ProteoKnight: Phage Virion Protein Classification with CNN and Uncertainty Quantification

by

Abir Ahammed Bhuiyan
20101197
Samiha Afaf Neha
20101266
Md. Ishrak Khan
20101051

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2024

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Abir Ahammed Bhuiyan
20101197

Samiha Afaf Neha
20101266

Md. Ishrak Khan
20101051

Approval

The thesis project titled "ProteoKnight: Phage Virion Protein Classification with CNN and Uncertainty Quantification" submitted by

1. Abir Ahammed Bhuiyan(20101197)
2. Samiha Afaf Neha(20101266)
3. Md. Ishrak Khan(20101051)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 22nd, 2024.

Examining Committee:

Supervisor:
(Member)

Md. Khalilur Rhaman, PhD
Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Ms. Jannatun Noor Mukta
Assistant Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

____

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Dedication

“Dedicated to the resilient and enduring spirit of the people of Palestine, whose courage and strength in the face of adversity continue to inspire us.”

Acknowledgement

First and foremost, we thank Allah subhanahu wa ta'ala for his blessings, which have enabled us to successfully complete our thesis without any disruptions.

We would like to express our deepest gratitude to our thesis Supervisor and Co-Supervisor Professor Dr. Mohammad Khalilur Rhaman Sir and Ms. Jannatun Noor Mukta, for their expert guidance, invaluable support, and continuous encouragement throughout the course of this research. Their wisdom, knowledge, and commitment to the highest standards inspired and motivated us.

Abstract

Bacteriophages, often known as phages, have a significant impact on the dynamics of microbial ecosystems. This has led to their increased utilization in several research areas, such as bacterial genome engineering, phage therapy, disease diagnostics, and viral host identification. The structure of phages is made up of proteins called phage virion proteins (PVP). Classifying these proteins is important for genomic research, which in turn helps us understand the complex interactions between phages and their hosts in the context of making antibacterial drugs. Replacing the tedious traditional procedures, a growing number of computational strategies are being employed to annotate phage protein sequences acquired using high-throughput sequencing. Among these techniques, deep learning approaches demonstrate improved performance in classification outcomes. Such procedures require special sequence encodings for the model to perceive the protein sequences with their distinctive features. Numerous ways have been examined and assessed, while novel methods continue to emerge in order to optimize the task in terms of resource utilization and prediction accuracy. The objective of our work, ProteoKnight, is to explore and develop a unique encoding technique for phage proteins and demonstrate its effectiveness via classification. In our work, we make use of the time-separated PVP dataset that [47] introduced. Furthermore, this study aims to address the lack of research conducted on uncertainty analysis by exploring the domain of uncertainty in binary PVP classification using Monte Carlo Dropout (MCD) method. The experimental findings demonstrate the effectiveness of our strategy for binary classification, achieving a prediction accuracy of 90.2%. However, the accuracy for multi-class classification remains suboptimal. Furthermore, our uncertainty analysis reveals that the class and sequence length show variability in prediction confidence for our suggested classification approach.

Keywords: Proteins, Classification, Phage Virion, Deep Learning, CNN, Uncertainty, Monte Carlo Dropout, DNA-Walk.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI	Artificial Intelligence
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
CCT	Compact Convolutional Transformers
CGR	Chaos Game Representation
CTD	Compostion, Transition, Distribution
DL	Deep Learning
DPC	Dipeptide Composition
FCGR	Frequency Chaos Game Representation
GBC	Gradient Boosting Classifier
GP	Gaussian Processes
MCD	Monte Carlo Dropout
MVE	Mean-Variance Estimation
PseAAC	Pseudo-Amino Acid Composition
PSSM	Position Specific Scoring Matrix
PVP	Phage Virion Protein
RefSeq	NCBI Reference Sequences
RF	Random Forest
SCM	Scoring Card Method
SVM	Support Vector Machine
UQ	Uncertainty Quantification
ViT	Vision Transformer

Chapter 1

Background

1.1 Proteins and PVP

Proteins are essential macromolecules responsible for the structural and functional mechanisms of living things. They play a vital role in the organization, functioning, and regulation of the body's various tissues and organs and predominantly execute their functions within cellular environments. Amino acids connect to one another to generate peptide bonds, which are the fundamental building blocks of proteins, as illustrated in Fig. 1.1. In every living organism, there are 20 different types of standard amino acids. These amino acid residues are often denoted by a single letter (e.g., Alanine: A, Cysteine: C, etc.). Phage structural proteins (i.e., virion proteins) are a subclass of the protein family corresponding to the bacteriophages, which are the most prevalent kind of biological creatures [16]. The PVPs are mainly associated with building structural constituents of the bacteria [41], such as the baseplate and head as shown in Fig. 1.2, enabling efficient host-phage binding for genome transfer. The amino acid sequences, particularly those involved in structure synthesis, exhibit significant diversity in sequences among phages and their respective groups [9]. Consequently, characterizing these sequences becomes a challenging yet crucial task, as the shortage of annotations for phage proteins has emerged as a hindrance in numerous research studies focused on phage genomics [7].

1.2 Uncertainty in Deep Learning

Uncertainty in deep learning classification refers to the model's challenge in providing a definitive and precise prediction, recognizing the inherent complexity and ambiguity in real-world data. Unlike conventional models that produce deterministic results, deep learning systems frequently encounter uncertainty arising from diverse causes.

Epistemic uncertainty arises from the model's lack of knowledge about the data distribution, and aleatoric uncertainty emerges from the inherent stochasticity or randomness in the data. Even with extensive training data, aleatoric uncertainty persists due to the intrinsic unpredictability of observed phenomena. Various methods such as Bayesian approaches, ensemble methods, and Monte Carlo dropout techniques have been developed to quantify such uncertainties for a particular inference task. These validation processes ensure that the model's predictions are not

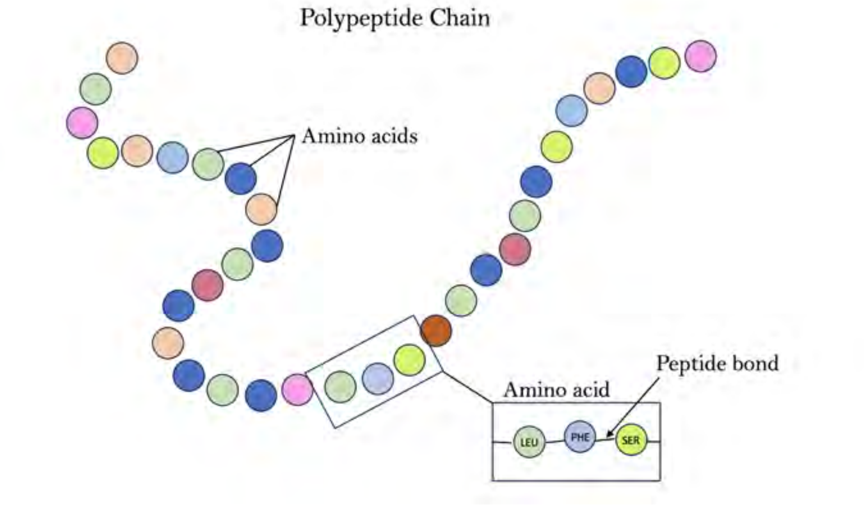


Figure 1.1: Generalized illustration of a typical protein sequence.

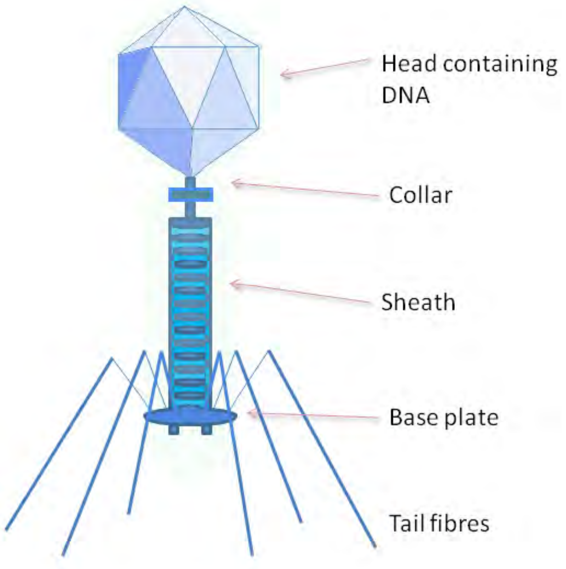


Figure 1.2: Components comprising the structure of a Phage.

just based on deterministic outcomes but also considering the inherent uncertainties present in the data. Such an approach is particularly crucial in applications where the consequences of model predictions have significant implications, as it allows for a more comprehensive and cautious assessment of the model's confidence in its results.

1.2.1 Monte Carlo Dropout

The concept of utilizing dropout was introduced by Gal and Ghahramani [17], who employed it as a means of approximating probabilistic Bayesian models for deep Gaussian processes. Dropout is a frequently employed regularization approach that serves to mitigate the issue of overfitting. Bayesian neural networks aim to acquire knowledge about the posterior distribution of weights, conditioned on a given input sample. However, the computation of these posteriors in an analytical manner is infeasible. Sampling techniques can be employed to estimate the weight posteriors as an alternative. The proposed approach involves conducting numerous stochastic evaluations utilizing distinct weight samples within our model.

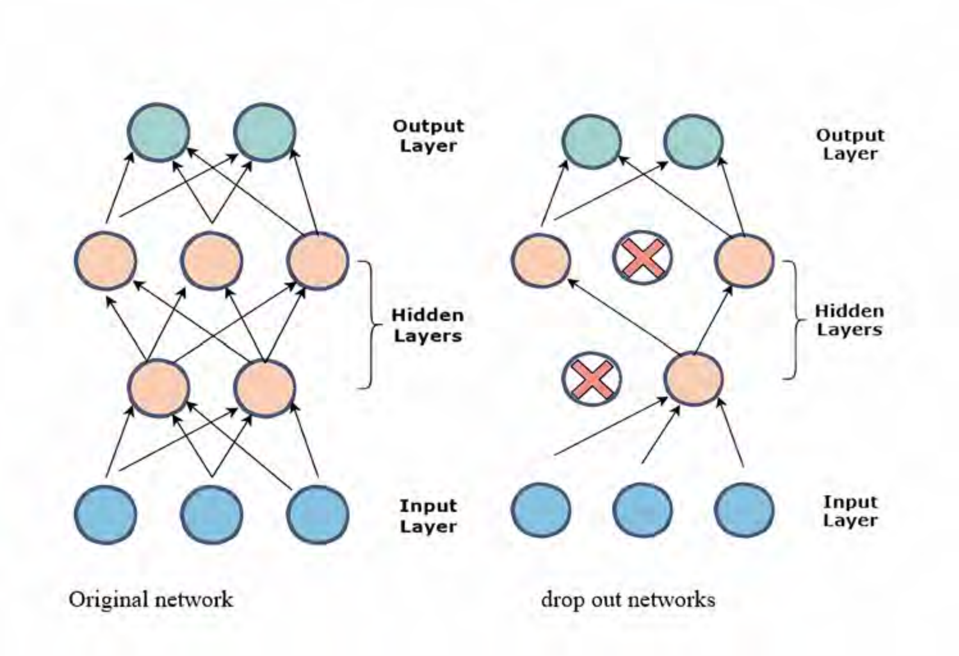


Figure 1.3: Dropout Mechanism for Neural Networks [45]

Monte Carlo Dropout is such a sampling technique utilized to estimate the weight posterior in a given set of data. The functioning of the mechanism involves the utilization of a technique called ”dropout” during the training phase. Dropout refers to the process of randomly selecting whether or not to retain nodes inside a neural network. This decision is made based on the outcome of a Bernoulli random variable, which follows a specific probability distribution denoted as P as shown in equation 1.1, where t indicates a specific trial for variable w. The dropout approach, as seen in Figure 1.3, incorporates this probability parameter. Therefore, when the same input is sent through a model, the resulting output may vary slightly depending on which nodes are activated or dropped out. Each of these outputs represents a distinct sample within our network.

$$Z_{w,t} \sim Bernoulli(p) \,\forall\, w \in \mathbf{W} \tag{1.1}$$

In this scenario, essentially T samples from the weight distribution are selected as shown in equation 1.2) and useed to perform T stochastic forward passes with dropout where the input is denoted as $\mathbf{X}$ and the corresponding output is $\mathbf{Y}$. This allows the calculating of the expected value of predictions Y and the variance of these predictions across the iterations as measured using equations 1.3 1.4. Thus, using T stochastic forward passes using different samples of weights $\{W_t\}_{t=1}^{T}$, dropout acts as a form of stochastic sampling in this scenario. The observed variability in the T stochastic outputs primarily represents the uncertainty inherent in the model or the epistemic uncertainty . In other words, when the predictions exhibit a large variance, it suggests that the model is characterized by significant epistemic uncertainty.

$$\mathbf{W}_t = \text{train}(f; \mathbf{X}, \mathbf{Y}) \tag{1.2}$$

$$\mathbb{E}(\hat{\mathbf{Y}} \mid \mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{X} \mid \mathbf{W}_t) \tag{1.3}$$

$$Var(\hat{\mathbf{Y}} \mid \mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{X})^2 - \mathbb{E}(\hat{\mathbf{Y}} \mid \mathbf{X})^2 \tag{1.4}$$

Chapter 2

Introduction

2.1 In-Silico Classification of PVP

With the advent of high-throughput sequencing technology, rapid additions of sequences are being observed in standard biological databases such as UniProt [48] as illustrated in Fig.2.1, giving rise to the need for proper annotation of the sequences. Enhancement and forwarding of annotations for the phage family is vital for exploring effective anti-bacterial drug synthesis [30][24], disease diagnosis [3], food production [1] , bacterial genome remodeling [14], etc. The identification and characterization of PVPs are commonly achieved through the utilization of mass spectrometry and protein array techniques [6] [19]. However, these methods are associated with significant time, labor, and computational expenses [16]. The conventional alignment-based approaches employed for predicting homology relationships also fail to yield satisfactory outcomes in the classification of virion proteins as the sequences show variety in residue frequency and position, leading to dissimilar sequences having similar structural conformations and vice versa. These constraints on annotation, along with the rapid increase of sequenced data, have resulted in 50–90 % unannotated phage genes [4], with only 33% annotated phage proteins in the RefSeq phage protein database.

In light of these issues, in-silico or computational approaches leveraging machine learning techniques for faster and lower-cost classification of phage proteins are increasingly gaining acclaim. The amino acid sequences are initially encoded using various encoding schemes to enhance machine readability. These encodings vary in dimension and feature selection, and each encoding scheme extracts specific features from the protein sequence necessary for their identification. The prediction models subsequently use these encoded sequences to perform their calculations. As mentioned previously, alignment-based methods do not always work well with PVP categorization due to the lack of collinearity [5] in viral genomes, resulting from instances such as horizontal gene transfer, high mutation rates, etc. Thus, alignment-free methods proved more plausible for approaching the classification task. These methods can be categorized into model-based and feature-based. This study aims to utilize machine learning by focusing on the feature-based technique, which involves extracting features from raw sequences.

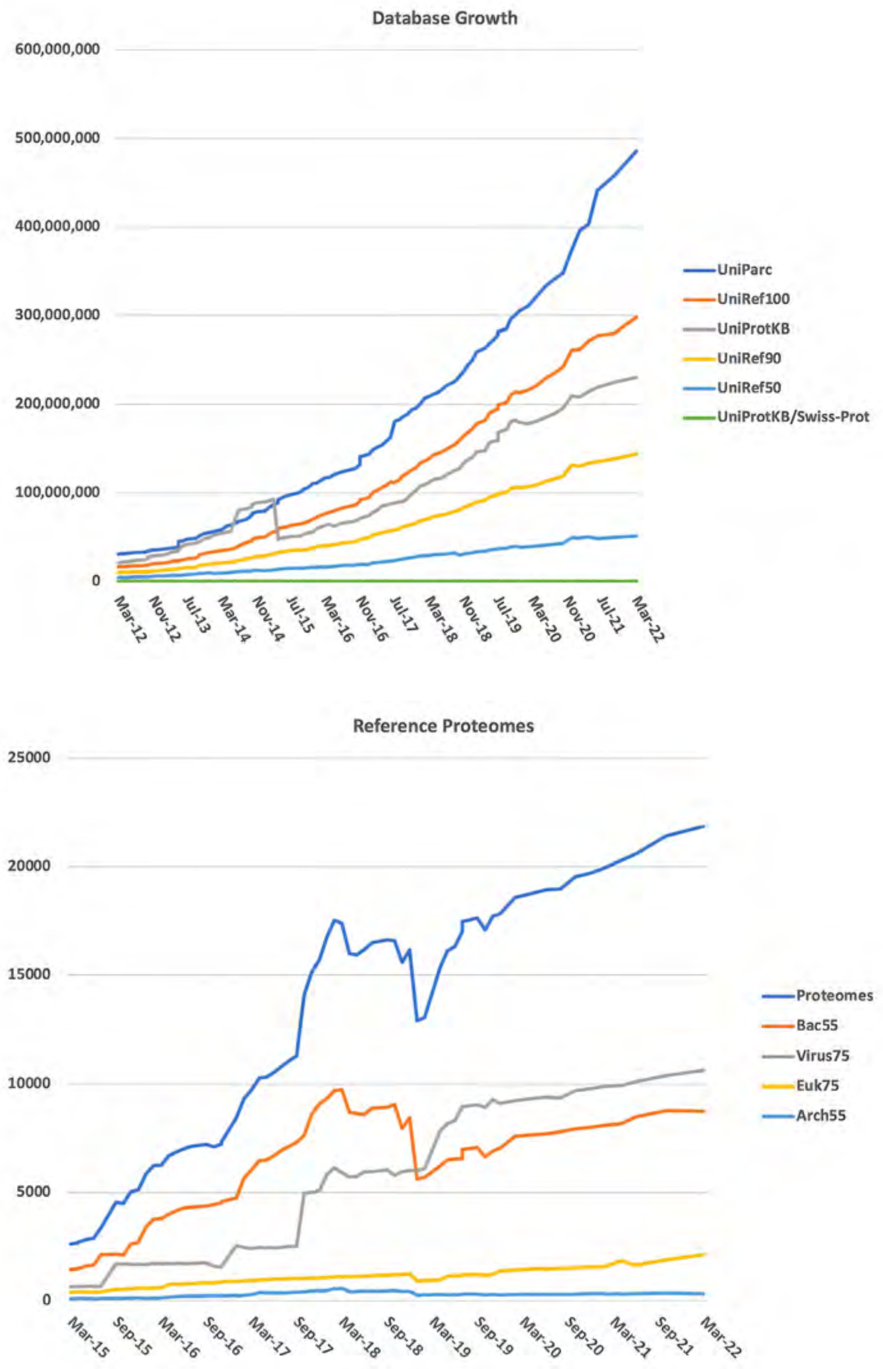


Figure 2.1: Growth of UniProt databases over the last 10 years [48]

These features are made up of different frequencies of peptides, side chains, k-mers, and the vectorized one-hot numerical encoding of amino acid residues. Studies analyzing sequence data have demonstrated that when represented as 2-D images, they exhibit enhanced characteristics due to increased feature space [46]. The augmentation of dimensionality typically results in improved performance and simplifies the process of feature extraction by applying a standard text-to-image encoding on the sequences, rather than relying on the collection of several distinct features based on the sequences' physiochemical properties. Therefore, numerous studies, such as those mentioned in [47] and [39], have suggested the utilization of image encoding techniques for biological sequences. The only study that applies picture encoding for PVP classification is the one conducted by [47], which employs FCGR. However, these encoding methods based on k-mer frequency have a tendency to remove the spatial information of the sequences because of their compact transformation, as mentioned in the study by Akbari et al. (2022) [39]. This highlights the need to discover a more effective encoding strategy. Given the constraints of both earlier and contemporary encodings, researchers are motivated to create innovative and more effective attributes that are customized to their particular objectives. However, these attributes may vary depending on the taxa and techniques employed.

Coming onto the classification techniques, most traditional machine learning methods approach the PVP task as binary classification, classifying either PVP or non-PVP [25] [49], structural or non-structural [11], capsid or non-capsid [18] etc. Based on recent literature reviews [30] [36] pertaining to the classification of phage virions using machine learning techniques, it has been observed that Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM) and Scoring Card Method (SCM) were commonly used algorithms that demonstrated satisfactory classification accuracy ranging from 70% to 80% on test sets. With room for improvement in classification results for phage sequences with low sequence conservation, deep learning methods came into popularity, demonstrating better results over traditional machine learning methods. Some of these models aim to solve the binary classification of PVPs, such as VironFinder [32], iVIREONS [10] while others, like DeepPVP [40], and PhaANN [26], perform as multi-class classifiers by first identifying PVP vs. non-PVP sequences, followed by classifying the PVP into different structural groups. These deep learning models showcase better performance in terms of accuracy while simultaneously incorporating more PVP classes for the classification task.

2.2 Prediction Uncertainty for Phage Proteins

Although these deep learning models provide impressive accuracy, no study has yet been conducted on the impact of uncertainty in phage virion classification. Studies show that deep learning methods often give overconfident results [13] which might lead to high uncertainty in situations previously unseen by the model. On a contrasting note, increased accuracy of prediction is at times positively correlated to the model's uncertainty [29]. The aforementioned uncertainty can be categorized into two distinct types: Aleatoric Uncertainty, which is centered around the data, and Epistemic Uncertainty, which is centered around the model. The nature of the two uncertainties is illustrated in Figure 2.2.

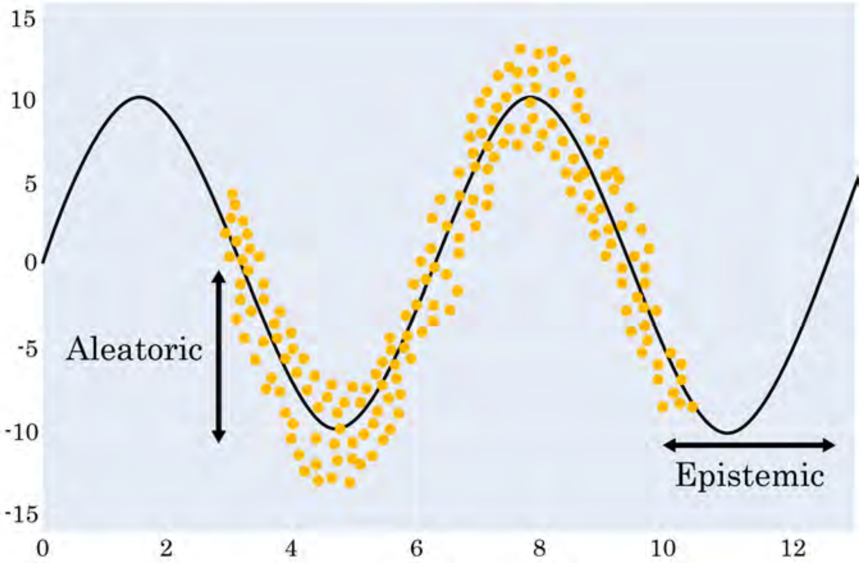


Figure 2.2: A schematic view of aleatoric and epistemic uncertainty in prediction. [31]

These uncertainties are particularly worrisome in safety essential scenarios, such as medical diagnosis and drug engineering, where result reliability is of utmost importance [34]. In such cases, it is more relevant to know whether a particular test result is accurate rather than knowing the average accuracy of a set of test cases, which can be ensured by analyzing the uncertainty trend of the model. Rather than having point estimates for a task, uncertainty analysis attempts to deduce the variance over the distribution of the data in consideration. This analysis provides insight into both the reliability and accuracy of the model. It is important to note that a high likelihood probability for a specific output does not necessarily indicate the model's confidence in that outcome. Although much work has been done in uncertainty analysis for deep learning frameworks, the tasks dealt with are primarily based on either computer vision, image processing, or natural language processing targeting human language. In cases of protein sequences depicting biological language, studies regarding uncertainty has been a field yet to be explored. Especially for PVP, where the sequence conservation is low and training data is going through a specific set of curations, there is a reasonable possibility of model uncertainty. Investigating this uncertainty can lead to the building of better models with higher robustness, along with providing a validation standard as to which kinds of sequences tend to give better accuracy but with lower confidence, needing further assessment before annotation.

2.3 Problem Statement

Newly sequenced phage virion proteins have heightened the need for efficient and accurate computational classification techniques. An essential aspect of this task involves refining the sequence representation schemes before training the models in order to extract valuable local and global information. Previously, feature extraction for phages has primarily focused on one-dimensional data. In this approach, the numerical representation of the strings is used as input for classification models, or numerous characteristics are manually derived based on the sequence composition, as mentioned earlier. While these methods yield satisfactory outcomes, the potential for improved prediction through increasing dimensionality remains largely unexplored. The sole task identified in this context is the utilization of [47], which employs the k-mer based FCGR encoding. However, this encoding method restricts the representation of spatial information in the resulting images. image encodings have been developed for DNA sequences that have yielded excellent outcomes in their specific applications through the utilization of the classical DNA-Walk method. Thus, it is a compelling and significant question whether such an encoding can be extended for the intricate protein sequences, specifically the phage family, in order to yield more accurate prediction outcomes for PVPs.

Furthermore, in the previous studies on PVP categorization, the uncertainty component of the predictive models used was not taken into account. In literature, approaches for analyzing uncertainty in textual data mostly focus on natural languages and overlook the inclusion of biological sequences. However, it is important to acknowledge that biological sequences possess distinct linguistic properties and can be considered a kind of language in their own right [37]. The predictions on these sequence data might depict uncertainty due to wrong model parameter initialization or an imbalance of train-test distribution leading to inconsistent or wrong outcomes with high confidence, which might be critical in cases where phage classification is needed for disease diagnosis or drug synthesis. The extensive curation required for such datasets might also lead to uncertainty, giving unreliable output likelihoods if test sequences are unlike anything during training. Thus, it is imperative to assess the degree of uncertainty in the process of PVP prediction. This evaluation provides insight, as depicted in Figure 2.3, into the confidence level of prediction methods for specific data samples.

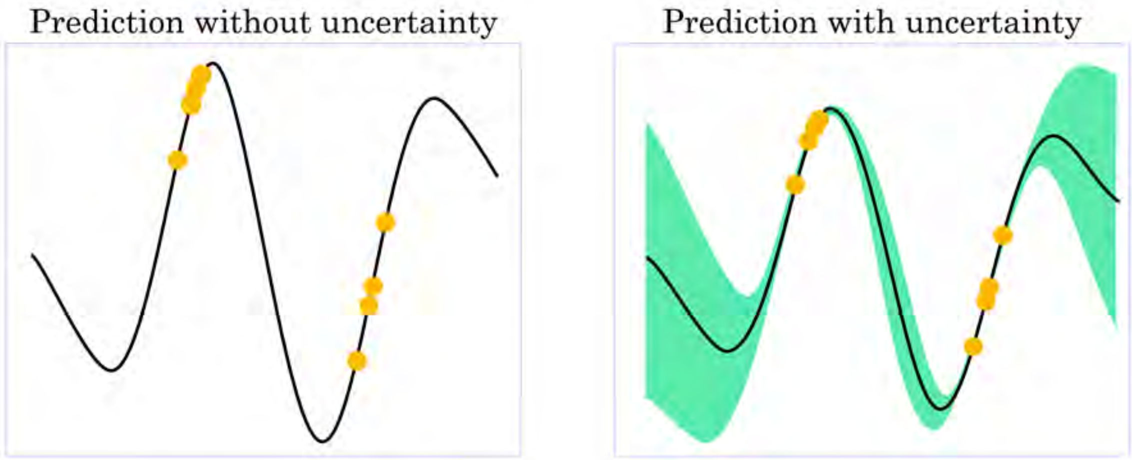


Figure 2.3: Uncertainty of prediction

2.4 Research Objective

In the preceding chapter, we examined the computational methodologies employed in the categorization of phage virions using machine learning and deep learning techniques and discussed about various encoding schemes . Additionally, we explored the various ways utilized for quantifying uncertainty in the analysis of image and text-based data. However, the prior investigations revealed the following limitations or constraints:

- Lack of image based encoding exploration for protein family compared to DNA or RNA families.
- Existing image based encoding works for phage proteins are susceptible to spacial information loss
- The primary emphasis of uncertainty estimation studies lies in the realm of non-proteomic data.
- Lack of analysis pertaining to the evaluation of deep learning techniques in the context of both accuracy and uncertainty for PVP annotation.

2.5 Core Contributions

Given the existing limitations in the research on PVP and its encoding, our study makes significant contributions to the academic discussion by providing the following core contributions:

- Proposed a novel encoding strategy, "Knight Encoding" where the text based protein sequences are converted into image based data.
- Conducted deep learning based classification to evaluate proposed encoding algorithm.
- Explored areas of uncertainty quantification using Monte Carlo Dropout (MCD) for PVP classification on pre-trained image classification models.

2.6 Thesis Organization

This thesis is organized in a manner that offers a systematic framework for thoroughly examining study findings, methodologies, and significant observations. The first chapter of this thesis book provides an overview of the essential background knowledge necessary for comprehending the work, including explanations of concepts such as PVP and Uncertainty Quantification in Deep Learning. In addition, Chapter 2 provides comprehensive introductions, highlighting the main contributions of this study, research objectives, and problem description. Chapter 3 provides an explanation of existing works and methodologies for classifying PVP. Chapter 4, which is the integral part in our study discusses the unique encoding algorithm, used to convert protein sequences into images. In addition, Chapter 4 also discusses the utilization of deep learning-based image classification models and clarifies the process of uncertainty analysis employing Monte Carlo Dropout. All the findings can be located in Chapter 5. Chapter 6 contains the result discussion and a concise analysis of the limitations. Finally, Chapter 7 outlines the future work and concludes this study.

Chapter 3

Literature Review

3.1 Computational PVP Annotation Techniques

In the field of bacteriophage research, culture-based methodologies were widely employed until the advent of high-throughput sequencing technology. This technological advancement facilitated the rapid accumulation of sequence data, necessitating the development of efficient and cost-effective methods for sequence identification. Due to their labor-intensive and expensive nature, culture-based methods have resulted in a growing preference for the utilization of machine learning techniques in phage investigations [36]. Supervised learning methods, such as Random Forest, Naive Bayes, and Support Vector Machines, are commonly employed in the classification of phage virions, which involves prediction based on evidence or labeled data. In order to enhance the classification outcomes, researchers have explored the utilization of diverse deep learning architectures, including Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), and Transformers.

In the paper [27], Charoenkwan et al. introduced the use of a score card method (SCM) in conjunction with propensity scores of dipeptides for the purpose of detecting PVPs. The utilization of these strategies shown superior performance compared to other intricate classifiers such as SVM, as they offer enhanced interpretability regarding the physiochemical properties of the PVPs. The training dataset exhibited an accuracy of 92.52%, whereas the independent dataset shown an accuracy of 77.66%. The model demonstrated a significantly high degree of accuracy during training on the dataset. However, its performance experiences a substantial fall when evaluated on an independent dataset. Besides, Ding et al. [12] proposed a method called PVPred, which uses the analysis of variance (ANOVA) method along with incremental feature selection (IFS) techniques. This approach achieves an accuracy rate of 85.02% and 71.3% for the training and independent datasets, respectively, through the identification of an optimal feature set. In a previous study [22], a comparable methodology was employed to identify PVPs. The approach utilized a support vector machine (SVM) technique, incorporating ideal g-gap dipeptide composition. The identification process involved variance analysis (ANOVA) and the minimal-redundancy-maximum-relevance (mRMR) with iterative feature selection (IFS). The results demonstrated an accuracy of 87.95% on training datasets and 75.53% on independent datasets.

Furthermore, Charoenkwan et al. [28] introduced Meta-iPVP, a novel method that employs a distinctive technique for feature representation. This method incorporates four different machine learning methods to encode seven input features into a probabilistic matrix . Afterwards, the generated probabilistic matrix is employed as input for the SVM model to carry out the classification of PVPs, achieving an accuracy of 0.871 and a Matthews Correlation Coefficient (MCC) of 0.642. In a previous study [21], another SVM-based approach was employed to predict PVPs and non-PVPs'. This method used diverse sequence composition parameters, including dipeptide composition, atomic composition, and amino acid composition as data features. The results demonstrated an accuracy of 87% on the training datasets and 79.8% on the independent datasets. Additionally, the study conducted by Barman et al. [44] further ensemble learning techniques, concluding that the Gradient Boosting Classifier (GBC) outperforms other methods in terms of accuracy, not only on test datasets but also on independent datasets. This is in contrast to other approaches that demonstrate strong performance solely on the test data. Feng et al. [11] employed an additional conventional machine learning approach wherein they proposed a method based on NB for the prediction of PVPs and non-PVPs. This method utilized proteins' main sequence features such as amino acid composition (AAC) and dipeptide composition (DPC). The researchers achieved an accuracy rate of 79.15%, a sensitivity rate of 75.76%, and a specificity rate of 80.77% when evaluating a training dataset. Nevertheless, the effectiveness of their approach on a separate dataset was not addressed. Zhang et al. [15] did an additional study that introduces a unique ensemble approach for predicting bacteriophage virion proteins using phage protein sequences. The approach employs hybrid feature spaces that integrate multiple methodologies, such as CTD (composition, transition, and distribution), bi-profile Bayes, PseAAC (pseudo-amino acid composition), and PSSM (position-specific scoring matrix). The logistic regression method based on RF, demonstrates superior performance in comparison to prior studies, attaining a sensitivity of 0.853, an accuracy of 0.831, and a MCC of 0.662 on the independent testing dataset.

Th research conducted by Seguritan et al. [10] introduce a novel artificial neural network framework for the categorization of PVP via deep learning approach for the very first time. The model initially distinguishes the structural protein from the non-structural proteins by utilizing a topology consisting of 20 input nodes, 90 hidden layer nodes, and a single output node. This approach achieves an accuracy of 86.5% in correctly predicting the protein type among a dataset of 12,000 protein sequences (6,000 structural and 6,000 non-structural sequences), which were obtained from the GenBank non-redundant database. The inclusion of estimated isometric points as a feature in ANNs further lead to improved classification of proteins into main capsid and tail proteins, surpassing the accuracy achieved by the structural ANN. The model results were additionally validated using in vivo gene assembly, and the resulting structures were compared to the predicted models. A different deep learning approach, referred to as VirionFinder [32], use the biological characteristics of amino acids as encodings in order to predict entire PVPs. This is achieved by utilizing a 1D convolutional neural network. Cantu et al. developed an artificial neural network (ANN) model called PhANN, which represents one of the initial attempts to expand the binary PVP classification problem into a multi-class prediction task [26]. The study employs a total of 11 unique artificial neural networks (ANNs) for training purposes. These ANNs are trained using a specific subset of diverse sequences with various features, such as the frequency of ”side chains” of 2-mers (consisting of 49 features) and 3-mers (comprising 343 characteristics). Additionally, a 12th ANN is trained utilizing all 11,201 features constructed. Each artificial neural network (ANN) in the study was composed of an input layer, two hidden layers consisting of 200 neurons each, and an output layer including 11 neurons. The purpose of these networks was to classify 10 sub-classes of PVP protein, as well as categorize the remaining classes under the designation of ”other” within the structural categorizations. Fang et al. completed a study that proposes the use of a one-dimensional convolutional neural network (1-D CNN) architecture, known as DeepPVP, for the classification of binary and multi-class phage virions [40]. The primary component of this architectural design initially categorizes a sequence as either PVP or non-PVP. Subsequently, an expanded module is employed to classify the PVPs into ten major classes, such as major capsid, baseplate, and tail fiber, among others. This approach is comparable to PhANNs, but it achieves enhanced accuracy in classification. The latest research on this subject was undertaken by Shang et al [47]., who introduced a unique sequence encoding method utilizing chaos game representation (CGR). The resulting images generated from this approach are then inputted into a high-performance vision transformer classifier. In contrast to PhANN and DeePVP, PhaVIP employs a classification system that categorizes PVPs into seven prevalent structural sub-classes, as opposed to ten sub-classes used by the aforementioned methods. Despite this reduction in sub-classification granularity, PhaVIP demonstrates superior precision, recall, and F-1 scores when compared to existing deep learning approaches. The frameworks PhANN, DeePVP, and PhaVIP have been identified as the sole models capable of performing multi-class classification of virion proteins with state-of-the-art outcomes. A concise comparison of these three models is presented in Table 3.1. The deep learning approaches mentioned in this study are regarded as the benchmark for our research on uncertainty, and their intricate mechanics is further examined in the subsequent sections of the paper.

Tool Name	Model Used	Precision	Recall	F1 Score
PhANN	ANN	0.76	0.91	0.83
DeePVP	1-D CNN	0.96	0.88	0.92
PhaVIP	ViT	0.90	0.91	0.90

Table 3.1: Result analysis of existing deep learning approaches

3.2 Encoding Schemes for Protein Sequences

Both machine learning and deep learning models have been developed for the purpose of classifying protein sequences, with a particular focus on phage virion proteins. Nevertheless, despite sharing a common objective, these models employed distinct encoding strategies. Certain researchers have attempted to forecast protein sequences by utilizing visual depictions of protein sequences using images, while others have employed numerical values as a basis for prediction. The next section provides an explanation of the utilized encoding techniques.

Project Name	Encoding Technique	Encoding Type	Model
PhaVIP	FCGR	Image	ViT
PhANN	K-mer frequency count	Text	ANN
DeePVP	One-hot	Text	CNN
PVPred-SCM	N/A	N/A	SCM
This Work	Knight Encoding	Image	CNN

Table 3.2: List of classifying models along with respective encoding techniques

FCGR: In PhaVIP [47] before feeding the sequence to the ViT or Vision Transformer, protein sequence have been converted to a CGR (Chaos Game Representation) image. In essence, CGR is a generalized scale-independent Markov probability table for the sequence [23]. FCGR is a variation of Chaos Game Representation where they use the k-mer frequency count of a sequence and make probability distribution to make a graph or image.

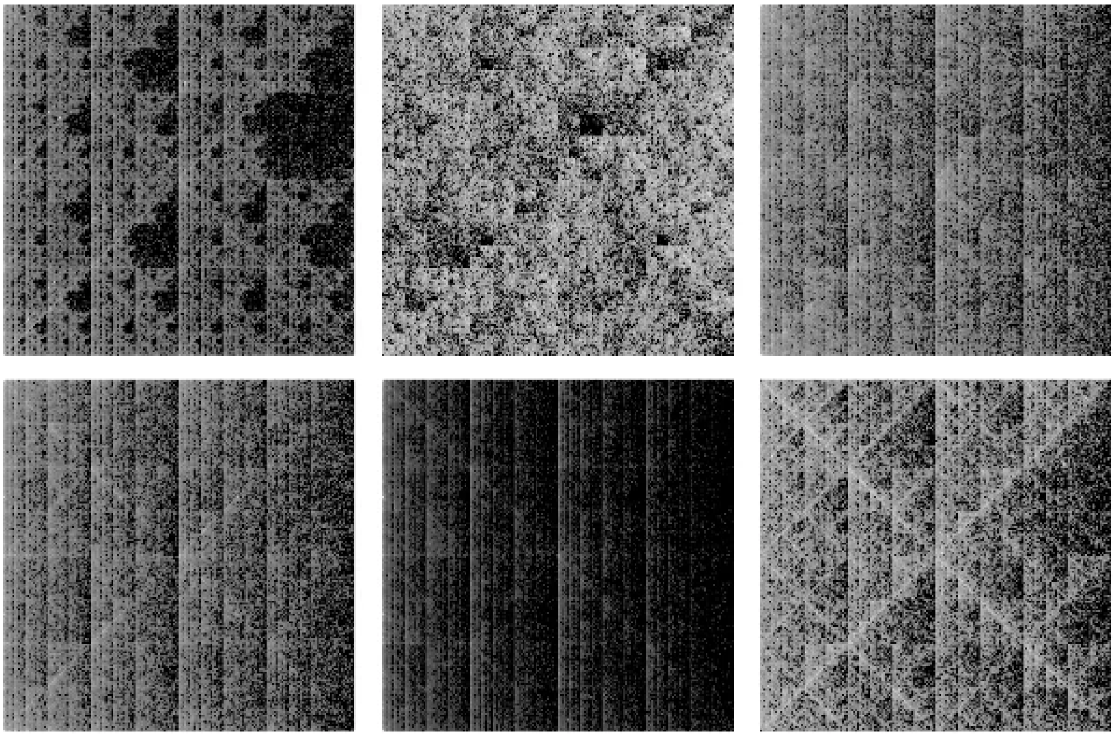


Figure 3.1: CGR representation of Sequence [23]

K-mer: A k-mer refers to a contiguous subsequence of length k within a given string or sequence. K-mers refer to subsequences that are derived from a particular sequence. In order to obtain all k-mers from a given sequence, it is necessary to extract the initial k characters, afterwards shifting by a single character to generate the subsequent k-mer, and repeating this process iteratively. Given a protein sequence ”MIGMD,” an analysis of its 3-mers reveals the presence of three distinct 3-mers inside the sequence.

```
Sequence: MIGMD
3-mer #0: MIG
3-mer #1:  IGM
3-mer #2:   GMD
```

In PhANN [26], they have used the composition of 2-mers/dipeptides (di), 3-mers /tripeptides (tri) or 4-mer/ tetrapeptide (tetra), or side chain groups (sc) to train ANNs and have trained 12 ANN model in total based on the k-mers. Moreover, the 12th ANN contains all the features of the previous composition of the k-mers.

SCM (Scoring Card Method): The scoring card method classifies the query sequence based on the comparison between the score of the protein and the threshold value [8]. In PVPred-SCM they developed a Scoring Card Method and an IGA-based (Interactive genetic algorithm) machine learning algorithm to classify the PVP proteins [27].

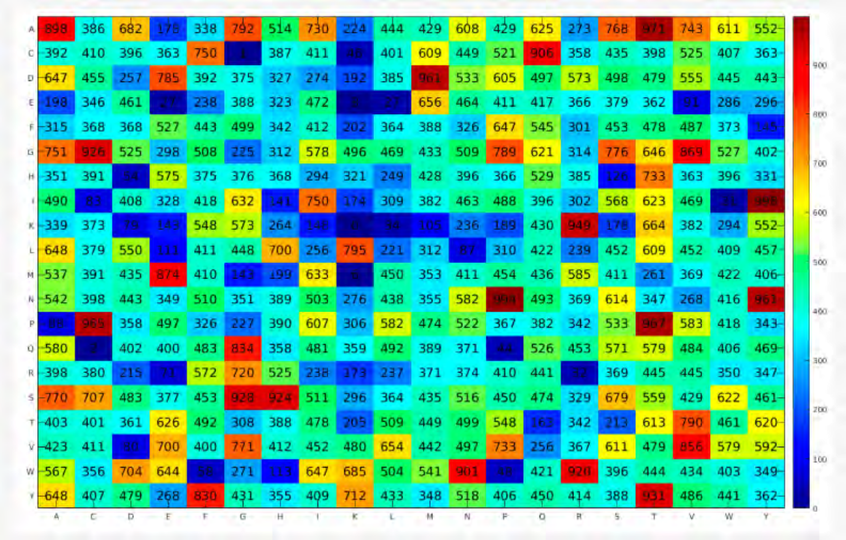


Figure 3.2: Scoring Card of Sequences [27]

One-Hot encoding: One-hot encoding is a popular encoding technique where categorical values are represented as numerical values this is important for training machine learning models. In DeePVP they first transformed the sequence into one-hot vectors or matrix then passed that matrix in 1D CNN [40].

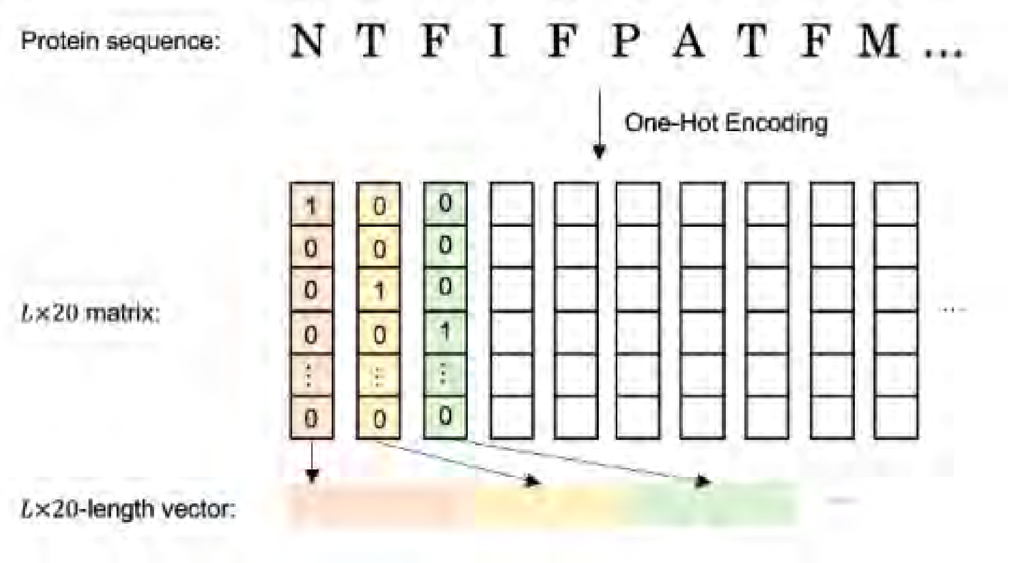


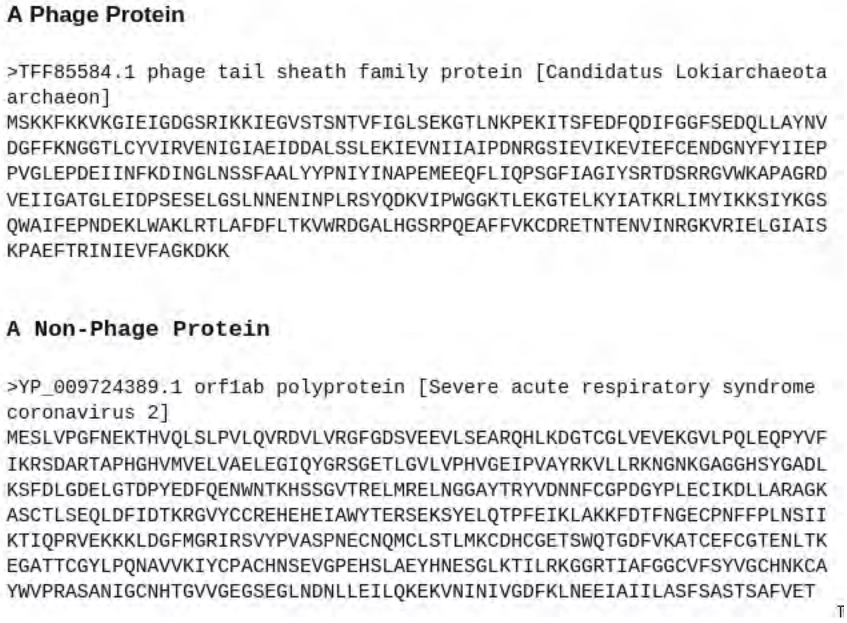
Figure 3.3: One-Hot encoding of a Sequence

Chapter 4

Methodology

4.1 Dataset Description

The datasets commonly employed for the detection of PVPs primarily comprise of raw protein sequences that have been categorized as either PVP or non-PVP, as shown in Figure 4.1. The sequences consist of strings of 20 amino acid characters with varying lengths. These sequences can be analyzed to extract different features, which can then be used to deduce the physiochemical properties of the PVP, aiding in its classification. In the context of deep learning tasks, as discussed previously, the sequences are encoded into various schemes without the need for further feature engineering for the purpose of PVP identification.



A Phage Protein

```
>TFF85584.1 phage tail sheath family protein [Candidatus Lokiarchaeota
archaeon]
MSKKFKKVKGIEIGDGSRIKKIEGVSTSNTVFIGLSEKGTLNKPEKITSFEDFQDIFGGFSEDQLLAYNV
DGFFKNGGTLCYVIRVENIGIAEIDDALSSLEKIEVNIIAIPDNRGSIEVIKEVIEFCENDGNYFYIIEP
PVGLEPDEIINFKDINGLNSSFAALYYPNIYINAPEMEEQFLIQPSGFIAGIYSRTDSRRGVWKAPAGRD
VEIIGATGLEIDPSESELGSLNNENINPLRSYQDKVIPWGGKTLEKGTELKYIATKRLIMYIKKSIYKGS
QWAIFEPNDEKLWAKLRTLAFDFLTKVWRDGALHGSRPQEAFFVKCDRETNTENVINRGKVRIELGIAIS
KPAEFTRINIEVFAGKDKK
```

A Non-Phage Protein

```
>YP_009724389.1 orf1ab polyprotein [Severe acute respiratory syndrome
coronavirus 2]
MESLVPGFNEKTHVQLSLPVLQVRDVLVRGFGDSVEEVLSEARQHLKDGTCGLVEVEKGVLPQLEQPYVF
IKRSDARTAPHGHVMVELVAELEGIQYGRSGETLGVLVPHVGEIPVAYRKVLLRKNGNKGAGGHSYGADL
KSFDLGDELGTDPYEDFQENWNTKHSSGVTRELMRELNGGAYTRYVDNNFCGPDGYPLECIKDLLARAGK
ASCTLSEQLDFIDTKRGVYCCREHEHEIAWYTERSEKSYELQTPFEIKLAKKFDTFNGECPNFFPLNSII
KTIQPRVEKKKLDGFMGRIRSVYPVASPNECNQMCLSTLMKCDHCGETSWQTGDFVKATCEFCGTENLTK
EGATTCGYLPQNAVVKIYCPACHNSEVGPEHSLAEYHNESGLKTILRKGGRTIAFGGCVFSYVGCHNKCA
YWVPRASANIGCNHTGVVGEGSEGLNDNLLEILQKEKVNINIVGDFKLNEEIAIILASFSASTSAFVET
                                                                      T
```

Figure 4.1: PVP and Non-PVP sequence example

4.2 Dataset Used in PVP Classification

For the purpose of this study, we have used one of the two existing benchmark datasets used in deep learning-based PVP classification. The dataset used in [26][40], which is the largest among all previous benchmark datasets, was built using annotations prior to June 2020. Subsequently, additional sequences have been incorporated, and in certain instances, sequences have undergone re-annotation. Because of these changes, the authors of [47] made a new dataset by getting the most up-to-date sequence annotations (through December 2022) from the RefSeq viral protein database, which has been used in this research. In the study conducted by [47], many data reconditionings were performed. The sequences that possessed low-confidence labels, which introduced ambiguity, were eliminated from the dataset. Subsequently, a search was conducted utilizing diverse keywords to identify and extract the structural proteins from the remaining sequences. The non-structural proteins were collected through a search process that involved identifying enzymes with names ending in the suffix 'ase'. The CD-hit algorithm was employed to identify clusters of sequences exhibiting a similarity threshold of 90%. The longest sequence within each cluster was selected as the representative sequence. The ultimate database comprises a total of **35,213** PVP sequences and **46,883** non-PVP sequences with varying lengths. The PVP sequences are further categorized into eight groups, curated for the purpose of conducting multi-class classification. An illustration of overall the class breakdown and the multiclass data count is shown in Figure 4.2 and Table 4.2 respectively.

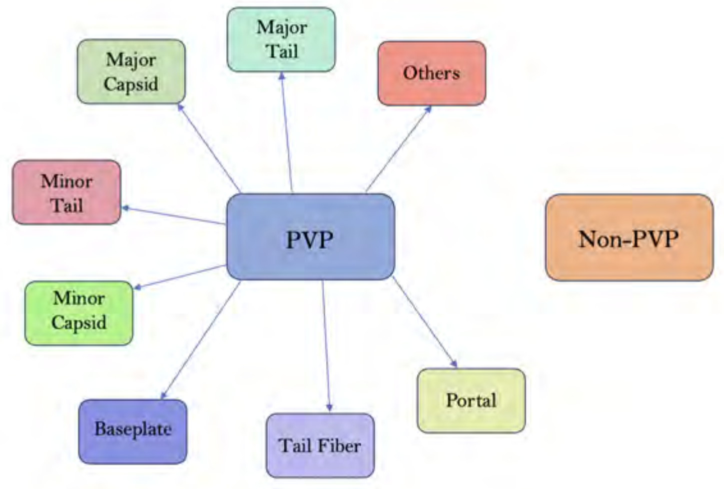


Figure 4.2: Illustration of Dataset Classes

Primary Classes	Number of Entries
PVP	35,213
non-PVP	46,883

Table 4.1: Binary-Class Data Entries

PVP classes	Number of Entries
Baseplate	3362
Portal	2770
Tail Fiber	2305
Major Capsid	2443
Minor Capsid	398
Major Tail	5083
Minor Tail	1458
Others	17385

Table 4.2: Multi-Class Data Entries of PVP

4.3 Proposed ‘Knight’ Encoding technique

As discussed in the earlier chapters, protein sequences lack image-based encoding in comparison to DNA sequences. DNA and proteins are fundamental components of living organisms, with DNA consisting of nucleotide building blocks (A, T, C, G) and proteins being composed of 20 distinct amino acids. Random walk or DNA walk has been commonly employed for encoding DNA in order to analyze the sequences [2] and predict sequence families with high accuracy. Akbar et al. [39] employ DNA-walk in their research to classify viral genomes using CNNs. They suggest that this encoding serves as a substitute for k-mer based encodings, like FCGR, which tend to have spacial information loss of encoded sequences. They also assert the use of such method to be feasible for other sequence types like RNA and proteins, although no such work has existed up until now. Considering this, we have devised a novel walk-based encoding technique only for protein sequences to assess its efficacy in characterizing protein families.

This encoding approach utilizes the concept of polar coordinates to interpret a protein sequence. Each amino acid in the sequence is assigned an angle value such that their summation is equal to 360°. This angle indicates the direction that an amino acid will walk towards from its current position. The letters (or amino acids) are depicted using a distinctive color-coded marker or circular point for further distinction, and positioned on the image according to their corresponding angle values and distance of movement, known as the radius. The equations 4.1, 4.2 are used to find the x direction and y direction movement for a residue to be encoded, starting from its current position.

$$x = r \times \cos(\theta) \tag{4.1}$$

$$y = r \times \sin(\theta) \tag{4.2}$$

For understanding purpose, the entire encoding algorithm can be divided into three major parts:

- **Definitions** - List of amino acids and their respective color codes.
- **Angle Calculation** - Angle assignment to each amino acid residues.
- **Placement** - Position and order calculation for placing circular encoded points corresponding to each amino acid.

Definitions

A list is defined that includes the letter representation of each amino acid, along with a complimentary dictionary that maps each amino acid to a corresponding color. These are subsequently utilized for the computation of angles and placement of points.

List of amino acids for the encoding,

```
self.amino_acids = ['A', 'C', 'D', 'E', 'F',
                    'G', 'H', 'I', 'K', 'L',
                    'M', 'N', 'P', 'Q', 'R',
                    'S', 'T', 'V', 'W', 'Y']
```

Amino acid to color mapping,

```
self.colors = {
        'A': (255, 0, 0),
        'C': (255, 255, 0),
        'D': (0, 234, 255),
        'E': (170, 0, 255),
        'F': (255, 127, 0),
        'G': (191, 255, 0),
        'H': (0, 149, 255),
        'I': (255, 0, 170),
        'K': (237, 185, 185),
        'L': (185, 215, 237),
        'M': (231, 233, 185),
        'N': (220, 185, 237),
        'P': (185, 237, 224),
        'Q': (143, 35, 35),
        'R': (35, 98, 143),
        'S': (143, 106, 35),
        'T': (107, 35, 143),
        'V': (115, 115, 155),
        'W': (204, 204, 204),
        'Y': (0, 64, 255)
        }
```

Angle Calculation

- We have employed the structure of a 20-sided polygon called ‘Icosagon’ to represent the 20 amino acids.The amino acids are evenly distributed throughout the 20 vertices of the Icosagon. Each point is separated from the others by a constant angle of 18 degrees (360°/ 20 = 18°), as seen in the Figure 4.3.

- The angles for each letter in the amino acid are determined based on the index position of the amino acids in the list mentioned previously. For instance, ‘C’ is at index 1 and ‘G’ is at index 5.

  So for ‘C’, the associated angle is,

  $$\text{index of C} = 0$$

  $$\theta = 1 \times 18^\circ = 18^\circ$$

  Similarly for ‘G’ the angle will be,

  $$\text{index of G} = 5$$

  $$\theta = 5 \times 18^\circ = 90^\circ$$

  As we shall be using polar coordinate formulas in our study, the angles are converted from degree to radians.

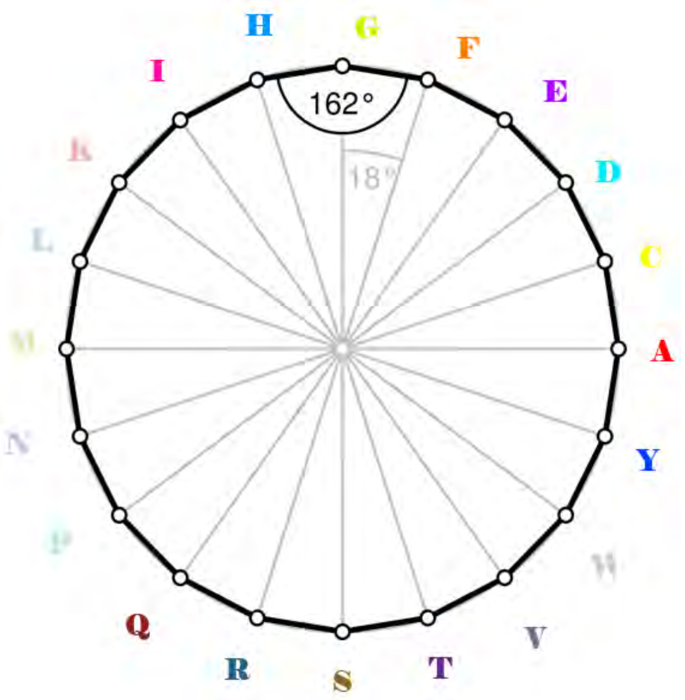


Figure 4.3: Icosagon

Placement

- If the dimensions of the image are defined as $M \times M$, then the encoding will start from the origin, which is located at the coordinates $(x, y) = (M/2, M/2)$, or in other words, from the middle of the image.

- For the first amino acid (or the first letter of our sequence), $(x, y) = (M/2, M/2)$ or the center of our image will act as the starting point i.e current coordinates.

- Using the fixed radius and corresponding angle of an amino acid, the horizontal and vertical displacement (x', y') of the encoding is determined using 4.1 4.2, relative to the current coordinated (x, y), as shown in Equations. 4.3 4.4.

$$x = x + x\prime \tag{4.3}$$

$$y = y - y\prime \tag{4.4}$$

  A circular point will be positioned at the coordinates (x, y) to represent the amino acid being encoded, with its corresponding color. The polar coordinate radius and point size remain consistent for all amino acids. These specifications are listed in Table 4.3.

- Subsequently, the next character of the protein sequence will utilize the coordinates (x, y) of the previous character as its starting point. A new horizontal and vertical shift (x', y') will be generated based on the amino acids associated angle. Using these new shift values (x', y'), a new set of coordinates (x, y) are calculated for placing the current amino acid character.

Image Resolution	**Radius**	**Point Size**
512×512	15	2
640×640	15	2

Table 4.3: Parameters that were used to encode each sequence

- The remaining characters in the sequence will continue to follow the previously specified technique until the entire sequence is encoded as shown in Figures. 4.4, 4.5, 4.6, 4.7. If a point hits the boundary of the image during encoding, it will begin encoding from the center of the image at, $(x, y) = (M/2, M/2)$.

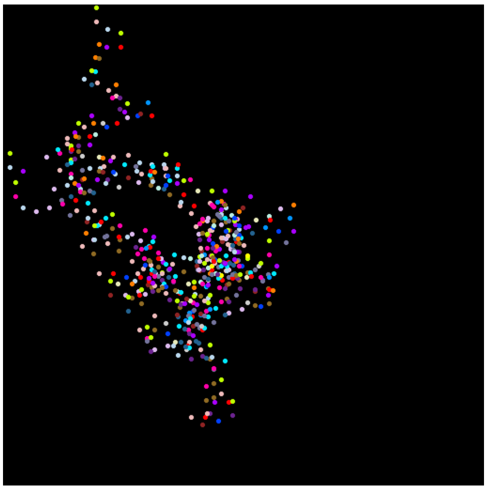


Figure 4.4: PVP(YP_009900749.1 minor capsid protein [Lactococcus phage 62503])

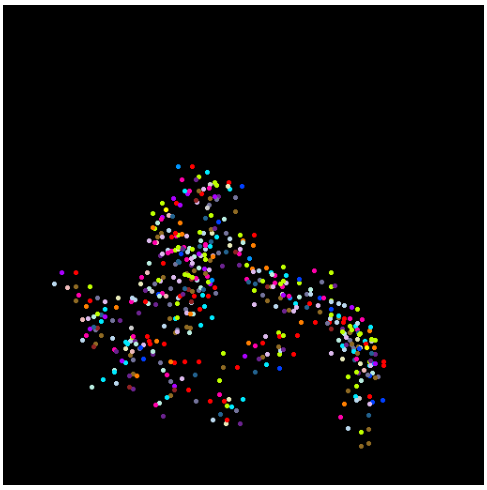


Figure 4.5: PVP (YP_009836980.1 minor tail protein [Gordonia phage Adgers])

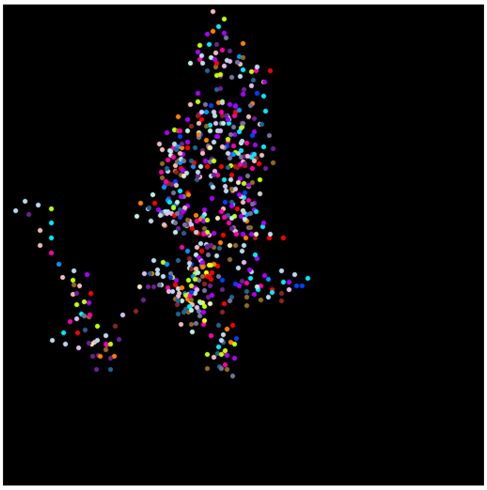


Figure 4.6: non-PVP(YP_009847768.1 (NAD(+)) DNA ligase [Vibrio phage USC-1])

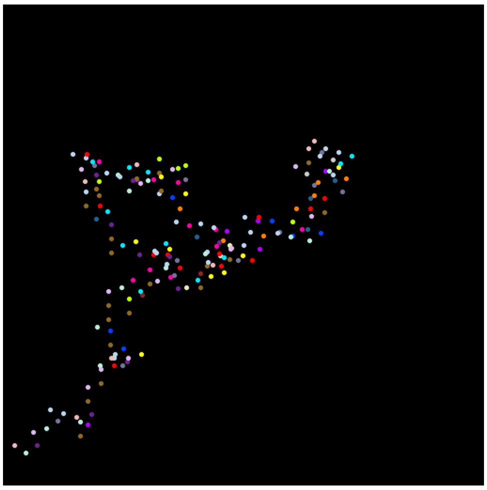
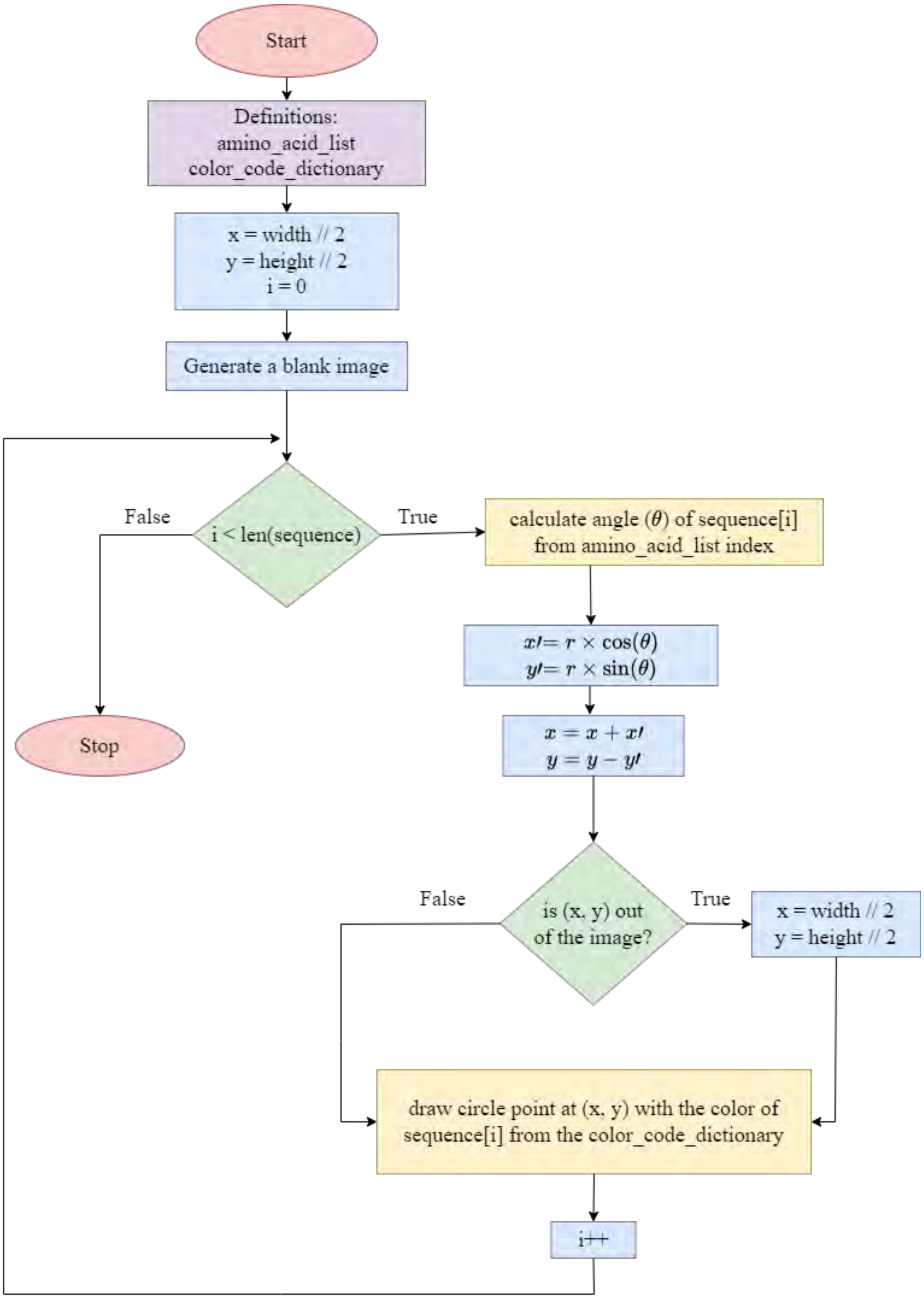


Figure 4.7: non-PVP(YP_009621118.1 1,4-dihydroxy-6-naphthoate synthase [Vibrio phage Ceto])

4.4 Algorithm Flowchart

The code implementation of this encoding can be found on this link.

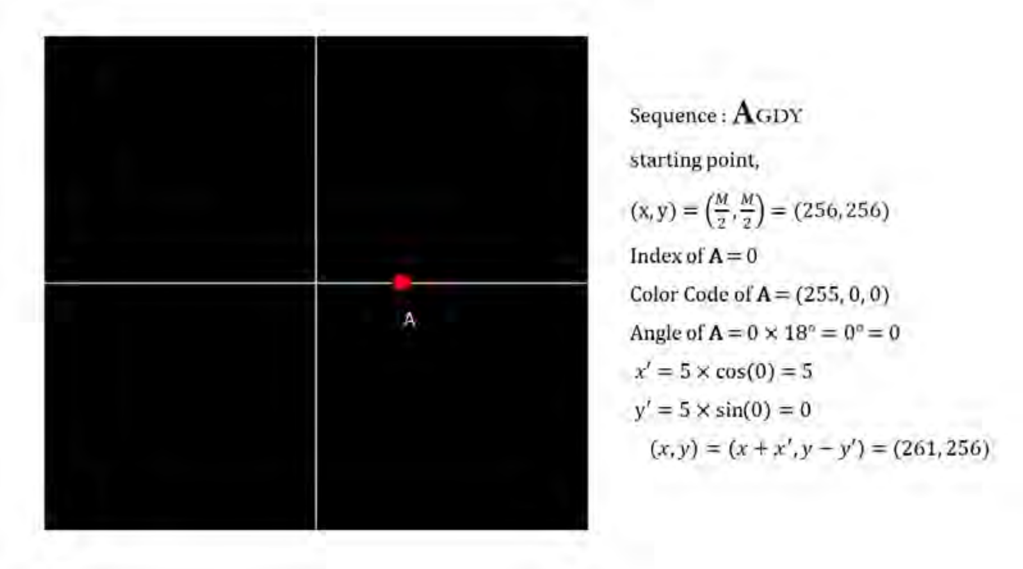


Start
Definitions:
amino_acid_list
color_code_dictionary
x = width // 2
y = height // 2
i = 0
Generate a blank image
i < len(sequence)
False
True
calculate angle (θ) of sequence[i] from amino_acid_list index
$x\prime = r \times \cos(\theta)$
$y\prime = r \times \sin(\theta)$
$x = x + x\prime$
$y = y - y\prime$
Stop
is (x, y) out of the image?
False
True
x = width // 2
y = height // 2
draw circle point at (x, y) with the color of sequence[i] from the color_code_dictionary
i++

Figure 4.8: Flowchart of the ‘Knight Encoding’ algorithm

4.5 Demonstration

Below is a demonstration of the encoding algorithm operating on a 512 by 512 image, utilizing the sequence ‘AGDY’.



Sequence : **A**GDY

starting point,

$(x, y) = \left(\frac{M}{2}, \frac{M}{2}\right) = (256, 256)$

Index of **A** $= 0$

Color Code of **A** $= (255, 0, 0)$

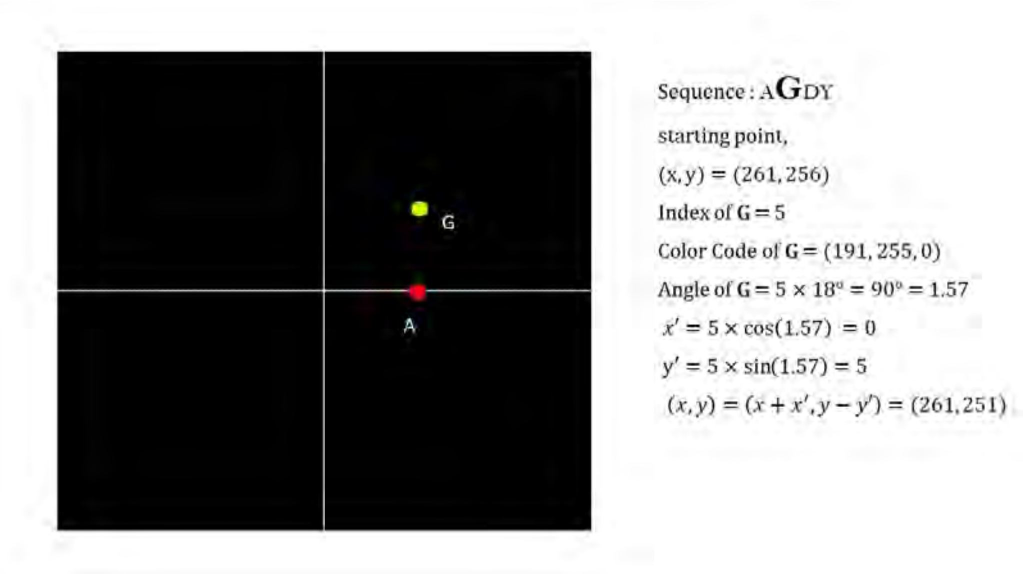
Angle of **A** $= 0 \times 18^{\circ} = 0^{\circ} = 0$

$x' = 5 \times \cos(0) = 5$

$y' = 5 \times \sin(0) = 0$

$(x, y) = (x + x', y - y') = (261, 256)$

Figure 4.9: Encoding A in the sequence ‘AGDY’



Sequence : A**G**DY

starting point,

$(x, y) = (261, 256)$

Index of **G** $= 5$

Color Code of **G** $= (191, 255, 0)$

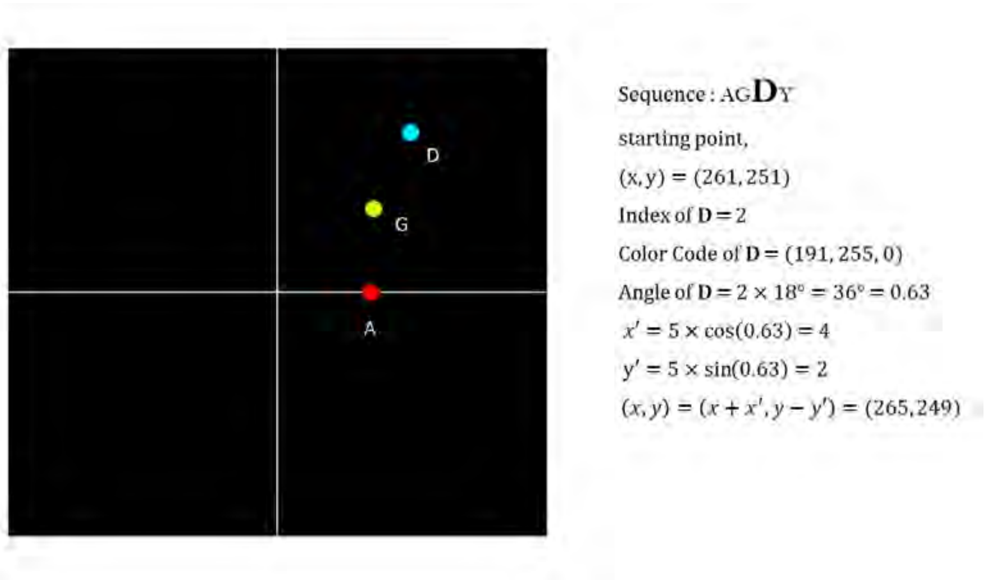
Angle of **G** $= 5 \times 18^{\circ} = 90^{\circ} = 1.57$

$x' = 5 \times \cos(1.57) = 0$

$y' = 5 \times \sin(1.57) = 5$

$(x, y) = (x + x', y - y') = (261, 251)$

Figure 4.10: Encoding G in the sequence ‘AGDY’



Sequence : AG**D**Y

starting point,

(x, y) = (261, 251)

Index of **D** = 2

Color Code of **D** = (191, 255, 0)

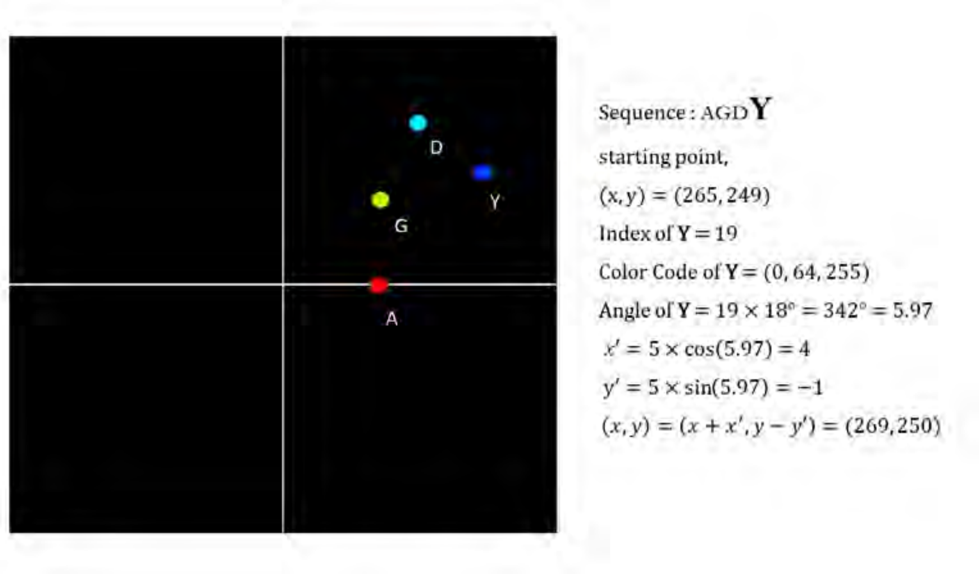
Angle of **D** = 2 × 18° = 36° = 0.63

$x' = 5 \times \cos(0.63) = 4$

$y' = 5 \times \sin(0.63) = 2$

$(x, y) = (x + x', y - y') = (265, 249)$

Figure 4.11: Encoding D in the sequence 'AGDY'



Sequence : AGD**Y**

starting point,

(x, y) = (265, 249)

Index of **Y** = 19

Color Code of **Y** = (0, 64, 255)

Angle of **Y** = 19 × 18° = 342° = 5.97

$x' = 5 \times \cos(5.97) = 4$

$y' = 5 \times \sin(5.97) = -1$

$(x, y) = (x + x', y - y') = (269, 250)$

Figure 4.12: Encoding Y in the sequence 'AGDY'

4.6 Deep Learning Models

Deep learning is a specialized domain within the study of machine learning that emphasizes the use of artificial neural networks (ANNs) with many layers, often referred to as deep neural networks (DNNs). The word "deep" is used to denote the existence of several concealed layers inside these networks. In contrast, the neural network is a computer model that draws inspiration from the intricate structure and intricate functionality of the human brain. The aforementioned model is a machine learning technique used in a range of applications within the fields of artificial intelligence and data analysis. In the current epoch characterized by the accessibility of cost-effective next-generation sequencing technology, we are confronted with a substantial increase in the amount of new phage genome sequences. Therefore, we have used image based encoding of PVP as our dataset which also follow the criteria set out by the International Committee on Taxonomy of Viruses (ICTV). Here we leveraged two of the most well performing architectures. The fundamental architecture of these models are discusses below.

4.6.1 Convulational Neural Network

The Convolutional Neural Network (CNN) is a specialized kind of artificial neural network that has been developed exclusively for the purpose of processing and evaluating visual input, including but not limited to photos and videos. They are well recognized as one of the most often used models in contemporary usage. This computational model, known as a neural network, utilizes a modified version of the multilayer perceptron. It incorporates one or more convolutional layers, which may be either fully linked or pooled. Moreover, the convolutional layers provide feature maps that capture certain regions of the picture, which are then partitioned into rectangular segments and sent for nonlinear processing. CNNs have shown remarkable efficacy in several domains, including but not limited to picture categorization, object identification, face recognition, and image synthesis. The success of computer vision applications may be attributed to its inspiration from the structure and function of the human visual system. The model has many benefits, including a notable degree of accuracy in picture recognition tasks, the ability to autonomously identify significant characteristics without human intervention, and the implementation of weight sharing. CNNs have shown their efficacy in the categorization of phage virion proteins by capitalizing on their inherent capacity to autonomously acquire and extract pertinent information from protein sequences. The process is briefly explained below.

Initially the data is prepared through collecting appropriate benchmark datasets of protein sequences from phage virions, with each sequence associated with a class label indicating its functional or structural properties. Protein sequences are typically composed of amino acids represented by letters (e.g., “ACDEFGH...”). By encoding the sequence, they are inputted into the CNN. Common encoding methods include one-hot encoding or using pre-trained embeddings (Word2Vec, ELMO, BERT) to convert the sequences into numerical input. The Convolutional Layer is where this model is unique. This layer uses filters (also called kernels) to scan across the input sequences. Each filter slides across the input, computing a dot product with the image segments it covers. This process is repeated across the entire encoded image, generating feature maps that capture different patterns. An Activation Function is also needed here. Applying an activation function (ReLU - Rectified Linear Unit) element-wise to introduce non-linearity and capture complex patterns of our encoding. To reduce the dimensionality of the feature maps produced by the convolutional layers, pooling layers are used. Max-pooling is a common choice, where the maximum value within a small window is retained while the rest are discarded. Then the data is flattened to 1D vector. This vector is then fed into one or more fully connected layers. For output, softmax activation function is used to produce class probabilities, indicating the likelihood of each protein belonging to a specific class. Then the data is trained. Backpropagation and the Adam optimization algorithms are used to adjust the network's weights during training. Sometimes Hyperparameter Tuning is performed to gain better results. Experimenting with different hyperparameters, such as the number of convolutional layers, filter sizes, pooling strategies, and learning rates, to optimize the model's performance.

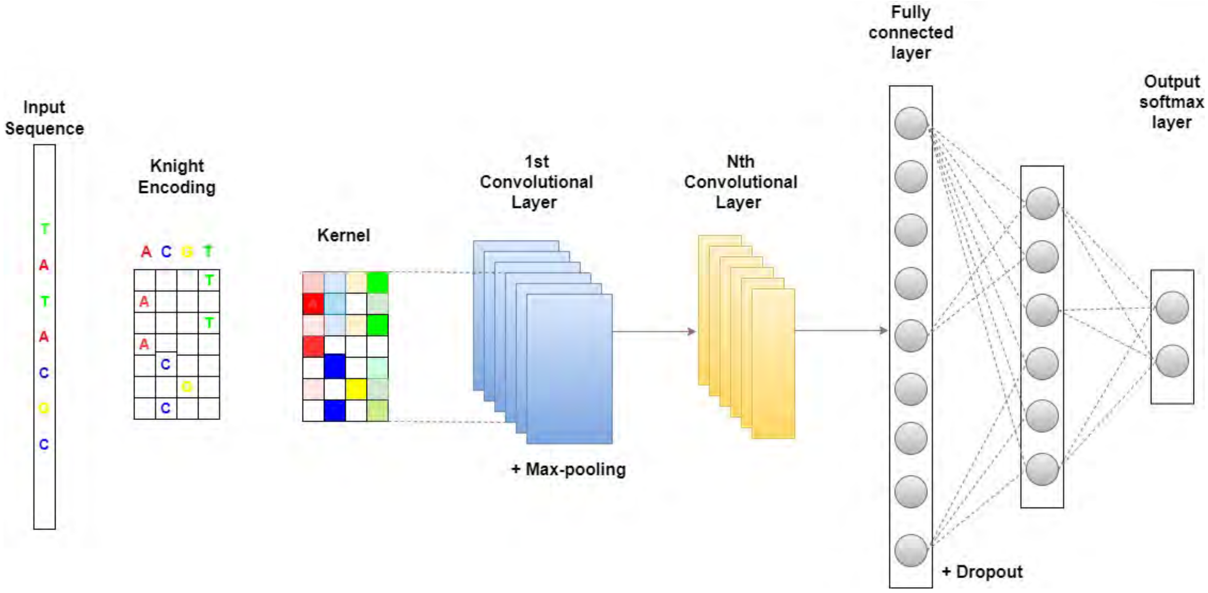


Figure 4.13: CNN architecture

This is how CNNs are used for phage virion protein classification. They excel at capturing local and hierarchical patterns in protein sequences, making them well-suited for tasks that involve recognizing motifs or features associated with specific protein functions or properties within phage virions [42].

4.6.2 Compact Convolutional Transformer

The Compact Convolutional Transformer (CCT) is a type of neural network architecture that combines elements of CNNs and Vision Transformers (ViTs) to process image data efficiently. This architecture is designed to address some of the limitations of both CNNs and ViTs when applied to computer vision tasks.

Now the classification of proteins using a CCT entails analyzing and categorizing the distinctive characteristics of proteins, which are usually depicted in the form of images obtained through techniques like electron microscopy, X-ray crystallography, or other protein visualization methods [43]. This process can be broken down into different steps which are briefly described below.

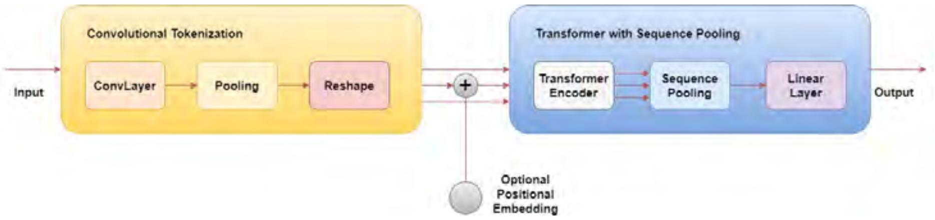

Convolutional Tokenization
Input
ConvLayer
Pooling
Reshape
Transformer with Sequence Pooling
Transformer Encoder
Sequence Pooling
Linear Layer
Output
Optional Positional Embedding

Figure 4.14: Compact Convolutional Transformer architecture main blocks

The first step is to prepare the data accordingly. After collecting the protein images they need to be processed. This encompasses activities such as standardizing, adjusting dimensions, and even enhancing the data to improve the variety of the training dataset. The CCT begins with one or more layers of a CNN. This section of the network is assigned with extracting specific features from the protein images. After this the generated feature maps are divided into patches.These patches are then flattened and linearly projected (similar to how patches are prepared in a standard Vision Transformer) to create a sequence of embeddings. Each embedding corresponds to a distinct part of the protein image [33]. To capture the global context the series of embeddings are loaded into the Transformer component of the CCT. The Transformer employs self-attention processes to analyze the complete sequence, effectively capturing global dependencies and connections among various components of the image. This is an essential step in analyzing the protein's image based structure and functions. The result of the Transformer is transmitted through a classification head, usually consisting of a few fully connected layers. Based on the learnt features, this portion of the network predicts the protein's class and outputs the final classification. The CCT model performs training using a dataset of protein images that have been labeled. It acquires the ability to identify patterns and features linked to various protein classes. The training process involves improving the weights of both the CNN and Transformer components in order to minimize the classification error. Following the training phase, the model is evaluated using an independent test dataset to measure its accuracy, precision, recall, and other relevant metrics. Ensuring that the model generalizes effectively to unfamiliar data is very important at this stage.

4.6.3 Pre-Trained Models

We employed CNN and CCT for our image based datset for classification. There are plenty of pre-existing pre-trained models. Instead of creating and training a model from scratch, we utilized these existing resources. After conducting experiments with several iterations of CNN and transformer architectures, we finally identified four models that exhibit the highest level of compatibility with our dataset and available resources.

After creating our new image dataset of the protein sequences using ‘Knight Encoding’, the next important task was to classify whether a sequence was PVP or non-PVP. Moreover if a PVP was detected, among the 8 sub-classes the question arised about the classification of the detected PVP. To predict these results Deep Learning Image classification models were necessary. Moreover, we focused on finding an image classification model that has fewer parameters but can achieve greater results. Because of that reason for these image classification tasks we have selected the below Pre-trained models,

- GoogLeNet
- CCT_7
- EfficientNet_V2_small
- MobileNet_V3

Each model was tweaked to suit our desired result and to fit the datasets. Among these GoogLeNet showed the most optimized result.

GoogLeNet:

```
=========================================================================
Layer (type (var_name))             Input Shape          Output Shape
=========================================================================
GoogLeNet (GoogLeNet)             [1, 3, 224, 224]     [1, 2]
BasicConv2d (conv1)               [1, 3, 224, 224]     [1, 64, 112, 112]
MaxPool2d (maxpool1)              [1, 64, 112, 112]    [1, 64, 56, 56]
BasicConv2d (conv2)               [1, 64, 56, 56]      [1, 64, 56, 56]
BasicConv2d (conv3)               [1, 64, 56, 56]      [1, 192, 56, 56]
MaxPool2d (maxpool2)              [1, 192, 56, 56]     [1, 192, 28, 28]
Inception (inception3a)           [1, 192, 28, 28]     [1, 256, 28, 28]
Inception (inception3b)           [1, 256, 28, 28]     [1, 480, 28, 28]
MaxPool2d (maxpool3)              [1, 480, 28, 28]     [1, 480, 14, 14]
Inception (inception4a)           [1, 480, 14, 14]     [1, 512, 14, 14]
Inception (inception4b)           [1, 512, 14, 14]     [1, 512, 14, 14]
Inception (inception4c)           [1, 512, 14, 14]     [1, 512, 14, 14]
Inception (inception4d)           [1, 512, 14, 14]     [1, 528, 14, 14]
Inception (inception4e)           [1, 528, 14, 14]     [1, 832, 14, 14]
MaxPool2d (maxpool4)              [1, 832, 14, 14]     [1, 832, 7, 7]
Inception (inception5a)           [1, 832, 7, 7]       [1, 832, 7, 7]
Inception (inception5b)           [1, 832, 7, 7]       [1, 1024, 7, 7]
AdaptiveAvgPool2d (avgpool)       [1, 1024, 7, 7]      [1, 1024, 1, 1]
Dropout (dropout)                 [1, 1024]            [1, 1024]
Linear (fc)                       [1, 1024]            [1, 2]
```

```
==========================================================================
Trainable params: 5,601,954
Non-trainable params: 0
Total mult-adds (G): 1.50
==========================================================================
Input size (MB): 0.60
Forward/backward pass size (MB): 51.62
Params size (MB): 22.41
Estimated Total Size (MB): 74.63
==========================================================================
```

4.7 Monte Carlo Dropout

4.7.1 How Uncertainty is Estimated using MCD

As mentioned before, we employ GoogleNet as our primary model for PVP classification. To assess the level of uncertainty in binary classification, we activate the models dropout layer with a dropout rate of 0.2% during the test phase. Consequently, we develop a method to classify data values into distinct groups based on sequence properties in order to analyze the level of uncertainty for each category. Unlike typical text based data, protein sequences do not reveal unique variability, apart from their class labels. The amino acid sequences might or might not display patterns necessary for classifying a certain string. The only diversity that is noticeable for these sequences is the length of these sequences. The data lengths for each class, PVP and non-PVP, range from hundreds to thousands of residues, as depicted in Figure 4.15 and Figure 4.16. Therefore, it can be a beneficial investigation for the models' robustness to determine whether the model is vulnerable to uncertainty based on the length of sequences. The train data, both PVP and non-PVP, were divided based on an equilibrium sequence length δ. This ensured that there was nearly equal amount of data with lengths less than or equal to δ and length greater than δ. The purpose of considering this equilibrium is to mitigate any data bias that may be present in the training data across various sequence lengths. The sequences that are smaller than δ are referred to as **short** sequences, while sequences that are larger than δ are categorized as **long** sequences. Table 4.4 presents this data distribution and δ values of PVP and non-PVP train data.

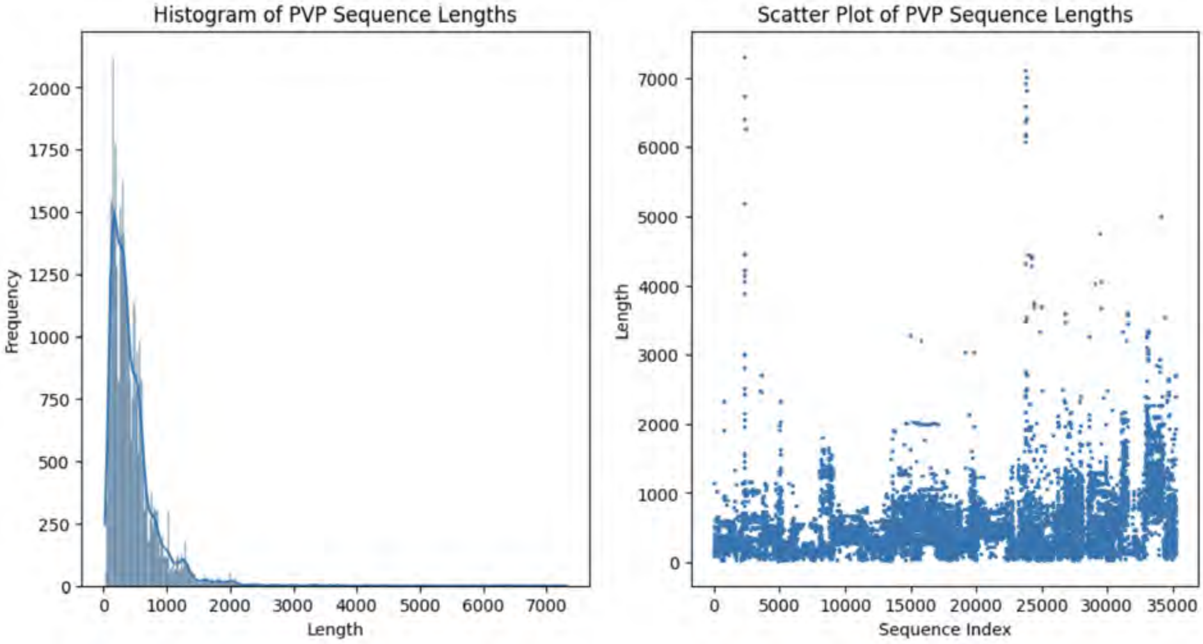


Figure 4.15: Length Distribution of PVP sequences

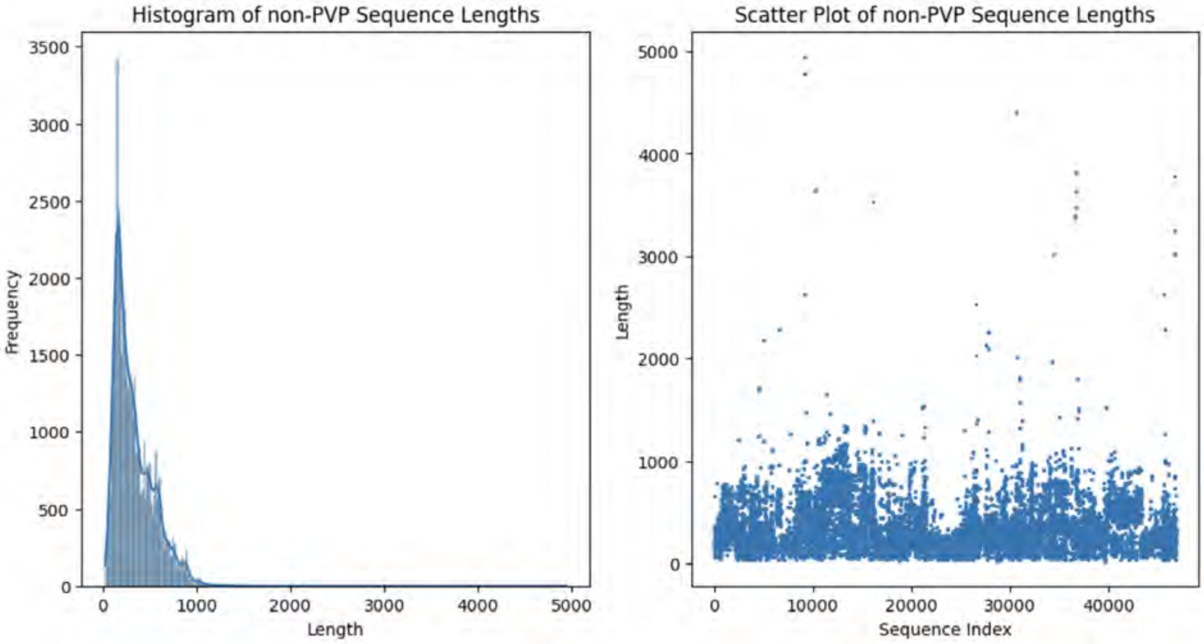


Figure 4.16: Length Distribution of non-PVP sequences

The data points are separated into four categories, PVP (less than 350), PVP (greater than 350), non-PVP (less than 275), and non-PVP (greater than 275). For every category, a total of 100 sequences are randomly chosen, and each sequence (after image encoding) is predicted 100 times using our dropout model. This provides a prediction distribution for every sequence within each category. The predictions passed through the softmax activation function, which transforms the raw logit output of our model into prediction probabilities. The mean and variance of

Class	#Sequences less than δ	#Sequences greater than δ	Total Sequences
PVP ($\delta = 350$)	12589	12060	24649
non-PVP ($\delta = 275$)	16502	16316	32818

Table 4.4: Data Distribution for PVP and non-PVP δ-values

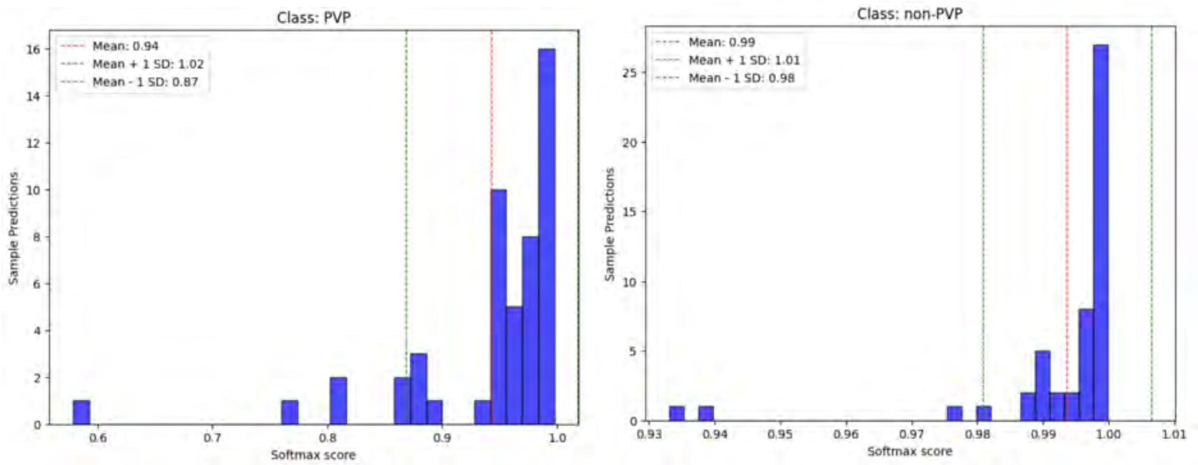


Figure 4.17: Softmax Distribution of PVP and non-PVP sample

these softmax probabilities from each category are then determined to compare the models' spectrum of uncertainty across the categories. As shown in Fig. 4.17 for a specific sample the we expect a distribution of probabilty, the x-axis here represents the softmax scores and y-axis indicate the number of predictions that gave a certain softmax score.

Typically, for correct prediction, the histogram should gather on the right side of the axis. This is because a softmax score >0.5 indicates that the model has classified the data as belonging to that particular class. On the contrary, predictions towards the left of the axis denotes incorrect predictions. The dispersion of the histogram i.e the variation in its predictions, on the other hand, signifies the level of confidence in the model's predictions . A prediction with reduced variance is more confident , while higher variance indicates greater prediction uncertainty in the model for that specific data.

4.8 Workflow

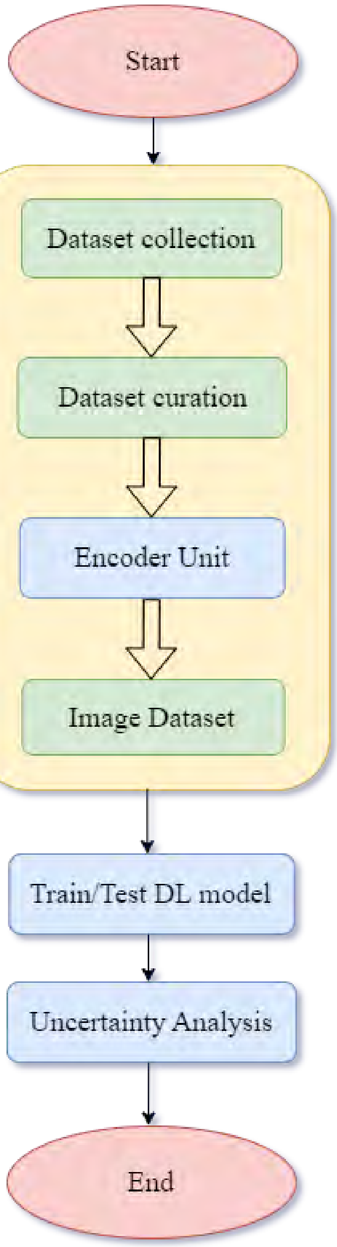


Figure 4.18: Research Workflow

Chapter 5

Experimental Analysis

5.1 Experimental Setup

TensorFlow, Keras, PyTorch and other Python libraries were used in the development of the experiment's training and testing procedures. To train and evaluate the models a computer with AMD Ryzen 7 3700X 8-core CPU, 16GB DDR4 and NVIDIA GeForce RTX 3060 Ti was used. Moreover, windows 11 Pro and Python interpreter version 3.11 was used as the platform to run the experiments.

5.2 Evaluation Metrics

Evaluation metrics are numbers that are used to determine how well a machine learning model or system works. These metrics enable us to assess our model's performance and compliance with task objectives in an unbiased way. The choice of an appropriate evaluation metric depends on the type of machine learning task that is being worked on. In our study we will be using the following metrics used for classification. Let's assume,

TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative

- **Accuracy:**
  Accuracy measures the proportion of correctly predicted instances out of the total instances in a classification problem but not suitable for imbalanced datasets, where one class significantly outnumbers the other.

  $$Accuracy = \frac{Number of Correct Predictions}{Total Number of Predictions}$$

  Here,
  Number of Correct Predictions = TP+TN
  and Total Number of Predictions = TP+TN+FP+FN

- **Precision:**
  Precision indicates the proportion of true positive predictions among all positive predictions made by the classifier. It helps measure the model's ability to avoid false positives. High precision means that when the model predicts a

positive instance, it's more likely to be correct.

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

- **Recall(Sensitivity):**
  Recall measures the proportion of true positive predictions among all actual positive instances in the dataset. It helps assess the model's ability to identify all relevant instances, minimizing false negatives. High recall means that the model can identify most of the positive instances.

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

- **F1-Score:**
  The F1-Score combines precision and recall into a single metric. It is the harmonic mean of these two values, providing a balance between false positives and false negatives. It's useful when you want to strike a balance between precision and recall, especially in situations with imbalanced classes.

  $$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

5.3 Results of Model Performance and Uncertainty Estimation

As discussed in the introduction, PVP classification task involves both binary and mult-iclass categorization. To gain a thorough grasp of the effectiveness of our suggested encoding, we have utilized deep learning as it has been shown in literature to outperform traditional machine learning methods in terms of prediction accuracy and the simplicity of features extraction. Rather than constructing an entirely new model from the scratch, we conducted training on our data by utilizing several existing pre-trained CNN and transformer models. We were particularly interested in evaluating the encoding efficiency on limited resources, which prompted us to choose our selected models. The following sections provide a concise summary of the performance of different architectures, along with a study of uncertainty for the best performing model.

5.3.1 Pre-Trained Model Performance

Task	Model	Params	Accuracy
Binary	Efficientnet_v2_small	21M	89.7%
	GoogLeNet	5.5M	90.0%
	CCT_7	4.5M	85.2%
	MobileNet_v3_small	2.5M	85.3%
Multiclass	Efficientnet_v2_small	21M	77.4%
	GoogLeNet	5.5M	78.8%
	CCT_7	4.5M	72.1%
	MobileNet_v3_small	2.5M	73.7%

Table 5.1: Model Comparisons on KnightEncoding

Task	Model	Params	Accuracy	F1 Score	Recall
Binary	GoogLeNet	5.5M	89.60%	89.87%	88.54%
Multiclass	GoogLeNet	5.5M	76.37%	76.37%	76.37%

Table 5.2: Evaluation results of the most optimal model

Work	Precision	F1 Score	Recall
PhaVIP	91%	90%	91%
PhANN	76%	83%	91%
DeePVP	97%	92%	88%
VirionFinder	44%	59%	91%
Meta-iPVP	53%	65%	82%
PVPred-SCM	39%	40%	41%
ProteoKnight	91%	90%	89%

Table 5.3: Comparative analysis of different approaches for Binary Classifcation

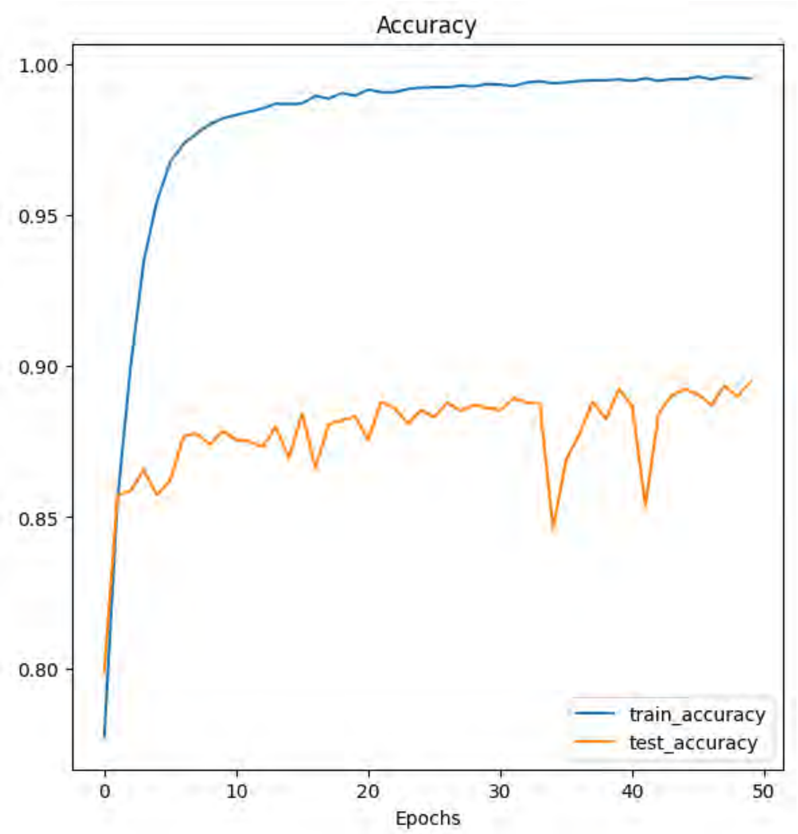


Figure 5.1: Accuracy graph for Binary Classification (GoogLeNet)

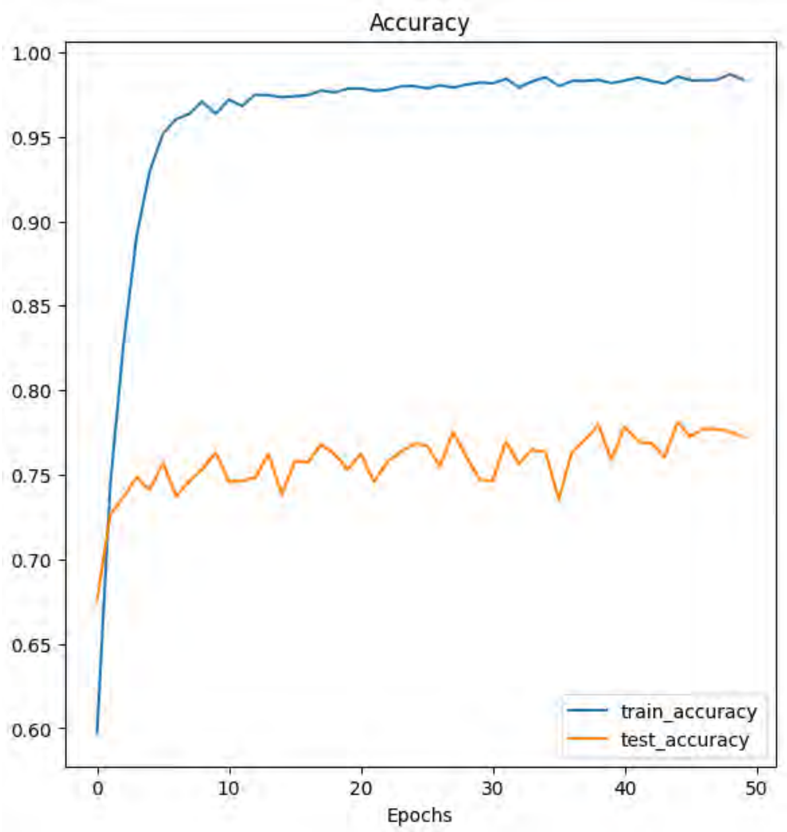


Figure 5.2: Accuracy graph for Multiclass Classification (GoogLeNet)

5.3.2 Uncertainty Analysis on PVP

Uncertainty estimations of deep learning models serve as a measure of the reliability of their predictions. As stated in the methodology, the variance of the prediction distribution during dropout passes is regarded as a standard uncertainty quantifying metric. Entropy (H) is another such commonly utilized metric in literature that focus on uncertainty, as demonstrated by Milanes et al. (2021) and Kendall et al. (2017) [35] [20]. In the context of machine learning, entropy is characterized as the disorder or uncertainty of the prediction model or simply the level of 'surprise' of the model [38] when it sees particular data sample(s). The lower the value of entropy, the better. Equation.5.1 is utilized in this paper to calculate the entropy of our binary classification samples, where P is the prediction probability for the particular data in concern.

$$Entropy = \left(P \times \log_2(\frac{1}{P})\right) + \left((1 - P) \times \log_2 \frac{1}{(1 - P)}\right) \tag{5.1}$$

In our uncertainty analysis, we employ both variance and entropy. Variance is computed for predictions across all four categories, while entropy is determined for specific sequence encodings that exhibit greater variance in comparison to others. For prediction, we primarily select the top-performing model, which is GoogLeNet.

We collect images from four distinct sequence categories: PVP short sequence, PVP long sequence, non-PVP short sequences, and non-PVP long sequences. The average and variance of the prediction distribution are graphed and compared for each category. During the class-based study, it was observed that the model exhibited lower variance when predicting non-PVP sequences compared to PVP sequences. Conversely, in terms of length, the model showed higher prediction variance for longer sequences compared to the shorter ones. We validated this uncertainty pattern by employing various dropout rates (0.1, 0.2, 0.3) and altering the sequences chosen for each of the four groups through numerous random shuffles. For purposes of illustration, all the figures displayed had been given the original dropout value of 0.2. The variance of each category is listed alongside the corresponding MCD value in Table 5.4. It can be seen that for all MCD values, the short sequences have lower variance than longer sequences, and non-PVP predictions display lower variance than PVP sequences. Based on these predictions, samples were selected from each category with the highest and lowest variance. The mean prediction entropy of these samples was then calculated and presented in Table. 5.5, from which it can be noted that the entropy values are lower in shorter sequences compared to longer ones. In the analysis based on class, non-PVPs have exhibited lower entropy.

Categories	**MCD 0.1**	**MCD 0.2**	**MCD 0.3**
PVP (short)	0.06801	0.08558	0.05536
PVP (long)	0.07276	0.09119	0.05926
non-PVP (short)	0.05914	0.07555	0.04860
non-PVP (long)	0.05994	0.07663	0.04900

Table 5.4: Variance for PVP and non-PVP with different dropouts

Categories	***Entropy* (High Var)**	***Entropy* (Low Var)**
PVP (short)	0.1259	0.1122
PVP (long)	0.3943	0.2111
non-PVP (short)	0.1398	0.1369
non-PVP (long)	0.1716	0.06539

Table 5.5: Entropy for PVP and non-PVP samples with high variance VS low variance

Chapter 6

Discussion

The objective of this work was to examine the practicality of utilizing a highly efficient DNA sequence encoding technique for the more intricate protein sequences in phage protein classification, and to evaluate if the outcomes would be equally effective as with DNA sequences. We utilized a variety of cutting-edge deep learning convolutional neural networks and transformer models to classify the encoded images.To conduct an early examination of our prototype encoding, we opted to utilize existing pre-trained models instead of developing a neural network from scratch. This approach allowed us to save time and resources while still gaining an advantage. We conducted a comprehensive evaluation of our encoding technique by testing it on over several different variants of CNN and transformer models. The results of this evaluation are presented in Table. 5.1, which lists our top four performing models. GoogleNet outperformed the other three models and achieved superior results in binary classification, with an accuracy of 90%. A results comparison of our method with existing approaches are listed in Table. 5.3 and it is observed that KnightEncoding reaches performance similar to the state-of-the-art on binary classification. The results of the multi-class classification, however, were not optimal as shown in Table. 5.2, suggesting that our encoding algorithm and prediction model utilization need further refinement. Furthermore, we performed an uncertainty analysis on the most optimal model to evaluate its strengths and weaknesses. To comprehend the source of uncertainty, we segregated the data based on shared attributes, specifically the class and sequence lengths.

The class-based uncertainty analysis revealed that our model demonstrated greater certainty when applied to non-PVP data as opposed to PVP data. This disparity may be ascribed to the larger volume of training data available for the non-PVP category in contrast to the PVP category, or as an alternative, it might be related to their distinctive underlying sequence composition. With regard to length based uncertainty, predictions showed higher variance for longer sequences indicating that the model is less confident for longer sequence data compared to the shorter sequences. These gives us an overall insight that our method is most suitable for classifying shorter sequences (especially short non-PVPs) compared to the longer ones.

It was further observed that the GoogLeNet model, while the dropout is active during inference, experiences a considerable decline in its capacity to accurately categorize the pvp sequences, regardless of their lengths. In contrast, the non-PVP classification yields reliable and accurate predictions. In order to demonstrate the disparity in predictions between training mode (with active dropout) and evaluation mode (with inactive dropout), graph illustrations of the softmax predictions for both PVP and non-PVP scenarios were constructed. As depicted in Figure. 6.3, the graph on the left displays the model's predictions on PVP without dropout. All of the estimates are in close proximity to 1, resulting in accurate predictions. The graph on the right displays predictions generated using the dropout. The softmax predictions in this scenario exhibit a dispersed distribution, primarily concentrated on the left side of the graph (i.e. prediction probability <0.5). This suggests that the model lacks the capability to accurately recognize PVP sequences. Similarly, Figure. 6.4 illustrates the inference on non-PVP without dropout and with dropout. The predictions made via dropout exhibit accuracy, as indicated by a softmax value close to 1 and a lower variance compared to PVP, demonstrating the model's high level of confidence in accurately predicting non-PVP. This atypical observation could serve as an anchor for future studies on quantifying uncertainty in biological sequences. In this regard, Figures. 6.1, 6.2 further demonstrate the trend of softmax scores for dropout induced predictions. Figure. 6.2 shows the softmax prediction distributions for PVP gathering towards left (i.e incorrect predictions) and the scores for non-PVP accumulating on the right (i.e correct predictions), as shown in Fig. 6.1.

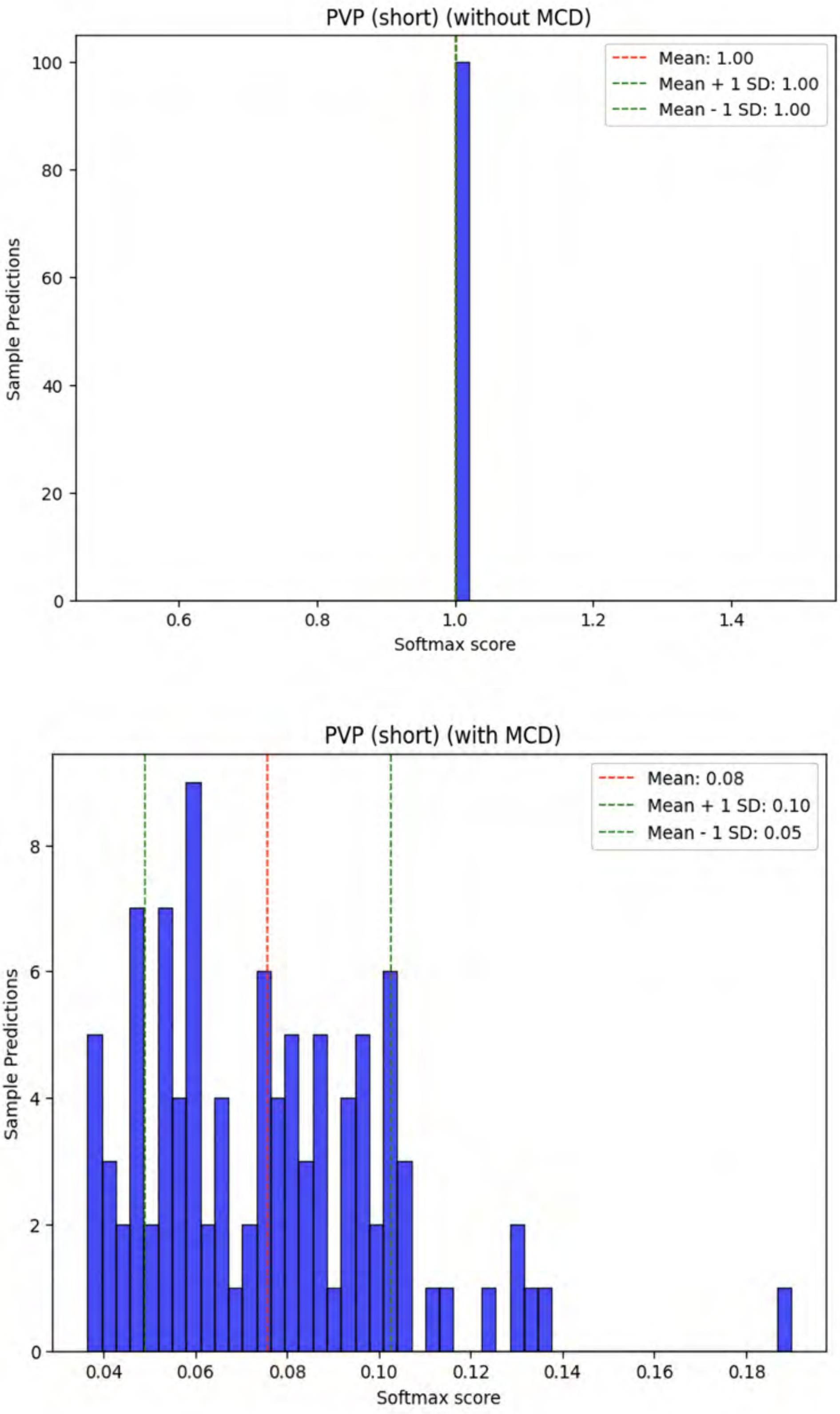


Figure 6.3: Softmax Distributions, with MCD (bottom) and without MCD (top) for short PVP sequences

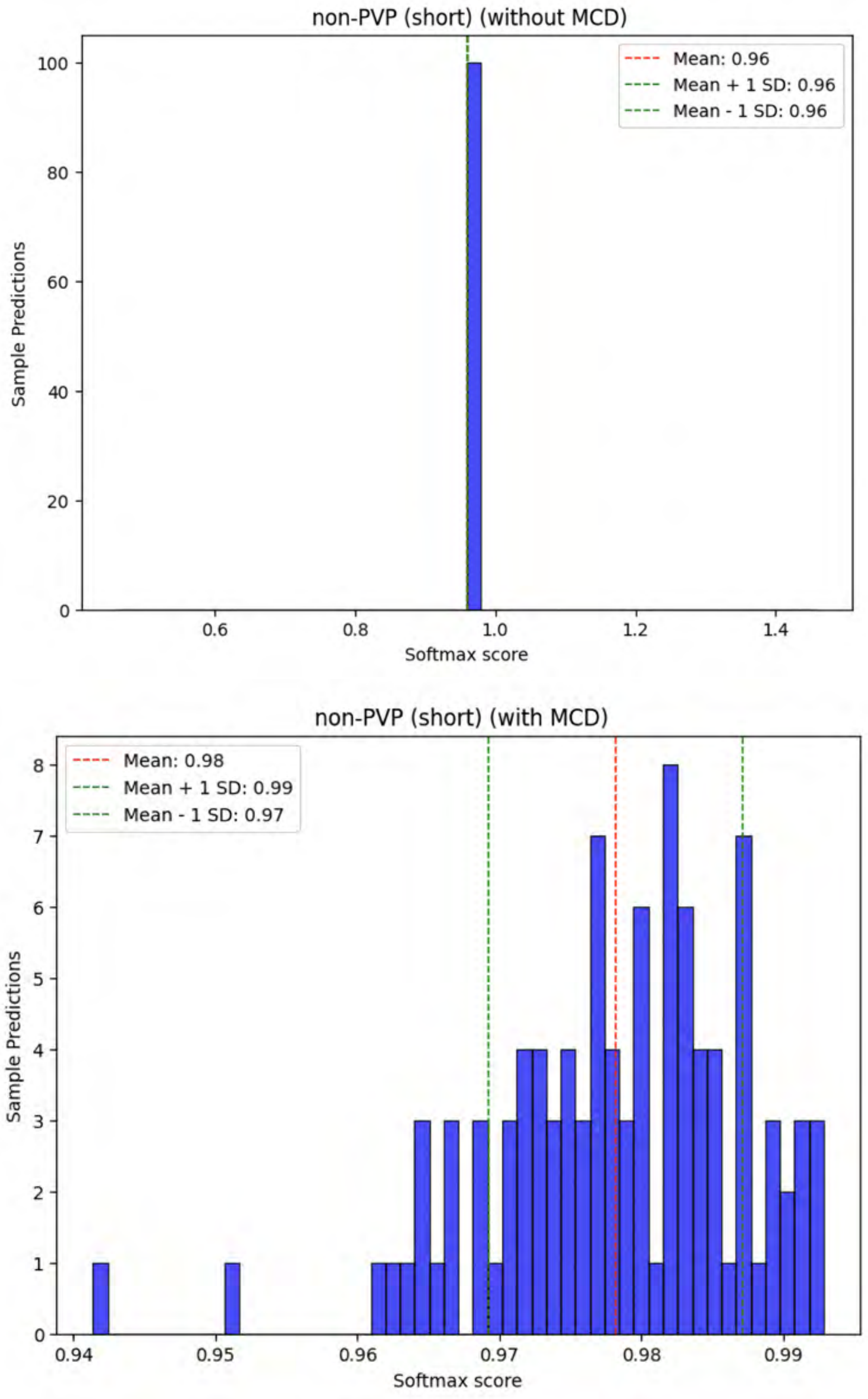
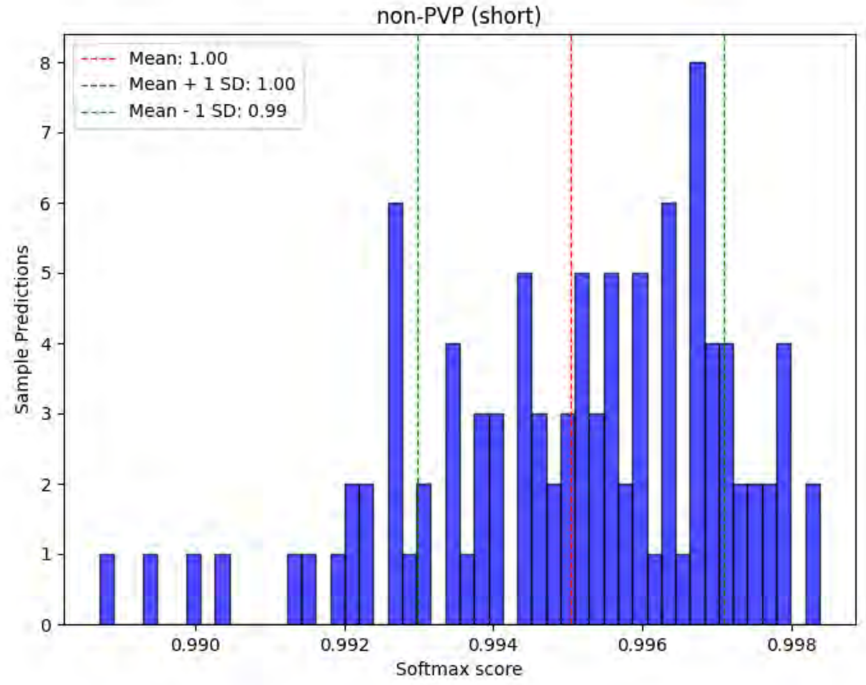


Figure 6.4: Softmax Distributions, with MCD (bottom) and without MCD (top) for short non-PVP sequences

6.1 Limitations

When considering the extent of this study, it is crucial to recognize the research constraints that could impact the classification prediction and uncertainty of certain data samples. Our proposed encoding method exhibited a limitation in which several instances of dot overlap were seen when encoding a sequence, particularly when the sequence is long. This phenomenon arises when many amino acids are encoded at the same position on the picture axis. This phenomenon ignores the sequential data associated with the dots positioned below, causing data loss. Although the limitation was effectively overcome for the less complex binary classification, but it posed a challenge for the more challenging multi-class classification. Furthermore, the optimization of hyperparameters related to encoding, such as point size, radius, and image resolution, still requires more refinement and improvement.



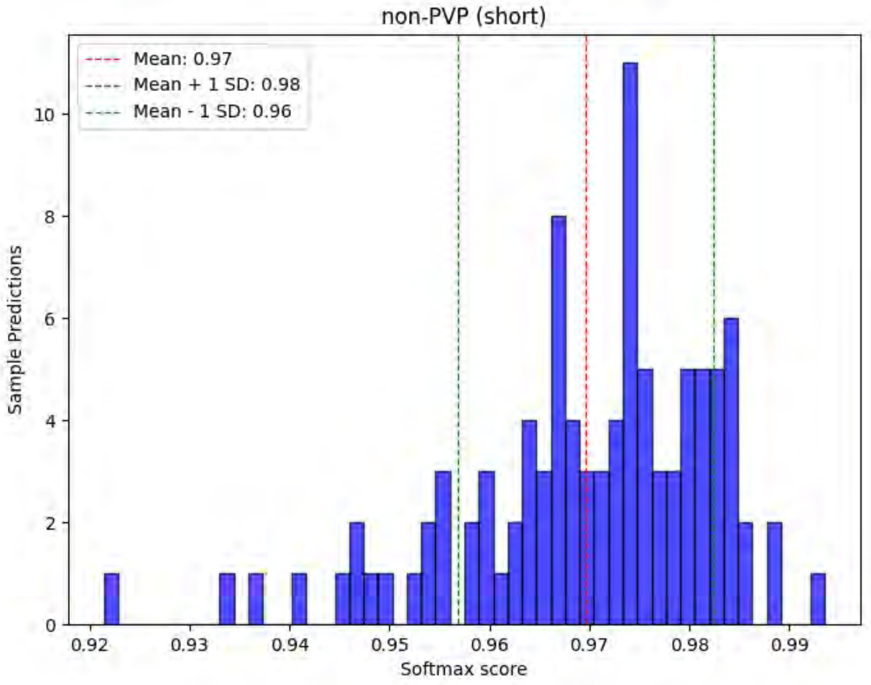
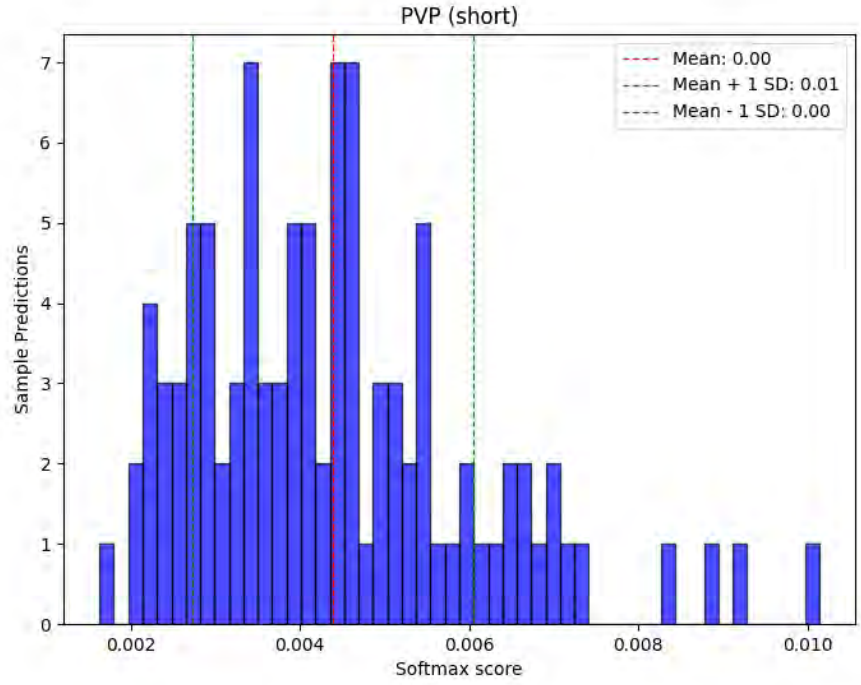


Figure 6.1: Softmax prediction distribution for sample non-PVP data



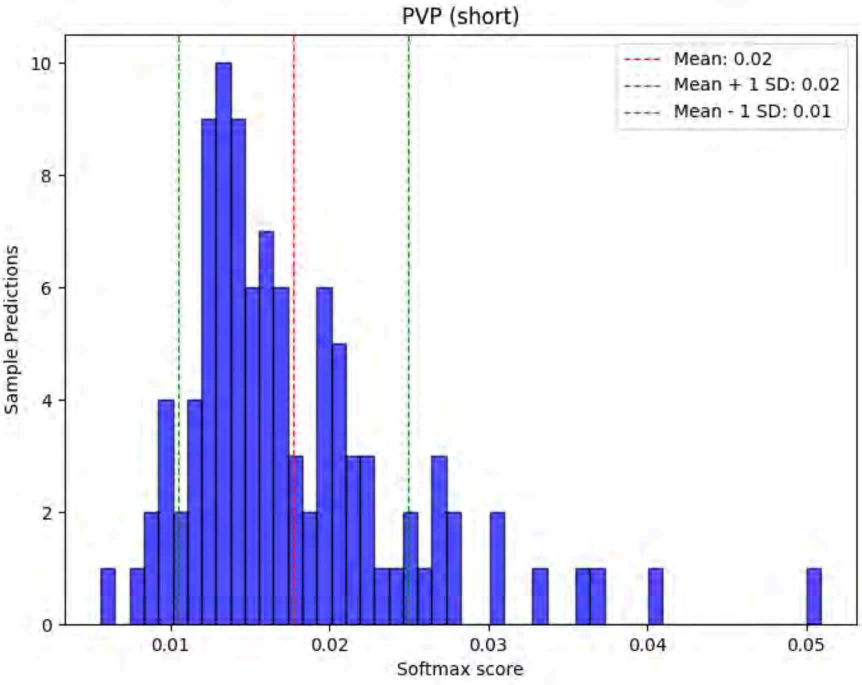


Figure 6.2: Softmax prediction distribution for sample PVP data

Chapter 7

Conclusion and Future Works

Phages demonstrate the ability to selectively target and infect bacteria, playing a pivotal role in ecology. The classification of phage proteins, often referred to as PVP, holds significant importance across various fields, especially in the context of pathogen-targeted therapy. Successful implementation of computational models for classification necessitates meticulous attention to feature extraction. This paper introduced and assessed a novel encoding approach for PVP, achieving satisfactory results in binary classification. Importantly, the proposed method overcomes limitations observed in existing image-based PVP encodings by preserving spatial information within sequences. Additionally, the study addresses uncertainties in biological data by employing Monte Carlo Dropout on the most efficient pre-trained model. This strategy allows for the identification of the model's susceptibility to different protein groups and varying sequence lengths, resulting in a precise and reliable methodology for PVP classification. While our study has made significant progress, further work is required to solve its existing constraints. To address the problem of encoding overlap, it is possible to utilize higher dimensions, such as 3D encodings or frame segregation, for each amino acid representation. This would allow the model to more effectively process the point representation of each sequence character, ensuring that no sequence information is compromised. Finally, our uncertainty analysis focused solely on data length and class, but it can be expanded to include additional characteristics, such as physiochemical compositions, in order to gain a more comprehensive understanding of the predictions. Ultimately, this study establishes a foundation for future research endeavors focused on examining the most effective methods of encoding proteins for accurate classification.

Bibliography

[1] H. Brüssow and F. Desiere, “Comparative phage genomics and the evolution of siphoviridae: Insights from dairy phages,” *Molecular microbiology*, vol. 39, no. 2, pp. 213–223, 2001.

[2] J. A. Berger, S. K. Mitra, M. Carli, and A. Neri, “Visualization and analysis of dna sequences using dna walks,” *Journal of the Franklin Institute*, vol. 341, no. 1-2, pp. 37–53, 2004.

[3] L.-F. Wang and M. Yu, “Epitope identification and discovery using phage display libraries: Applications in vaccine development and diagnostics,” *Current drug targets*, vol. 5, no. 1, pp. 1–15, 2004.

[4] R. F. Edwards RA, “Viral metagenomics,” *Nat Rev Microbiol*, 2005.

[5] S. Duffy, L. A. Shackelton, and E. C. Holmes, “Rates of evolutionary change in viruses: Patterns and determinants,” *Nature Reviews Genetics*, vol. 9, no. 4, pp. 267–276, 2008.

[6] R. Lavigne, P.-J. Ceyssens, and J. Robben, “Phage proteomics: Applications of mass spectrometry,” *Bacteriophages: Methods and Protocols, Volume 2 Molecular and Applied Aspects*, pp. 239–251, 2009.

[7] M. R. Henn, M. B. Sullivan, N. Stange-Thomann, *et al.*, “Analysis of high-throughput sequencing and annotation strategies for phage genomes,” *PLoS One*, vol. 5, no. 2, e9083, 2010.

[8] H.-L. Huang, P. Charoenkwan, T.-F. Kao, *et al.*, “Prediction and analysis of protein solubility using a novel scoring card method with dipeptide composition,” in *BMC bioinformatics*, Springer, vol. 13, 2012, pp. 1–14.

[9] V. Seguritan, N. Alves Jr, M. Arnoult, *et al.*, “Artificial neural networks trained to detect viral and phage structural proteins,” 2012.

[10] V. Seguritan, N. Alves Jr, M. Arnoult, *et al.*, “Artificial neural networks trained to detect viral and phage structural proteins,” 2012.

[11] P.-M. Feng, H. Ding, W. Chen, H. Lin, *et al.*, “Naive bayes classifier with feature selection to identify phage virion proteins,” *Computational and mathematical methods in medicine*, vol. 2013, 2013.

[12] H. Ding, P.-M. Feng, W. Chen, and H. Lin, “Identification of bacteriophage virion proteins by the anova feature selection and analysis,” *Molecular BioSystems*, vol. 10, no. 8, pp. 2229–2235, 2014.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] R. Menouni, G. Hutinet, M.-A. Petit, and M. Ansaldi, “Bacterial genome remodeling through bacteriophage recombination,” *FEMS microbiology letters*, vol. 362, no. 1, pp. 1–10, 2015.

[15] L. Zhang, C. Zhang, R. Gao, and R. Yang, “An ensemble method to distinguish bacteriophage virion from non-virion proteins based on protein sequence characteristics,” *International journal of molecular sciences*, vol. 16, no. 9, pp. 21 734–21 758, 2015.

[16] A. G. Cobián Güemes, M. Youle, V. A. Cantú, B. Felts, J. Nulton, and F. Rohwer, “Viruses as winners in the game of life,” *Annual review of virology*, vol. 3, pp. 197–214, 2016.

[17] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*, PMLR, 2016, pp. 1050–1059.

[18] C. Galiez, C. N. Magnan, F. Coste, and P. Baldi, “Viralpro: A tool to identify viral capsid and tail sequences,” *Bioinformatics*, vol. 32, no. 9, pp. 1405–1407, 2016.

[19] Y. Yuan and M. Gao, “Proteomic analysis of a novel bacillus jumbo phage revealing glycoside hydrolase as structural component,” *Frontiers in Microbiology*, vol. 7, p. 745, 2016.

[20] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” *Advances in neural information processing systems*, vol. 30, 2017.

[21] B. Manavalan, T. H. Shin, and G. Lee, “Pvp-svm: Sequence-based prediction of phage virion proteins using a support vector machine,” *Frontiers in microbiology*, vol. 9, p. 476, 2018.

[22] J.-X. Tan, F.-Y. Dao, H. Lv, P.-M. Feng, and H. Ding, “Identifying phage virion proteins by using two-step feature selection methods,” *Molecules*, vol. 23, no. 8, p. 2000, 2018.

[23] B. Cigan, *Chaos game representation of a genetic sequence*, Jan. 2019. [Online]. Available: https://towardsdatascience.com/chaos-game-representation-of-a-genetic-sequence-4681f1a67e14.

[24] S. McCallin, J. C. Sacher, J. Zheng, and B. K. Chan, “Current state of compassionate phage therapy,” *Viruses*, vol. 11, no. 4, p. 343, 2019.

[25] M. Arif, F. Ali, S. Ahmad, M. Kabir, Z. Ali, and M. Hayat, “Pred-bvp-unb: Fast prediction of bacteriophage virion proteins using un-biased multi-perspective properties with recursive feature elimination,” *Genomics*, vol. 112, no. 2, pp. 1565–1574, 2020.

[26] V. A. Cantu, P. Salamon, V. Seguritan, *et al.*, “Phanns, a fast and accurate tool and web server to classify phage structural proteins,” *PLOS Computational Biology*, vol. 16, no. 11, pp. 1–18, Nov. 2020. DOI: 10.1371/journal.pcbi.1007845. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1007845.

[27] P. Charoenkwan, S. Kanthawong, N. Schaduangrat, J. Yana, and W. Shoombuatong, “Pvpred-scm: Improved prediction and analysis of phage virion proteins using a scoring card method,” *Cells*, vol. 9, no. 2, p. 353, 2020.

[28] P. Charoenkwan, C. Nantasenamat, M. M. Hasan, and W. Shoombuatong, “Meta-ipvp: A sequence-based meta-predictor for improving the prediction of phage virion proteins using effective feature representation,” *Journal of Computer-Aided Molecular Design*, vol. 34, pp. 1105–1116, 2020.

[29] B. Ghoshal and A. Tucker, “Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection,” *arXiv preprint arXiv:2003.10769*, 2020.

[30] C. Meng, J. Zhang, X. Ye, F. Guo, and Q. Zou, “Review and comparative analysis of machine learning-based phage virion protein identification methods,” *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, vol. 1868, no. 6, p. 140 406, 2020.

[31] M. Abdar, F. Pourpanah, S. Hussain, *et al.*, “A review of uncertainty quantification in deep learning: Techniques, applications and challenges,” *Information fusion*, vol. 76, pp. 243–297, 2021.

[32] Z. Fang and H. Zhou, “Virionfinder: Identification of complete and partial prokaryote virus virion protein from virome data using the sequence and biochemical properties of amino acids,” *Frontiers in microbiology*, vol. 12, p. 615 711, 2021.

[33] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, “Escaping the big data paradigm with compact transformers,” *arXiv preprint arXiv:2104.05704*, 2021.

[34] E. Hüllermeier and W. Waegeman, “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods,” *Machine Learning*, vol. 110, pp. 457–506, 2021.

[35] D. Milanés-Hermosilla, R. Trujillo Codorniú, R. López-Baracaldo, *et al.*, “Monte carlo dropout for uncertainty estimation and motor imagery classification,” *Sensors*, vol. 21, no. 21, p. 7241, 2021.

[36] Y. Nami, N. Imeni, and B. Panahi, “Application of machine learning in bacteriophage research,” *BMC microbiology*, vol. 21, no. 1, pp. 1–8, 2021.

[37] D. Ofer, N. Brandes, and M. Linial, “The language of proteins: Nlp, machine learning & protein sequences,” *Computational and Structural Biotechnology Journal*, vol. 19, pp. 1750–1758, 2021.

[38] J. Starmer, *Entropy (for data science)*, Aug. 2021. [Online]. Available: https://youtu.be/YtebGVx-Fxw?si=bjQc1IKhNgXo45ND.

[39] S. Akbari Rokn Abadi, A. Mohammadi, and S. Koohi, “Walkim: Compact image-based encoding for high-performance classification of biological sequences using simple tuning-free cnns,” *Plos one*, vol. 17, no. 4, e0267106, 2022.

[40] Z. Fang, T. Feng, H. Zhou, and M. Chen, “Deepvp: Identification and classification of phage virion proteins using deep learning,” *GigaScience*, vol. 11, giac076, 2022.

[41] M. Kabir, C. Nantasenamat, S. Kanthawong, P. Charoenkwan, and W. Shoombuatong, “Large-scale comparative review and assessment of computational methods for phage virion proteins identification,” *EXCLI journal*, vol. 21, p. 11, 2022.

[42] M. Shujaat, J. S. Jin, H. Tayara, and K. T. Chong, “Iprom-phage: A two-layer model to identify phage promoters and their types using a convolutional neural network,” *Frontiers in Microbiology*, vol. 13, p. 1 061 122, 2022.

[43] W. Sun, Y. Pang, and G. Zhang, “Cct: Lightweight compact convolutional transformer for lung disease ct image classification,” *Frontiers in Physiology*, vol. 13, p. 1 066 999, 2022.

[44] R. K. Barman, A. K. Chakrabarti, and S. Dutta, “Prediction of phage virion proteins using machine learning methods,” *Molecules*, vol. 28, no. 5, p. 2238, 2023.

[45] N. Buhl, *Training vs. fine-tuning: What is the difference?* Nov. 2023. [Online]. Available: https://encord.com/blog/training-vs-fine-tuning/.

[46] J. S. Lee, J. Kim, and P. M. Kim, “Score-based generative modeling for de novo protein design,” *Nature Computational Science*, pp. 1–11, 2023.

[47] J. Shang, C. Peng, X. Tang, and Y. Sun, “Phavip: Phage virion protein classification based on chaos game representation and vision transformer,” *Bioinformatics*, vol. 39, no. Supplement$_1$, pp. i30–i39, Jun. 2023, ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btad229. eprint: https://academic.oup.com/bioinformatics/article-pdf/39/Supplement_1/i30/50741407/btad229.pdf. [Online]. Available: https://doi.org/10.1093/bioinformatics/btad229.

[48] “Uniprot: The universal protein knowledgebase in 2023,” *Nucleic Acids Research*, vol. 51, no. D1, pp. D523–D531, 2023.

[49] Y. Zhang and Z. Li, “Rf_phage virion: Classification of phage virion proteins with a random forest model,” *Frontiers in Genetics*, vol. 13, p. 1 103 783, 2023.

Thesis Github Repository

All the source code along with the notebook files and datasets that were used to produce this thesis can be found at https://github.com/eniac00/ProteoKnight.