representing time sequence data as input to the transformer is crucial. Techniques like tokenization or continuous embeddings can be used to encode the temporal information effectively. For example, representing each time step as a token or using position encodings to preserve temporal order can be used.

3. Sequence Length: Transformers are known to have high memory requirements due to their self-attention mechanism. Considering the sequence length of the time series data and eval-

uate if it fits within the memory limitations of the hardware or the model architecture is necessary. If the sequence is too long, it may be necessary to truncate or down-sample the data.

4. Model Architecture: Transformer models come in different sizes and variants. Consider the architecture's depth (number of layers), width (hidden units or dimensions), and the number of attention heads.

5. Regularization Techniques: Transformers, like other deep learning models, can be prone to overfitting, especially with limited data. Regularization techniques like dropout, weight decay, or layer normalization can help prevent overfitting and improve generalization.

6. Task-Specific Modifications: Depending on the specifics of the time sequence classification task, task-specific modifications to the transformer model may be necessary. This can include adding additional layers, modifying the loss function, or incorporating domain-specific knowledge.

7. Training and Optimization: Transformers are typically trained using stochastic gradient descent (SGD) or its variants. Consider the learning rate, batch size, optimizer choice, and the number of training iterations. It may also be necessary to explore learning rate schedules or techniques like early stopping to improve convergence.

8. Evaluation Metrics: Defining appropriate evaluation metrics for the time sequence classification task is crucial. Accuracy, precision, recall, F1 score, or area under the receiver operating characteristic curve (AUC-ROC) are common metrics used for classification tasks. Metrics that align with the task requirements must be selected.

2.3 Section Summary

This section provided the necessary background knowledge to comprehend the methodologies employed in the project. It encompassed the utilization of qLAMP and dLAMP techniques for DNA sequence amplification, as well as an overview of previous approaches outlined in existing literature. Additionally, state-of-the-art deep learning algorithms and techniques were introduced, which were utilized to enhance and refine the existing models. Building upon this foundation, the subsequent section will discuss the project's desired requirements and objectives to be achieved by its completion.

Chapter 3

Requirements Capture

Section 3 of the project encompasses the technical requirements and the expected minimum performance criteria for the final outcomes. The primary objective is to surpass the performance achieved in the previous work conducted by Malpartida-Cardenas et al. in 2022 [7] by employing state-of-the-art deep learning algorithms. The project aims to explore various transformer architectures to identify the most effective ones that yield superior results. Additionally, if time permits, the project intends to enhance the performance and accuracy by incorporating melting curve data. This approach will enable the model to leverage knowledge gained from pre-existing datasets and apply it to new datasets, thereby improving the overall performance of the system.

3.1 Technical Requirements

Similar to the previous research stated in Malpartida-Cardenas et al. in 2022, the final output is a deep learning model which will take in an input amplification curve from a qLAMP experiment, and accurately define what target class the curve is to belong in. For the technical requirements to be met, the following steps must be taken.

- Data Pre-processing
  - Pre-process and refine the given multiplex LAMP data to remove outliers and standardise the curves.
- Literature Research
  - Conduct literature research on LAMP and its procedures for multiplex LAMP.
  - Explore literature on amplification curves and melting curves used in LAMP.
- Deep Learning Research
  - Research k-fold cross-validation and its usages for limited data.
  - Investigate simple machine learning algorithms like KNN, random forest, regularization methods, and others.

– Explore the application of transformers in time series data classification tasks.

3.2 Performance Requirements

As mentioned earlier, the project's performance requirements aim to surpass the accuracy of the existing model. In the referenced paper [7], a classification accuracy of 91.33% was reported on a dataset consisting of 54,186 positive amplification events. The final product will be evaluated using the same dataset for a fair comparison. Figure 3.1 displays the confusion matrix obtained by employing amplification curve analysis and the k-NN algorithm to evaluate the data.

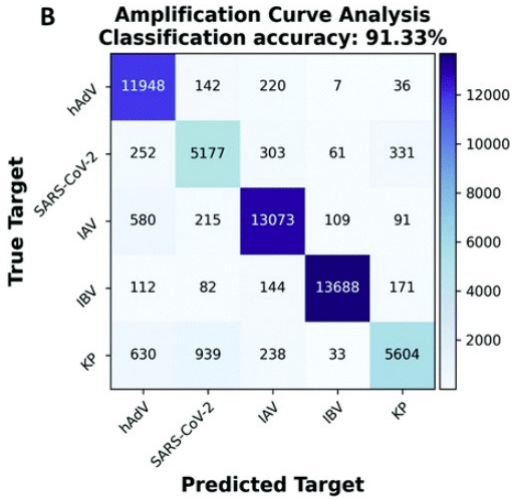


Figure 3.1: Performance of model developed by Malpartida-Cardenas et al [7]

The performance of the project will be assessed based on similar accuracy metrics and compared against the performance of the ACA (Amplification Curve Analysis) model.

3.3 Section Summary

To summarize, the following steps must be taken in order to complete the project aims:

1. Perform data pre-processing by eliminating outliers and normalizing the amplification curve to eliminate discrepancies arising from varying experimental environments.
2. Gain insight into the current model's structure and methodologies employed for solving the multiplex LAMP classification problem.
    - Recreate the model and conduct testing using new data to determine the desired minimum accuracy on baseline models.
3. Design a Transformer model that leverages the activation curve to classify data into the target class.
    - Explore various architectures and hyperparameters to identify the configuration that yields the highest performance.

4. Attempt to use melt curve data as an alternative option to the amplification curve.

5. Compare the performance of the new transformer model using the activation curve or melt curve against the previous baseline models to evaluate the success of the project.

The following section will outline the design and decision-making process undertaken throughout the project, as well as provide an overview of the final product.

Chapter 4

Analysis and Design

Based on the project specifications, background information, and final goal, this section will outline the foundational design of the project and the engineering decisions made to realize this design. It will provide a high-level overview of the project's design and subsequently delve into the three main components in greater detail.

4.1 High-level Overview

The development of the Transformer model for multiplex LAMP can be largely divided into three main parts.

In the first part, the objective is to replicate the baseline models and evaluate their performance using the provided data. This step is crucial for accurate comparison with the Transformer model. Initially, the k-NN model developed by Malpartida-Cardenas et al. [7] will be emulated by implementing the k-NN algorithm with similar settings and parameters as described in the referenced paper. The performance of the emulated k-NN model will be evaluated using the available data. Additionally, other machine learning models such as random forest, L1 regularization, L2 regularization, L1 and L2 regularization, stochastic gradient descent (SGD), gradient boosting (GBT), and support vector machines (SVM) will be explored for comparison. Each model will be trained and evaluated using the provided data, and performance metrics such as accuracy, precision, recall, and F1-score will be calculated. The best-performing baseline model among the tested machine learning models will be selected for comparison against the Transformer model.

The second part involves training the Transformer model on the dLAMP data to determine the optimal hyperparameters and architecture for achieving the highest performance. Different combinations of hyperparameters and architectures will be experimented with, including varying numbers of layers, attention heads, hidden units, and other relevant parameters. The goal is to identify the configuration that produces the best results on the dLAMP dataset.

The final part of the development process focuses on acquiring dLAMP melt curve data. Similar procedures used for the amplification curve will be followed for the melt curve. Starting with

the baseline models for machine learning algorithms, the Transformer model will be employed to further enhance the accuracy. The possibility of incorporating the melt curve in combination with the amplification curve will be explored based on the performance of the two models.

By dividing the development process into these three parts, the model can be trained and optimized initially on the dLAMP amplification dataset, with the potential to incorporate the melt curve data as an additional dataset, depending on the performance of the models.

4.2 Baseline Models

In this section, a further explanation of the baseline model and its comparison to the project will be provided. Thorough testing will be conducted on different models, evaluating various hyperparameters. This testing will involve adjusting the parameters of the model, such as the variable 'k' in KNN, as well as determining the appropriate type of data to be used. Only the models that yield the best performance will be used for comparison against the final transformer model. The list below outlines the different models to be tested and their corresponding hyperparameters:

- Random Forest Classifier: maximum depth, number of trees, minimum samples per leaf, minimum samples for a split
- k-Nearest Neighbors (kNN) Classifier: number of neighbors (k), distance metric, weighting scheme
- L1 Regularisation (LR_L1): regularization strength (alpha), learning rate, number of iterations/convergence tolerance
- L2 Regularisation (LR_L2): regularization strength (alpha), learning rate, number of iterations/convergence tolerance
- L1 and L2 Regularisation (LR_L1_L2): regularization strength (alpha), learning rate, number of iterations/convergence tolerance
- Stochastic Gradient Descent (SGD): learning rate, regularization strength, loss function, penalty, number of iterations/convergence tolerance
- Gradient Boosting (GBT): number of boosting stages, learning rate, maximum depth, subsample ratio
- Support Vector Machine (SVM): kernel type, regularisation parameter (C), kernel coefficient (gamma)

Moreover, the input variables for the machine learning tasks can be broadly categorized into three parts. Firstly, the original amplification curve itself can be utilized as a feature. Additionally, the five parameters obtained from the sigmoid curve fitting of the amplification curve, as described in Miglietta et al. [24], can also serve as features. Therefore, all three combinations, namely the

amplification curve alone, the five parameters alone, and the combination of the amplification curve with the five parameters, were tested separately as input features for the machine learning models.

4.3 Transformers

In contrast to the baseline models, the input data for the Transformer model will solely consist of the amplification curve. This decision is based on the understanding that Transformer models excel in processing sequential data rather than fixed features. As the five parameters derived from the sigmoid curve fitting do not possess a sequential nature, they are not included as input for the Transformer model. By focusing solely on the amplification curve, the Transformer model can leverage its strength in capturing dependencies within sequential data to achieve optimal performance.

The hyperparameters of a transformer begins with the design of the architecture of the transformer. The following list provides the hyperparameters that are to be tested, and the significance of each parameter in the context of time series classification tasks [21, 22].

1. Number of Layers: The transformer model is composed of multiple layers of self-attention and feed-forward neural networks. The depth of the model, which is determined by the number of layers, plays a crucial role in its capacity to comprehend complex patterns. In general, deeper models have the ability to capture more intricate dependencies, although they may come at the cost of increased computational complexity.

2. Number of Attention Heads: The self-attention mechanism is divided into multiple attention heads, with each head focusing on different aspects of the input. The number of attention heads directly influences the model's ability to capture diverse patterns and relationships. By increasing the number of attention heads, the model's expressiveness can be enhanced. However, it's important to note that this increase in expressiveness may come at the cost of additional computational resources required.

3. Hidden Size/Dimension: This hyperparameter determines the dimensionality of the hidden states in the transformer model. It defines the model's capacity to learn and represent information. Higher hidden size allows the model to capture more complex relationships but increases computational requirements.

4. Dropout Rate: Dropout is a regularization technique employed to mitigate over-fitting by randomly zeroing out a portion of the model's outputs during training. Its purpose is to enhance generalization capabilities. The dropout rate dictates the proportion of units that are dropped out, and selecting an appropriate rate is crucial and depends on factors such as the dataset and model complexity.

5. Learning Rate: The learning rate controls the step size in the optimization process. It determines how quickly the model learns from the training data. A high learning rate may result

in unstable training or overshooting, while a low learning rate may lead to slow convergence. Finding an optimal learning rate often requires experimentation.

6. Batch Size: The batch size refers to the number of samples processed in each training iteration. Increasing the batch size can enhance training efficiency; however, it also leads to higher memory demands. On the other hand, smaller batch sizes may result in noisy gradients during training. The choice of batch size depends on various factors, including the available computational resources and the characteristics of the dataset being used.

7. Weight Decay: Weight decay, also known as L2 regularization, adds a penalty term to the loss function that discourages large weights. It helps prevent over-fitting by promoting weight values closer to zero. The weight decay hyperparameter controls the strength of the regularization effect.

8. Attention Dropout: In addition to standard dropout, attention dropout can be applied specifically to the attention weights during the self-attention mechanism. It helps regularize the attention patterns and improve generalization.

9. Maximum Sequence Length: This hyperparameter sets the maximum length allowed for input sequences. It determines the memory requirements and computational constraints of the transformer model. Sequences longer than this length will need to be truncated or segmented.

4.4 Melt Curve

The melt curve analysis will be conducted following a similar procedure to that of the amplification curve, as described earlier. However, there will be a distinction in the data used and the features employed for the simple machine learning models. Instead of employing a five-parameter sigmoidal fitting, the focus will be on the melt peak. The melt peak refers to the temperature at which the maximum value of the derivative of the melt curve is observed. Since the provided melt curve data already represents a plot of temperature against $-\frac{dFluorescence}{dTemperature}$, the melt peak can be identified as the point where the y-value is highest.

To evaluate the baseline models, three different approaches will be employed: utilizing only the melt peak, using only the melt curve, or combining both. The transformer model, on the other hand, will be exclusively tested using the melt curve, which aligns with the rationale presented for the amplification curve analysis.

Chapter 5

Implementation

The previous section provided an overview of the design choices for the project. This section focuses on illustrating the implementation details and adjustments made to incorporate these design choices into the actual project deliverable. The goal of the implementation section is to successfully develop and deploy an efficient transformer model, which will be compared to previous baseline models to demonstrate its success.

5.1 Data Pre-processing & Analysis

The first step to the data was implementing the data pre-processing section to refine and filter out noisy data.

5.1.1 Amplification Curve Data for Training Model

To provide a summary of the original data, it consisted of a 5plex-LAMP assay performed on a digital real-time instrument called dLAMP [7]. The data included six distinct labels: human influenza A virus (IA), human influenza B virus (IB), severe acute respiratory syndrome coronavirus 2 (C19), human adenovirus (AD), Klebsiella pneumoniae (KP), and non-template control (NTC). The dataset comprised a total of 110,880 amplification events, out of which 54,186 were positive amplification reactions. Each target had approximately 6,000 to 14,000 positive amplification events. The amplification curves for all targets are depicted in Figure 5.1.

As evident in Figure 5.1, the original data consisted of numerous noisy curves, necessitating preprocessing steps prior to making them suitable for machine learning classifiers.

To start, the curves were normalized by adjusting their fluorescence levels so that the initial fluorescence values were the same. This step was necessary because different reactions could have varying amounts of starting target DNA sequences, resulting in different fluorescence levels at the beginning. This variation is clearly observable in Figure 5.1, where most curves exhibit two distinct baseline fluorescence levels, likely due to experiments conducted on different dates. To address this issue, each curve was normalized using the average of all the curves combined.

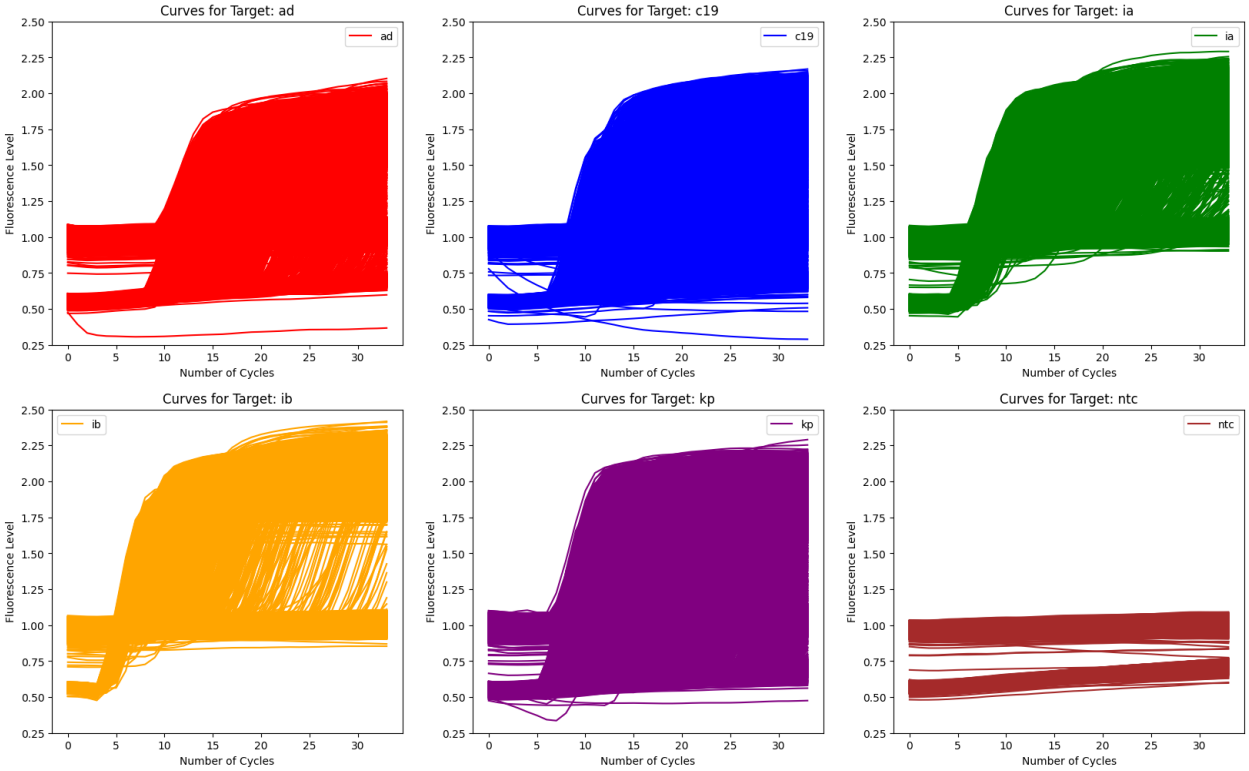


Figure 5.1: Plot of Unprocessed Amplification Curve by Target

Another problem with the original plot was the inclusion of amplification curves that extended beyond 35 cycles, the maximum specified cycles. These incomplete amplification curves needed to be eliminated as they could introduce noise during the training process. To address this, all curves that did not surpass a fluorescence intensity threshold of 0.2 by cycle 20 were removed.

After undergoing the aforementioned data pre-processing steps, a total of 50,456 curves remained in the dataset. Out of these, 11,576 curves corresponded to AD, 13,847 to IA, 14,151 to IB, 6,457 to KP, and 4,425 to C19. These remaining curves were subsequently re-plotted, and Figure 5.2 illustrates the resulting visualization.

As part of the exclusion criteria, the NTC (non-template control) curves were not considered for the project. These curves did not contribute to the classification task's training or testing phases. Instead, they displayed flat lines consistently fixed at 0 fluorescence.

The Python code for the preprocessing part of the project has been included in the appendix and can be accessed through the provided GitHub repository link.

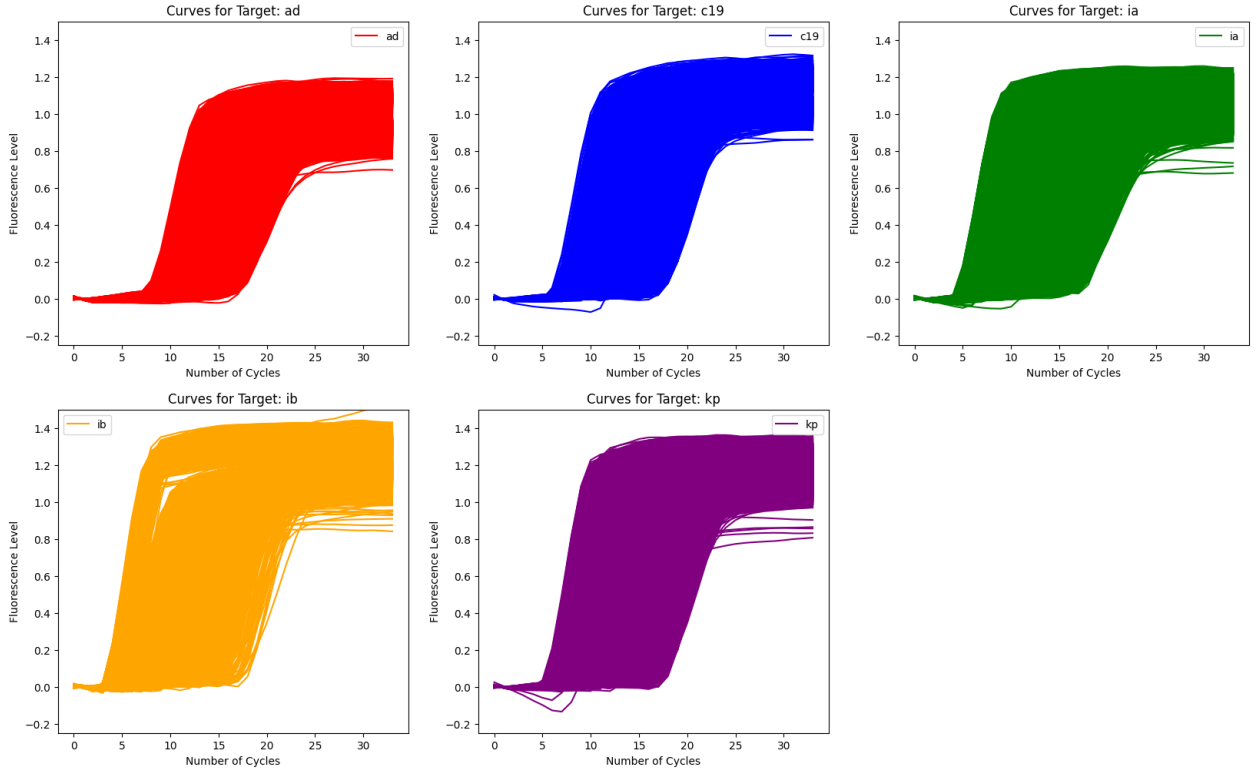


Figure 5.2: Plot of Pre-processed Amplification Curve by Target

5.1.2 Melt Curve Data

In addition to the amplification curve data, a separate model was trained using the melt curves. The melt curves provided a plot of the temperature increase against the slope of the original melting curve, represented as $-\frac{dFluorescence}{dTemperature}$. Before undergoing pre-processing steps, these melt curves were plotted as shown in Figure 5.3. Although the plotted melt curves clearly demonstrate distinct patterns, there were noticeable outliers throughout. For example, in the case of the target "ib," there is a melt curve with a late peak occurring around 90 degrees, deviating from the typical peak at 85 degrees observed in the other curves.

The melt curves were pre-processed using similar steps to the amplification curve.

The melt curves, after undergoing the pre-processing steps, are depicted in Figure 5.4. In comparison to Figure 5.3, it is evident that the melt curves for each target have been transformed into similar shapes through the pre-processing steps. Additionally, the outliers have been effectively addressed, and the baseline fluorescence levels appear to be more stable across all curves. From the melt curves, a notable characteristic is the discrepancy in fluorescence levels among different targets. For instance, the curves corresponding to the target "ib" exhibit significantly higher peak fluorescence compared to the other targets.

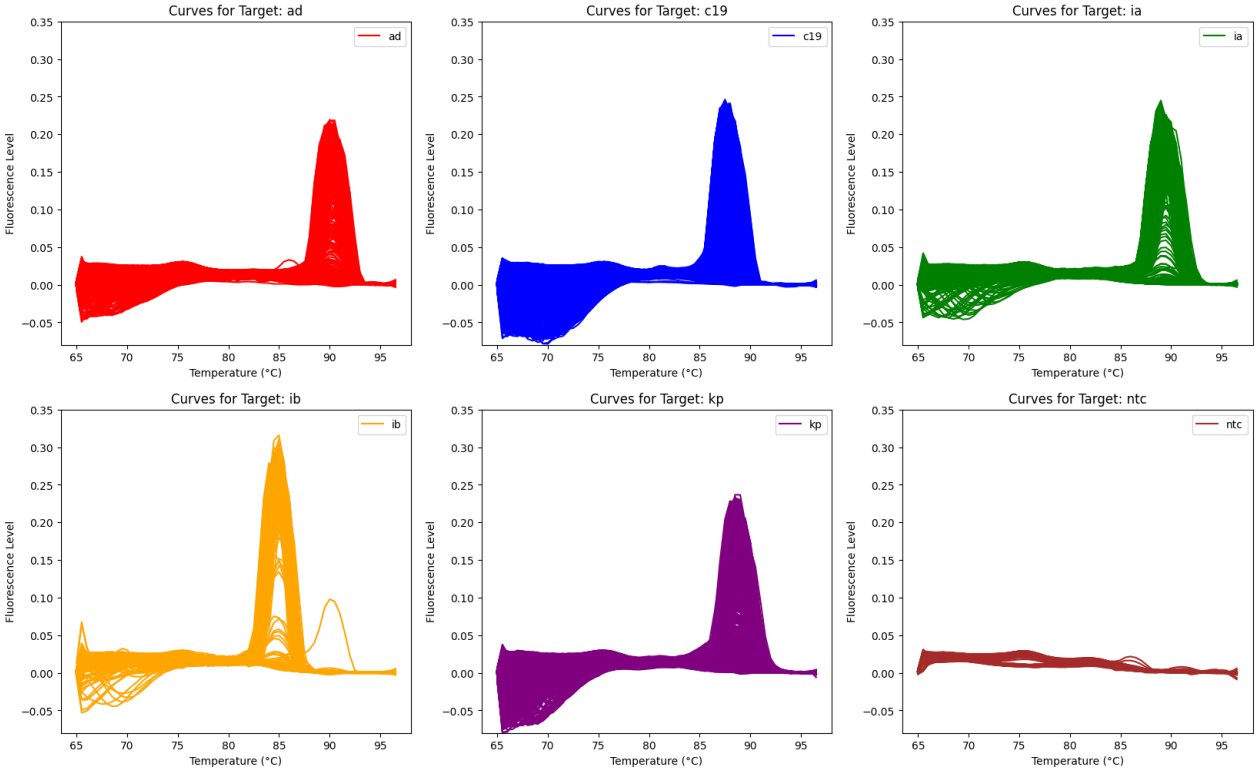


Figure 5.3: Plot of Unprocessed Melting Curve by Target

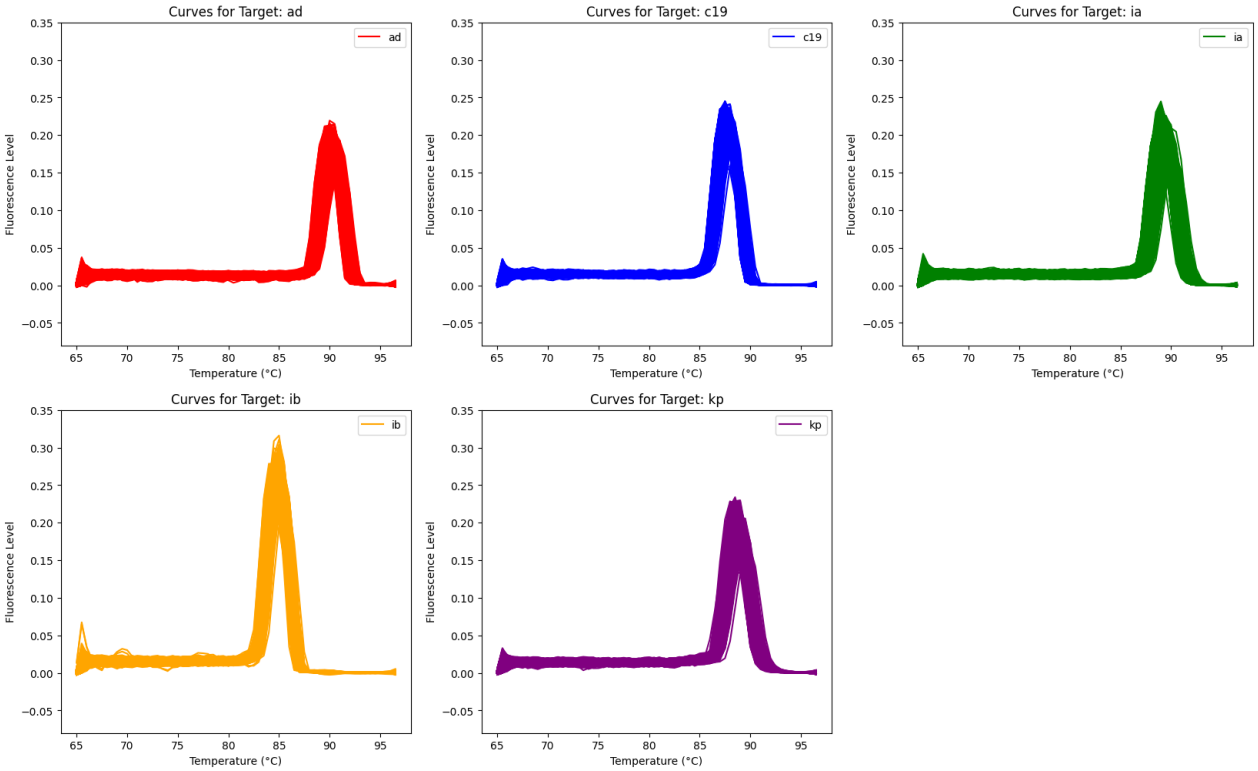


Figure 5.4: Plot of Pre-processed Melting Curve by Target

5.2 Baseline Models Implementation

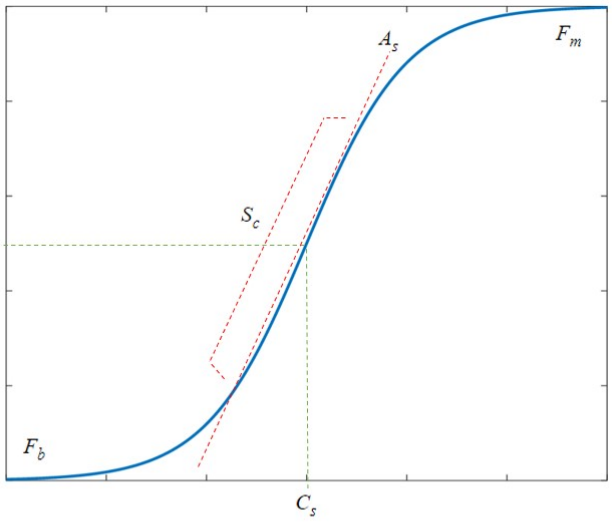
Comparing the performance of a new model or approach against baseline models is essential for evaluating its success. In this context, the published baseline model for comparison was the k-NN model proposed by Malpartida-Cardenas et al. [7]. Furthermore, other machine learning classifiers were also employed in the project to serve as additional baselines for comparison.

5.2.1 Input Data

The input data for the baseline models was categorized into three distinct types. The first category consisted of the five parameters obtained from the sigmoidal fitting process. The second category comprised the complete amplification curve data. Lastly, the third category involved combining the five parameters with the amplification curve. Each model was tested on all three input types, and their accuracy scores were compared to determine the model with the highest performance. However, in the case of the transformer model, only the amplification curve was utilized. This decision was based on the fact that the transformer model is specifically designed to process time-series data; therefore, the amplification curve provided the most relevant and reasonable input for the transformer model's architecture and objectives. The melt curve was processed separately, and will be further discussed in Section 5.4.

5 Parameter Sigmoidal Fitting

To generate additional features in addition to the amplification curve, the 5-parameter sigmoid fitting was applied to the amplification curve. This process resulted in the extraction of the additional features, namely F_b, C_s, S_c, A_s, and F_m. These features can be observed in Figure 5.5, which visually represents the target features to be extracted from the curve.



$$f(t,p) = \frac{F_m}{\left(1 + e^{-S_c(t-C_s)}\right)^{A_s}} + F_b$$

$$p = \left[F_m, F_b, S_c, C_s, A_s\right]^T$$

Figure 5.5: Features from 5 Parameter Sigmoid Function

The 5-parameter sigmoidal fitting was implemented by fitting the given amplification curve to the function shown in Figure 5.2.1. This fitting process utilized the curve_fit function from the SciPy optimization and root finding library [25]. A crucial aspect of the fitting process was defining appropriate bounds, which were set as illustrated in the function depicted in Figure 5.2.1. By considering the characteristics of the sigmoidal function for a typical amplification curve, the maximum and minimum bounds for each parameter (F_m, F_b, S_c, C_s, A_s) could be determined, facilitating the fitting process. To visualize the success of the fitting process, a sample of the fitted curve is presented in Figure 5.7. It is evident from the figure that the curve successfully represents the fitted data points, indicating a good fit.

```
def sigmoid_func(t, F_m, F_b, S_c, C_s, A_s):
    return F_m/(1+np.power(np.exp(-S_c*(t-C_s)), A_s))+F_b

curve_fit(sigmoid_func, range(1, 36), targets[i],
    p0=[1,0,0.2,15,1], bounds = ([0,-0.2,0,0,0],[2,0.2,0.5,35,10]))
```

Figure 5.6: Code for Sigmoid Function 5 Parameter Curve Fitting

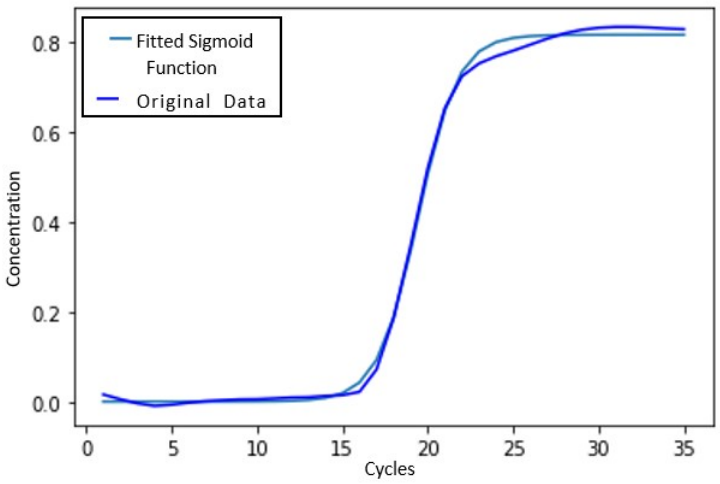


Figure 5.7: Sample Fitting of AD Data on Sigmoid Function with Parameters F_m=0.942, F_b = 0.001, S_c = 0.253, C_s = 19.254, and A_s = 2.611

5.2.2 Simple Machine Learning Models

After completing the pre-processing and feature extraction stages, the implementation of baseline models was carried out. In addition to the k-NN model proposed by Malpartida-Cardenas et al. [7], several other models, including the random forest classifier, k-Nearest Neighbors (kNN) classifier, L1 regularization, L2 regularization, L1 and L2 regularization, stochastic gradient descent, gradient boosting, and support vector machines, were tested. This allowed for a comprehensive comparison with the transformer model.

To ensure a thorough evaluation, each model was tested three times using three different types of input features. Furthermore, multiple rounds of testing were conducted with different hyper-parameters in order to identify the optimal configuration that produced the best performance. This rigorous testing approach aimed to select the machine learning model that demonstrated the highest level of performance for the given task.

5.3 Transformer Model Implementation

The transformer model was implemented using PyTorch, a Python library specifically designed for developing deep learning algorithms. During the design of the model, reference was made to the paper "Attention is All You Need" by Vaswani et al [20]. The model is illustrated in Figure 5.8. Additionally, the complete implementation code for the model, as well as the iterative search process for determining the optimal hyperparameters, can be found in the appendix.

5.3.1 Hyperparameter Search

Despite the significant computational resources, RAM, and time required to test each transformer model on the given data, the project made every effort to conduct a thorough analysis. To explore

```
class TransformerModel(nn.Module):
  def __init__(self, input_size, d_model, nhead, num_layers,
    output_size, max_len):
    super(TransformerModel, self).__init__()
    self.embedding = nn.Linear(input_size, d_model)
    self.pos_encoder = PositionalEncoding(d_model, max_len=max_len)
    self.transformer_layer = TransformerEncoderLayer(d_model, nhead)
    self.transformer_encoder = TransformerEncoder(self.transformer_layer
    , num_layers)
    self.fc = nn.Linear(d_model, output_size)

  def forward(self, x):
    x = self.embedding(x)
    x = self.pos_encoder(x)
    x = self.transformer_encoder(x)
    x = self.fc(x[:, -1, :])
    return x

class PositionalEncoding(nn.Module):
  def __init__(self, d_model, dropout=0.1, max_len=1000):
    super(PositionalEncoding, self).__init__()
    self.dropout = nn.Dropout(p=dropout)

    pe = torch.zeros(max_len, d_model)
    position = torch.arange(0, max_len, dtype=torch.float).unsqueeze(1)
    div_term = torch.exp(torch.arange(0, d_model, 2).float()
        * (-np.log(10000.0) / d_model))
    pe[:, 0::2] = torch.sin(position * div_term)
    pe[:, 1::2] = torch.cos(position * div_term)
    pe = pe.unsqueeze(0).transpose(0, 1)
    self.register_buffer('pe', pe)

  def forward(self, x):
    x = x + self.pe[:x.size(0), :]
    return self.dropout(x)
```

Figure 5.8: Implementation of Transformer Model using PyTorch from PyTorch Tutorial [26]

a wide range of hyperparameters, the Python library itertools was utilized [27]. The specific hyperparameters that were searched through are provided in Figure 5.9. This approach allowed for an extensive exploration of different combinations of hyperparameters to identify the optimal configuration for the transformer model. The following hyperparameters were tested:

- d_models: This variable represents the dimensionality or size of the model's hidden layers. This determines the capacity of the model and can affect its ability to capture complex patterns.

- nheads: The attention heads in the transformer model determine the number of parallel attention mechanisms that can focus on different parts of the input simultaneously. By using multiple attention heads, the model gains the capability to capture diverse dependencies and relationships within the input. This parallel processing improves the model's ability to understand and incorporate various aspects of the input data effectively.

- num_layers: Indicates the number of layers in the transformer model. Each layer consists of self-attention and feed-forward neural network modules. More layers can potentially enable the model to capture more intricate patterns and relationships in the data.
- lr: Learning rate refers to the step size at which the parameters of a model are updated during the training process. A higher learning rate can lead to faster convergence, allowing the model to reach an optimal solution more quickly. However, using an excessively high learning rate can result in instability or overshooting, causing the model to struggle in finding the optimal parameter values. Therefore, it is essential to strike a balance and choose an appropriate learning rate that promotes stable and efficient training.
- batch_size: Batch size refers to the quantity of input samples handled simultaneously during the training process. Enlarging the batch size can accelerate training, yet it necessitates greater memory resources.
- drop_out: Denotes the probability of randomly dropping out or deactivating individual neurons during training. Dropout is a regularization technique that can help prevent overfitting and improve generalisation.

```
d_models = [32, 64, 128]
nheads = [2, 4, 8]
num_layers = [2, 3, 4]
lr = [0.001, 0.0001, 0.00001]
batch_size = [32, 64]
drop_out = [0, 0.1]
epochs = [150, 200]
```

Figure 5.9: Hyperparameters Iterated for Transformer Model

The model with the best performance was the model with the parameters d_model=128, nhead=4, num_layers=3, lr=0.001, batch_size=64, and drop_out=0.1. This model produced a validation accuracy of 94.53%. Further results are to be discussed in the following testing and results section.

5.3.2 Positional Encoding

As the amplification curve is a time series data the temporal order of data is very significant. As standard transformers are not designed to understand the order of inputs, positional encoding was necessary [20]. This implementation is given in Figure 5.8.

5.4 Melt Curve Implementation

After successfully implementing simple machine learning models and the transformer model on the amplification curve data, the analysis progressed to utilizing the melt curve data. However,

unlike the amplification curve analysis that employed a 5-parameter sigmoidal fitting, the focus of the melt curve analysis was mainly on a distinct feature called the melt peak. This melt peak corresponds to the temperature at which the highest value of the derivative of the melt curve occurs.

To extract the melt peak as the primary input feature, the given curve, which was already plotted as Temperature against $-\frac{dFluorescence}{dTemperature}$, was examined. By finding the maximum value from this curve, the corresponding temperature represented the melt peak. This temperature value was then utilized as a crucial input feature for the subsequent analysis.

5.4.1 Simple Machine Learning Models for Melt curve

In the case of the melt curve, the input data for the machine learning models was divided into three categories: melt peak only, melt curve only, and melt peak + melt curve. For the melt peak only input, several adjustments were made to the machine learning model. This was necessary because the model had to be modified to accommodate the use of a single input feature. Apart from this, the same models used for the amplification curve analysis were also applied to the melt curve data. These models included the random forest classifier, k-Nearest Neighbors (kNN) classifier, L1 regularization, L2 regularization, L1 and L2 regularization, stochastic gradient descent, gradient boosting, and support vector machines. For each model, multiple parameters were tested, and the models that exhibited the highest accuracy were selected for comparison in the results section.

5.4.2 Transformer Model for Melt Curve

The same model and implementation approach were applied to the transformer for the melting curve data, with one notable difference being the length of the input data. While the original input sequence length for the amplification curve was 35, the melting curve had 64 data points. The model structure proposed in Figure 5.8 was retained, but the architecture of the model was adjusted due to different hyperparameters, leading to improved results for the melting curve.

For the melting curve analysis, various hyperparameters shown in Figure 5.10 were tested to determine their impact on performance. After rigorous evaluation, the model with the following hyperparameters emerged as the best performer: d_model=128, nhead=8, num_layers=3, lr=0.0001, batch_size=64, and drop_out=0.1. This model demonstrated exceptional performance, achieving a remarkable validation accuracy of 99.97%. Its detailed analysis and results will be further discussed in the testing and results section of the report.

```
d_models = [64, 128]
nheads = [2, 4, 8]
num_layers = [2, 3, 4]
lr = [0.001, 0.0001, 0.00001]
batch_size = [32, 64, 128]
drop_out = [0.1]
epochs = [150]
```

Figure 5.10: Hyperparameters Iterated for Transformer Model

5.5 Section Summary

This section highlighted the details involved in implementing different types of models on different types of data. It began by guiding through the preliminary steps, such as data pre-processing and baseline model implementation. Then, it went on to explain the implementation of the final transformer model for both the amplification curve and the melt curve data. The following section will discuss the results of these implementations.

Chapter 6

Testing and Results

This section focuses on the testing phase of the model and evaluates its performance. Additionally, it presents the results obtained from various models, including the baseline model comprising simple machine learning models, as well as the final transformer model.

6.1 5-fold Cross-Validation

K-fold cross-validation is a valuable technique employed to predict outcomes for unseen data, especially when dealing with limited data samples [28]. It serves as a popular choice for reducing bias in the evaluation step. For this project, 5-fold cross validation approach was used. Initially, 20% of the original dataset was reserved for future testing. The remaining 80% was divided into five separate parts called folds. In each iteration, one fold was used as the validation set, while the remaining four folds were combined to form the training set. This process was repeated five times, with each fold serving as the validation set once. The model that achieved the highest accuracy on the validation set was selected as the ultimate model. To evaluate its performance, this model was tested against the initially split-out 20% test dataset. By employing this method, the optimal combination of training data inputs that yielded the best performance for the model could be found.

6.2 Testing Baseline Models

The accuracy-based performance of each model is depicted in Figure 6.1. The figure showcases eight distinct machine learning models, each utilizing one of three input types: 5-parameter only, amplification curve only, or a combination of 5-parameter and amplification curve. To enhance clarity, the accuracy values are explicitly displayed on top of their respective bars. The figure highlights that, overall, the random forest algorithm achieved the highest accuracy value across all three input types. Furthermore, the combination of 5-parameter and amplification curve input consistently demonstrated the best performance across all machine learning models. Notably, the

combination of the random forest classifier and the combination of 5-parameter and amplification curve input yielded the highest accuracy of 93.69%.

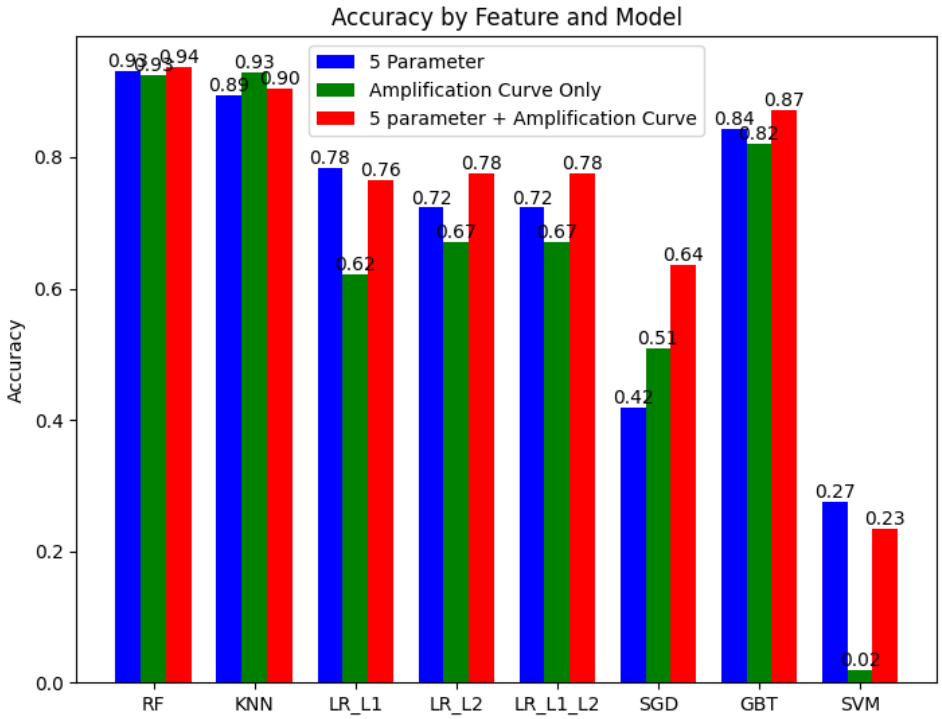


Figure 6.1: Accuracy of Baseline Models with Different Input Data on Test Data Set

The subsequent confusion matrix depicted in Figure 6.2 represents the top-performing machine learning model on the amplification curve data. The input data comprised the 5 parameter combined with the amplification curve, and the machine learning model employed was the random forest classifier: RandomForestClassifier(n_estimators=10, criterion='entropy'). Moreover, Table 6.1 presents the classification report for this model, encompassing key metrics such as precision, recall, and the F1-score.

Class	Precision	Recall	F1-Score	Support
AD	0.95	0.98	0.96	1691
IA	0.95	0.96	0.96	2272
IB	0.97	0.98	0.97	1634
KP	0.92	0.92	0.92	1778
C19	0.87	0.77	0.82	1043
Accuracy			0.94	
Macro Avg	0.93	0.92	0.93	8418
Weighted Avg	0.94	0.94	0.94	8418

Table 6.1: Classification Report (Random Forest, Amplification Curve)

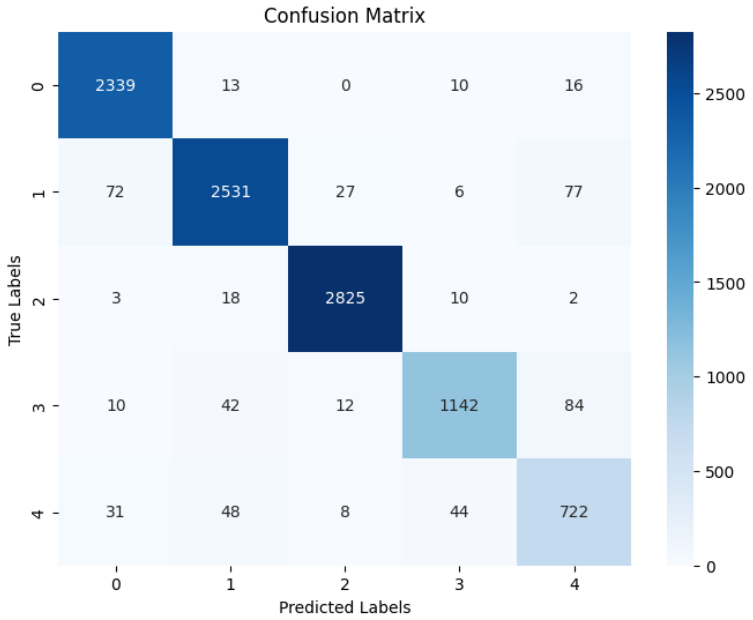


Figure 6.2: Accuracy of Baseline Models with Different Input Data on Test Amplification Data Set

6.3 Testing Transformer Model

The transformer model outperformed the previous baseline models in terms of performance. Figure 6.3 illustrates the training and validation steps. The training loss converges to zero at a faster rate compared to the validation loss, while both accuracies steadily improve and reach higher levels. Although there is a consistent gap between the training and validation loss, this does not necessarily indicate over-fitting, as the gap progressively diminishes with increasing epochs.

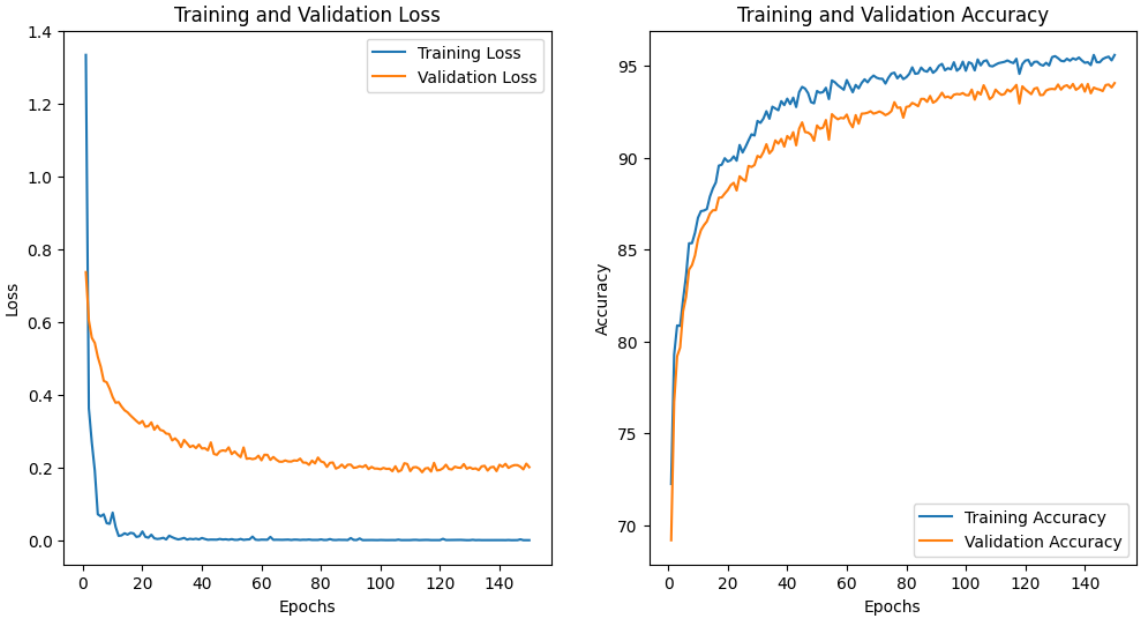


Figure 6.3: Training and Validation Loss and Accuracy for Transformer Model on Amplification Curve

The final transformer model achieved an accuracy of 94.72% on 10,092 amplification events. To provide a comprehensive evaluation of its performance, Figure 6.4 presents the confusion matrix, while Table 6.2 provides a detailed classification report encompassing accuracy, precision, recall, and F1-score.

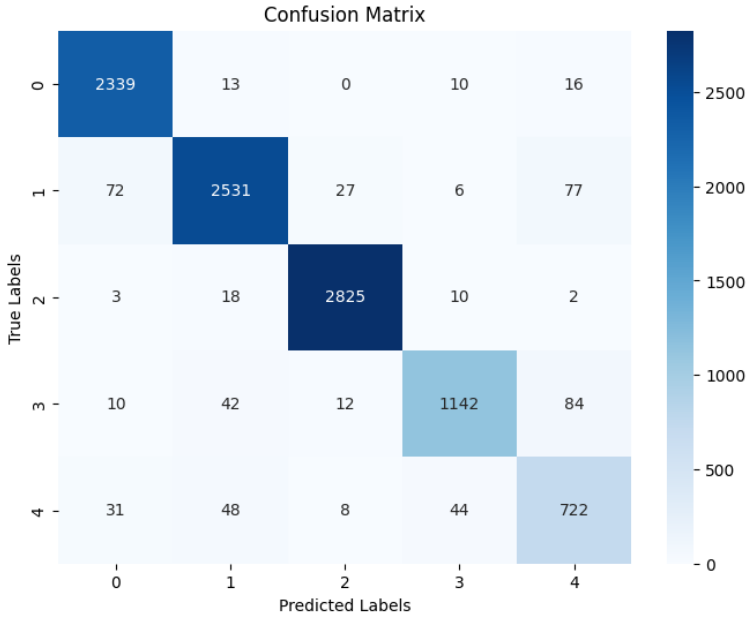


Figure 6.4: Confusion Matrix for Transformer Model on Amplification Curve

Class	Precision	Recall	F1-Score	Support
AD	0.95	0.98	0.97	2378
IA	0.95	0.93	0.94	2713
IB	0.98	0.99	0.99	2858
KP	0.94	0.89	0.91	1290
C19	0.80	0.85	0.82	853
Accuracy			0.95	
Macro Avg	0.93	0.93	0.93	10092
Weighted Avg	0.95	0.95	0.95	10092

Table 6.2: Classification Report (Transformer, Amplification Curve)

6.4 Melt Curve Results

6.4.1 Melt Curve Baseline Models

The melt curve produced significant results for the majority of machine learning models, suggesting that the melt curve data provides a more comprehensive description than the amplification curve data. A comprehensive presentation of the accuracy results from all models, based on three different data inputs, can be found in Figure 6.5. As expected, the accuracy wasn't as high when using only the melt peak as input data, given the constraints of using a single feature to describe five distinct classes. However, employing the melt curve as data input yielded promising outcomes, with several models nearing flawless performance.

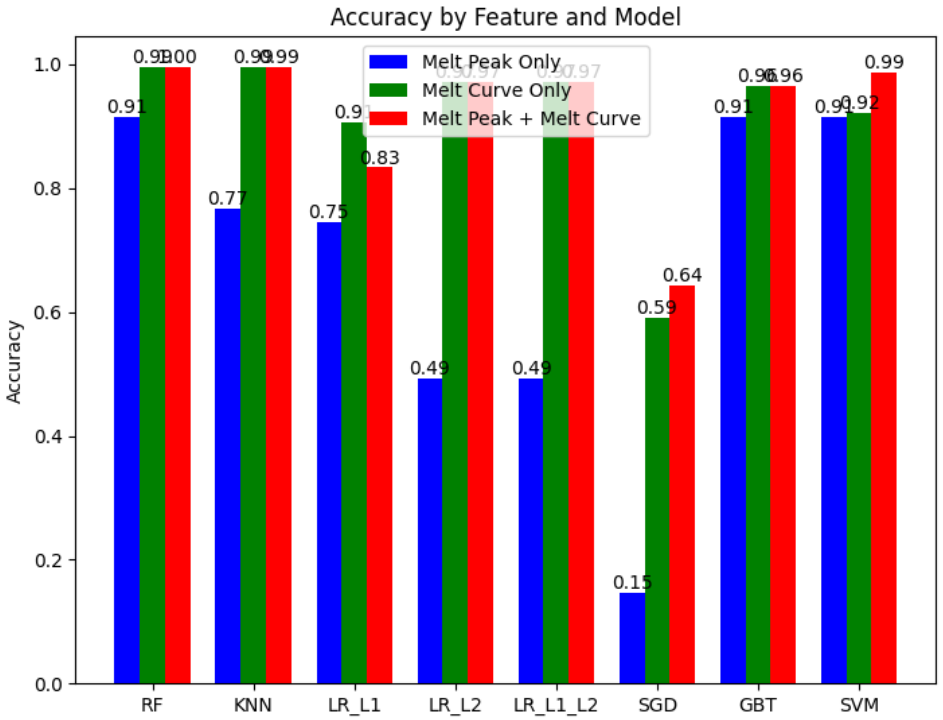


Figure 6.5: Accuracy of Baseline Models with Different Input Data on Test Melt Data Set

The most successful model was, once again, the random forest classifier, delivering an accuracy rate of 99.52% on the test dataset. This time, the configuration of the random forest classifier was RandomForestClassifier(n_estimators=10, criterion='entropy'). The confusion matrix and classification results of this specific model are depicted in Figures 6.6 and Table 6.3, respectively.

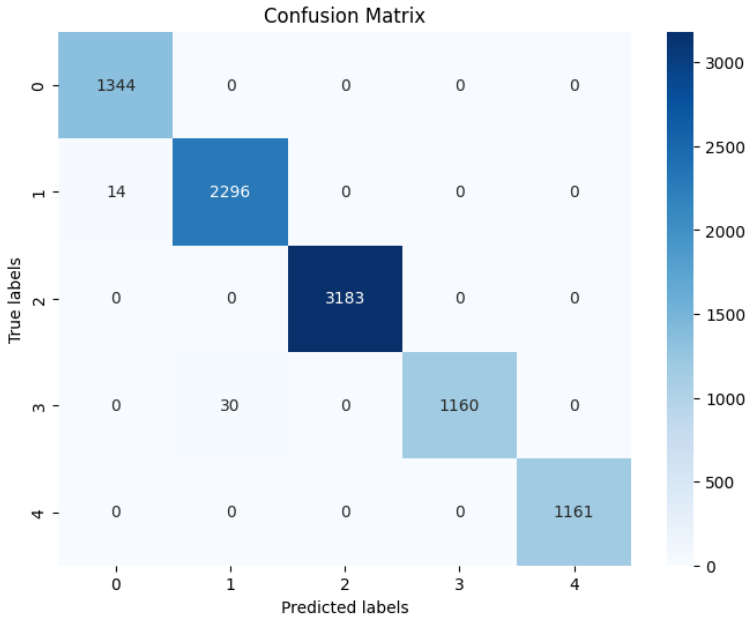


Figure 6.6: Confusion Matrix for Best Machine Learning Model (Random Forest) on Melt Curve

Class	Precision	Recall	F1-Score	Support
AD	0.99	1.00	0.99	1344
IA	0.99	0.99	0.99	2310
IB	1.00	1.00	1.00	3183
KP	1.00	0.97	0.99	1190
C19	1.00	1.00	1.00	1161
Accuracy			1.00	
Macro Avg	1.00	0.99	0.99	9188
Weighted Avg	1.00	1.00	1.00	9188

Table 6.3: Classification Report (Random Forest, Melt Curve)

6.4.2 Melt Curve Transformer Model

The transformer model used for the melt curve achieved higher accuracy more rapidly compared to the amplification curve. This difference may be attributed to the fact that the data in the melt curve was easier to classify. Figure 6.7 displays the training and validation loss and accuracy graphs. It is evident from the graph that the model reached a low loss and high accuracy after approximately 60 epochs, indicating that the additional 70 epochs may not have been necessary.

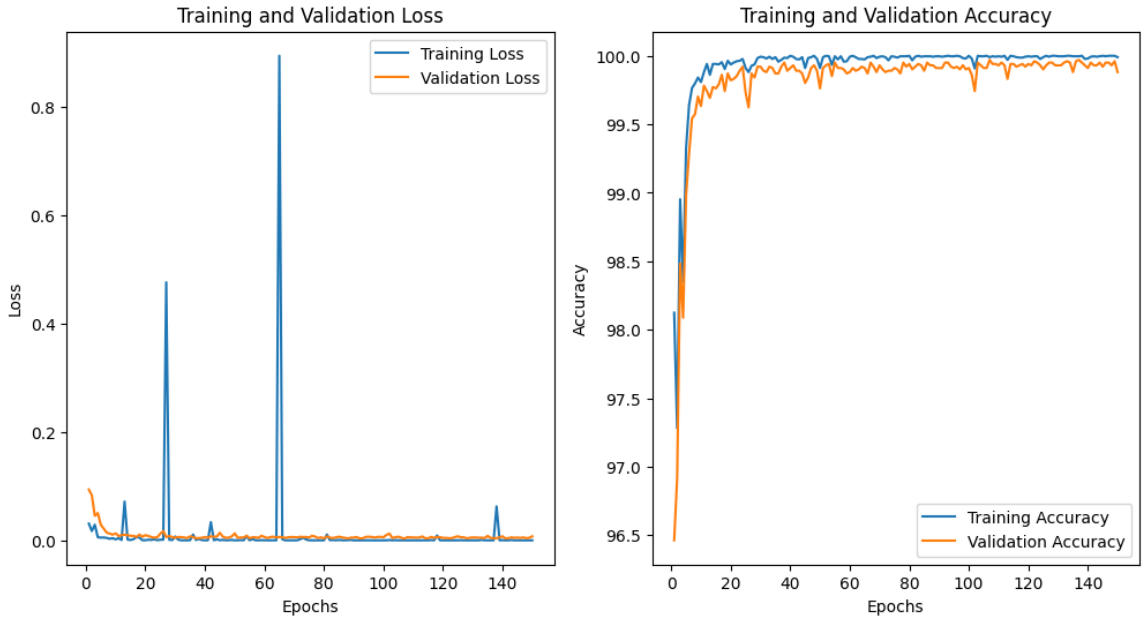


Figure 6.7: Training and Validation Loss and Accuracy for Transformer Model on Melt Curve

Building on the near perfect accuracy of the melt curve machine learning models, the melt curve transformer model achieved a perfect accuracy rate of 100.00% on the 10,092 amplification events.This remarkable outcome signifies a flawless confusion matrix with zero misclassifications, resulting in perfect precision, recall, and F1-score. The classification outcomes can be observed in Figure 6.8, which presents the confusion matrix. Additionally, Table 6.7 provides a classification report detailing the precision, recall, F1-score, and accuracy for the five different classes. The evaluation section will delve further into the broader implications of these results, including an analysis of their significance in terms of performance measures and biological implifications.

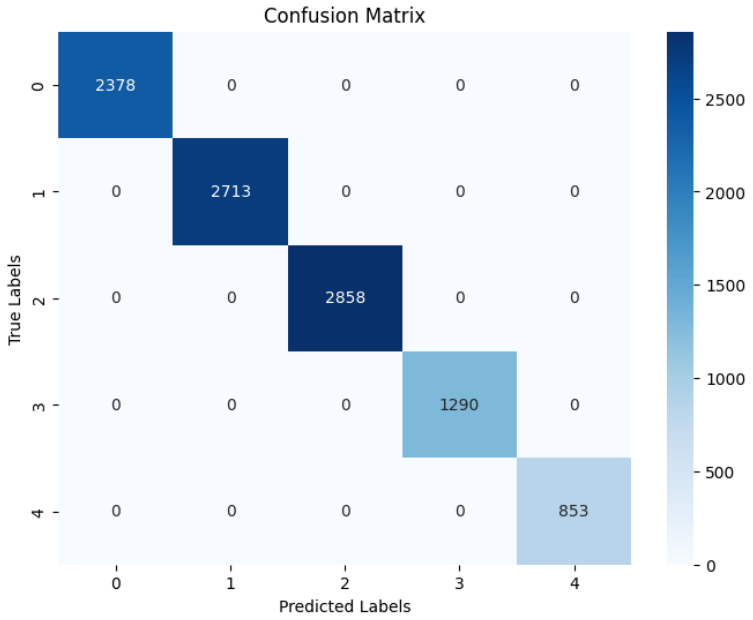


Figure 6.8: Confusion Matrix for Transformer Model on Melt Curve

Class	Precision	Recall	F1-Score	Support
AD	1.00	1.00	1.00	2378
IA	1.00	1.00	1.00	2713
IB	1.00	1.00	1.00	2858
KP	1.00	1.00	1.00	1290
C19	1.00	1.00	1.00	853
Accuracy			1.00	
Macro Avg	1.00	1.00	1.00	10092
Weighted Avg	1.00	1.00	1.00	10092

Table 6.4: Classification Report (Transformer, Melt Curve)

Chapter 7

Evaluation

This section will discuss the results obtained from the previous sections, along with the significance of each outcome. A comparison will be made between the baseline machine learning models and the transformer models. Furthermore, the input data derived from the amplification curve will be compared against the melt curve. The evaluation will encompass the performance of the different models, as well as the biological significance and potential implications of the results.

Through this analysis, valuable insights can be gained regarding the effectiveness of the models and the biological implications that can be inferred from the obtained results.

7.1 Model Performance

7.1.1 Transformer Model Performance

The evaluation focuses on precision, recall, F1-score, and accuracy metrics from Table 6.2 to assess the model's effectiveness in correctly predicting class labels across different classes. The results reveal the strengths and weaknesses of the model, highlighting its overall robustness and areas for improvement.

Precision Scores:

Precision, which measures the accuracy of positive predictions made by the model, serves as our initial evaluation criterion. The model demonstrates high precision values across most classes. Notably, AD, IA, and IB exhibit precisions of 0.95, 0.95, and 0.98, respectively. These exceptional precision values signify the model's strong ability to correctly identify positive samples for these specific classes. Nevertheless, it is worth mentioning that C19 displays a slightly lower precision of 0.80, suggesting some difficulty in accurately predicting samples for this particular class.

Recall Scores:

To evaluate the model's performance in correctly identifying positive samples, the recall scores can be used. Recall measures the model's ability to correctly identify positive samples among all the

actual positive samples. In most classes, the model demonstrates good recall values. Specifically, AD, IA, and IB exhibit recalls of 0.98, 0.93, and 0.99, respectively. These scores indicate that the model effectively captures the majority of positive samples for these classes. However, KP and C19 present slightly lower recalls of 0.89 and 0.85, respectively, suggesting that the model may encounter challenges in accurately identifying all positive samples for these particular classes.

F1-Scores:

F1-scores serve as a well-rounded metric that balances both precision and recall, providing a comprehensive evaluation of the model's performance. The model demonstrates high F1-scores for the majority of classes. Specifically, AD, IA, and IB achieve F1-scores of 0.97, 0.94, and 0.99, respectively, indicating a good balance between precision and recall for these classes. On the other hand, KP and C19 exhibit relatively lower F1-scores of 0.91 and 0.82, respectively, hinting a trade-off between precision and recall in these particular cases.

Accuracy:

With an overall accuracy of 0.95, the model demonstrates strong performance in correctly predicting class labels for the given dataset. This high accuracy underscores the effectiveness of the Transformer model in classification tasks, emphasizing its ability to yield accurate results across various classes.

Macro-Averaged and Weighted-Averaged Metrics:

The macro-averaged and weighted-averaged metrics provide additional details of the model's performance. Macro-averaged precision, recall, and F1-score all equal 0.93, indicating a balanced performance across classes without favoring any particular class. Similarly, the weighted-averaged metrics yield values of 0.95, taking into account the class distribution. These metrics validate the overall robustness and effectiveness of the model across the dataset.

In conclusion, the final transformer model exhibits impressive performance in the classification task, as shown by its high accuracy, precision, recall, and F1-score values. The model demonstrates the ability to accurately classify most classes, particularly excelling in AD, IA, and IB. However, it exhibits slightly lower performance for KP and C19. The evaluation emphasizes the model's overall robustness and highlights areas where further improvement may be necessary.

7.1.2 Comparison with Previous Models

When compared specifically to the k-NN model introduced by Malpartida-Cardenas et al. [7], the transformer model applied to the amplification curve exhibits remarkably effective performance. The k-NN model previously reported a success rate of 91.33% on a dataset comprising 54,186 positive amplification events. In contrast, our project utilized a transformer model and achieved a superior overall accuracy of 94.72% on a dataset encompassing 10,092 positive amplifications.

Furthermore, the model by Malpartida-Cardenas et al. reports a 94.66% accuracy on the melt curve, whereas our transformer model achieved a perfect accuracy of 100.00% on the given dataset. This superior performance of the transformer model on both the classification of LAMP events using amplification curves and melt curves suggests its enhanced effectiveness. Considering these results, it can be concluded that our project has achieved significant success.

7.1.3 Melt Curve Transformer Model Performance

In contrast to the amplification curve transformer model, the melt curve transformer model merits less discussion, primarily because it achieved perfect classification of all tasks, registering an impeccable accuracy of 100.00% on the 10,095 amplification events. Even the most proficient machine learning model, the k-NN algorithm applied to the melt peak + melt curve data, yielded an accuracy of 99.52%. This may suggest that a complex transformer model may not be necessary for the given dataset. However, in a biological or healthcare context, even the slightest enhancement in accuracy can be vital. Therefore, even a seemingly negligible increase in accuracy offered by the transformer model can be beneficial in real-world scenarios.

7.2 Amplification Curve vs Melt Curve

7.2.1 Performance Difference

The accuracy metrics clearly indicate that the melt curve consistently delivered superior performance across various types of machine learning models. Remarkably, the transformer model applied to the melt curve achieved flawless accuracy on the test set, with no instances of mislabelled data. Moreover, even a less sophisticated model like the random forest classifier produced a commendably high accuracy of 99.52%, suggesting that the melt curve provided a more comprehensive representation for the given classification task. However, despite these results, it would be premature to assert that the melt curve will invariably be the superior option for real-world applications, particularly taking into account the biological significance discussed in the subsequent section.

7.2.2 Biological Significance

The results clearly indicate that the melt curve provides a more descriptive representation of the target DNA sequence, as evidenced by the higher accuracy of the melt curve models. However, it is important to note that this does not imply that the melt curve model will always be the preferred choice. Real-life scenarios necessitate considering additional factors.

One advantage of LAMP over PCR is its isothermal nature, allowing for easier execution in diverse environments. Conversely, obtaining the melt curve requires heating the assay after the amplification process, rendering the reaction non-isothermal. This process is more costly and time-consuming compared to conducting the LAMP experiment alone. Thus, obtaining the melt curve data may not always be feasible.

Considering the practical implications, a judicious approach would be to employ both the amplification curve and the melt curve, depending on the specific requirements of the situation, as well as time and cost considerations. This approach ensures flexibility and allows for the utilization of the most appropriate method based on the given circumstances.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This project builds upon the work proposed by Malpartida-Cardenas et al. [7] and aims to enhance the model's performance by employing cutting-edge deep learning models on the amplification data from LAMP experiments. Through this endeavor, transformer models were successfully developed for both the amplification curve and melt curve obtained from LAMP experiments. These models exhibited outstanding performance in accurately classifying each respective curve into the target DNA sequence. The amplification curve transformer model achieved an impressive accuracy of 94.72%, while the melt curve transformer model achieved a flawless accuracy of 100.00%.

Although the melt curve model exhibited significantly higher accuracy in comparison, the practical implications of utilizing melt curves suggest the necessity for both models at different times. The amplification curve model proves more useful in scenarios where only the LAMP experiment is conducted, such as an isothermal testing environment being available. On the other hand, the melt curve model requires an increase in temperature and additional time, but offers higher accuracy. This makes it more suitable when the environmental conditions or cost requirements allow for such experimentation.

8.2 Future Work

This project presents various opportunities for improvement, as introducing new hyperparameters and incorporating additional datasets can potentially enhance the robustness and performance of the models. Particularly, the amplification curve analysis has more room for improvement compared to the melt curve, which already exhibits near-perfect performance. By exploring alternative hyperparameter configurations and incorporating more diverse datasets, it is possible to further optimize the models' performance and make them more reliable in real-world scenarios. Continuously seeking opportunities for improvement is essential for staying at the forefront of research and achieving even better results.

8.2.1 Transfer Learning

To enhance the performance and adaptability of deep learning models, employing a transfer learning approach can be highly effective. By freezing a pre-trained transformer model and applying transfer learning techniques, it becomes possible to improve accuracy and apply the algorithm to different datasets, accommodating various combinations of target primers. The current transformer model already demonstrates high accuracy on amplification curves, making the prospect of further enhancing this accuracy through transfer learning particularly appealing. Additionally, transfer learning presents opportunities to classify new target DNA sequences, enabling the pre-trained model to learn new curve patterns. This approach opens up exciting possibilities for multiplex LAMP, enabling efficient and accurate classification of amplification curves to their respective targets within a reasonable time frame.

Appendix A

Appendix

The appendix exclusively contains the code used for developing the models and specifically includes the code for the best performing models. The code can be accessed through the following link: https://github.com/sh1319/FYP-Intelligent-Algorithms-for-DNA-Detection. Within the repository, detailed information is provided regarding data visualization, pre-processing, as well as the implementation of baseline models and transformer models for both the amplification curve and melt curve. Furthermore, the repository includes the code for the best performing models that are discussed in the evaluation section.

Bibliography

[1] T. Notomi, H. Okayama, H. Masubuchi, T. Yonekawa, K. Watanabe, N. Amino, and T. Hase, "Loop-mediated isothermal amplification of DNA," *Nucleic Acids Res*, vol. 28, p. E63, Jun 2000.

[2] K. Nagamine, T. Hase, and T. Notomi, "Accelerated reaction by loop-mediated isothermal amplification using loop primers," *Mol Cell Probes*, vol. 16, pp. 223–229, Jun 2002.

[3] R. Augustine, A. Hasan, S. Das, R. Ahmed, Y. Mori, T. Notomi, B. D. Kevadiya, and A. S. Thakor, "Loop-mediated isothermal amplification (lamp): A rapid, sensitive, specific, and cost-effective point-of-care test for coronaviruses in the context of covid-19 pandemic," *Biology (Basel)*, vol. 9, no. 8, p. 182, 2020.

[4] S. Ryding, "What is rt-lamp technology?."

[5] W. S. Jang, D. H. Lim, J. Yoon, A. Kim, M. Lim, J. Nam, R. Yanagihara, S. W. Ryu, B. K. Jung, N. H. Ryoo, and C. S. Lim, "Development of a multiplex loop-mediated isothermal amplification (LAMP) assay for on-site diagnosis of SARS CoV-2," *PloS One*, vol. 16, no. 3, p. e0248042, 2021.

[6] Y. Dong, Y. Zhao, S. Li, Z. Wan, R. Lu, X. Yang, G. Yu, J. Reboud, J. M. Cooper, Z. Tian, and C. Zhang, "Multiplex, real-time, point-of-care rt-lamp for sars-cov-2 detection using the hfman probe," *ACS Sensors*, vol. 7, no. 3, pp. 730–739, 2022.

[7] K. Malpartida-Cardenas, L. Miglietta, T. Peng, A. Moniri, A. Holmes, P. Georgiou, and J. Rodriguez-Manzano, "Single-channel digital lamp multiplexing using amplification curve analysis," *Sens. Diagn.*, vol. 1, pp. 465–468, 2022.

[8] B. C. Delidow, J. P. Lynch, J. J. Peluso, and B. A. White, "Polymerase chain reaction: Basic protocols," *Methods in Molecular Biology*, vol. 15, pp. 1–29, 1993.

[9] Y. Ho Kim, I. Yang, Y.-S. Bae, and S.-R. Park, "Performance evaluation of thermal cyclers for pcr in a rapid cycling condition," *BioTechniques*, vol. 44, no. 4, pp. 495–505, 2008. PMID: 18476814.

[10] J. Gray and L. J. Coupland, "The increasing application of multiplex nucleic acid detection tests to the diagnosis of syndromic infections," *Epidemiology and infection*, vol. 142, pp. 1–11, Jan 2014.

[11] L. Kreitmann, L. Miglietta, K. Xu, K. Malpartida-Cardenas, G. D'Souza, M. Kaforou, K. Brengel-Pesce, L. Drazek, A. Holmes, and J. Rodriguez-Manzano, "Next-generation molecular diagnostics: Leveraging digital technologies to enhance multiplexing in real-time pcr," *TrAC Trends in Analytical Chemistry*, vol. 160, p. 116963, 2023.

[12] N. Tomita, Y. Mori, H. Kanda, and T. Notomi, "Loop-mediated isothermal amplification (lamp) of gene sequences and simple visual detection of products," *Nature Protocols*, vol. 3, no. 5, pp. 877–882, 2008.

[13] M. Soroka, B. Wasowicz, and A. Rymaszewska, "Loop-mediated isothermal amplification (lamp): The better sibling of pcr?," *Cells*, vol. 10, no. 8, 2021.

[14] A. Moniri, L. Miglietta, K. Malpartida-Cardenas, I. Pennisi, M. Cacho-Soblechero, N. Moser, A. Holmes, P. Georgiou, and J. Rodriguez-Manzano, "Amplification curve analysis: Data-driven multiplexing using real-time digital PCR," *Analytical Chemistry*, vol. 92, no. 19, pp. 13134–13143, 2020.

[15] L. Wan, T. Chen, J. Gao, C. Dong, A. H.-H. Wong, Y. Jia, P.-I. Mak, C.-X. Deng, and R. Martins, "A digital microfluidic system for loop-mediated isothermal amplification and sequence specific pathogen detection," *Scientific Reports*, vol. 7, 11 2017.

[16] X. Lin, X. Huang, K. Urmann, X. Xie, and M. R. Hoffmann, "Digital loop-mediated isothermal amplification on a commercial membrane," *ACS Sensors*, vol. 4, no. 1, pp. 242–249, 2019.

[17] X. Zhu, Y. Ge, T. Wu, K. Zhao, Y. Chen, B. Wu, F. Zhu, B. Zhu, and L. Cui, "Co-infection with respiratory pathogens among covid-2019 cases," *Virus Research*, vol. 285, p. 198005, 2020.

[18] Y. Li, c. Wang, Haizhou MDb, c. Wang, Fan MDb, X. Lu, H. Du, J. Xu, F. Han, L. Zhang, and M. Zhang, "Co-infections of SARS-CoV-2 with multiple common respiratory pathogens in infected children: A retrospective study," *Medicine*, vol. 100, p. e24315, March 2021.

[19] N. Zhang, L. Wang, X. Deng, R. Liang, M. Su, C. He, L. Hu, Y. Su, J. Ren, F. Yu, L. Du, and S. Jiang, "Recent advances in the detection of respiratory virus infection in humans," *Journal of Medical Virology*, vol. 92, no. 4, pp. 408–417, 2020.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[21] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," 2019.

[22] B. Zhao, H. Xing, X. Wang, F. Song, and Z. Xiao, "Rethinking attention mechanism in time series classification," 2022.

[23] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," 2023.

[24] L. Miglietta, K. Xu, P. Chhaya, L. Kreitmann, G. A. Hill-Cawthorne, F. Bolt, A. Holmes, P. Georgiou, and J. Rodriguez-Manzano, "Adaptive filtering framework to remove nonspecific and low-efficiency reactions in multiplex digital pcr based on sigmoidal trends," *Analytical Chemistry*, vol. 94, no. 41, pp. 14159–14168, 2022.

[25] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, "Scipy 1.7.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 18, no. 3, pp. 257–261, 2021.

[26] PyTorch, "transformer_tutorial.py." Code, 2023. In PyTorch Tutorials.

[27] Python Software Foundation, "itertools – functions creating iterators for efficient looping." https://docs.python.org/3/library/itertools.html. Accessed on 16/6/2023.

[28] P. Nellihela, "What is k-fold cross validation?," Nov 2022.