Dynamically Weighted Pairwise Cross-Attention Driven Feature Fusion in Hybrid Convolutional Neural Networks for Classification of COVID-19 Variants

Vatsal Shah[1], Love Fadia[1], Mohammad Hassanzadeh[1] ,Jonathan Wu[1], Majid Ahmadi[1], & George Pappas [2]

[1] Department of Electrical and Computer Engineering, University of Windsor, Windsor, Canada

[2] Department of Electrical and Computer Engineering, Lawrence Technological University, Michigan, United States of America

Correspondence: Mohammad Hassanzadeh, University of Windsor, Ontario, N9B 3P4, Canada. E-mail: mhassan@uwindsor.ca

Received: January 6, 2025 Accepted: February 17, 2025 Online Published: April 30, 2025

doi:10.5539/cis.v18n1p111 URL: https://doi.org/10.5539/cis.v18n1p111

Abstract

The extensive global impact of coronavirus is evident, causing widespread disruption to public health and economies around the world. This disease is caused by the severe acute respiratory syndrome virus. Accurate detection helps control the virus spread, reduces death rates, and lessens the overall impact on communities. Several significant research gaps exist in handling unbalanced datasets and achieving high accuracy with properly balanced data. These issues pose substantial challenges to the development of robust and reliable classification models. Unbalanced datasets, where certain classes are overrepresented, can bias models towards dominant classes, leading to suboptimal performance for underrepresented strains. To address these gaps, this paper introduces a novel and effective method to classify the three dominant variants of severe acute respiratory syndrome. Alpha, Delta, and Omicron. Here, we utilize a balanced dataset of 9000 images and we propose an innovative series of Dynamically Weighted Pairwise Cross-Attention feature fusion models designed to effectively integrate complementary genomic features, delivering robust and accurate performance across diverse genomic classification tasks. To achieve this, we first utilize Genomic Image Processing techniques, such as Frequency Chaos Game Representation and Markov Transition Field, to transform genomic sequences into informative visual representations, enabling more effective feature extraction and fusion. Then the resultant images are used to train our series of models. Our enhanced models outperform state-of-the-art results by achieving a remarkable accuracy of 99.62%.

Keywords: artificial intelligence, convolutional neural network, deep learning, genomic image processing

List of Abbreviations

COVID-19 Coronavirus Disease 2019

SARS-CoV-2 Severe Acute Respiratory Syndrome Coronavirus 2

RNA Ribonucleic Acid

DNA Deoxyribonucleic Acid

cDNA Complementary DNA

RT-PCR Reverse Transcription Polymerase Chain Reaction

WGS Whole Genome Sequencing

CNN Convolutional Neural Network

LSTM Long Short-Term Memory

DWPCA Dynamic Feature Fusion Using Weighted Pairwise Cross-Attention

ResNet Residual Network

XceptionNet Extreme Inception Network

FCGR Frequency Chaos Game Representation

KNN K-Nearest Neighbors

MTCN Multistage Temporal Convolutional Network

GAF Gramian Angular Field

MTF Markov Transition Field

SVM Support Vector Machine

MLP Multilayer Perceptron

SCV SARS-CoV-2 Variant

ML Machine Learning

DL Deep Learning

BL Broad Learning

CT Computed Tomography

ECA Efficient Channel Attention

UNET U-Net Convolutional Network

NCBI National Center for Biotechnology Information

RGB Red Green Blue

ROC Receiver Operating Characteristic

XceptionNet Extreme Inception Network

DWPCA-RDNet Dynamic Weighted Pairwise Cross-Attention ResNet-DenseNet

DWPCA-DXNet Dynamic Weighted Pairwise Cross-Attention DenseNet-XceptionNet

DWPCA-XRNet Dynamic Weighted Pairwise Cross-Attention XceptionNet-ResNet

ResNet Residual Network

DenseNet Densely Connected Convolutional Network

XceptionNet Extreme Inception Network

1. Introduction

Coronavirus (COVID-19) is a deadly disease that causes illnesses ranging from mild fever to severe pneumonia and was first identified in China (**Muralidar et al. 2020**). It is an infectious disease caused by the SARS-COV-2 virus. Emerging in 2019, SARS-COV-2 is classified as an RNA virus, meaning its genetic material is composed of RNA, not DNA. When the virus infects human cells (its host), it uses the host cell machinery to replicate its RNA and produce new viral particles. Importantly, the virus RNA remains in the cell cytoplasm and does not integrate into the host cell DNA (**Emrani et al.**; **2021**). Although SARS-COV-2 does not directly alter human DNA, its presence triggers an immune response in the host, leading to the activation of various immune-related genes in the human genome (**M. Li et al.**; **2022**). These interactions highlight the importance of studying the human genetic makeup, as individual genetic variations in human DNA can influence the immune response and significantly affect the severity and outcomes of the infection (**Chowdhury et al.**; **2020**). Additionally, as the virus evolves, its RNA undergoes mutations that result in the emergence of new variants. These mutations occur within the virus RNA genome, not in human DNA (**Harvey et al.**; **2021**). This underscores the importance of genomic studies of both the virus and the human host to better understand and mitigate the effects of SARS-COV-2 and its variants. These variants have specific mutations in their genetic structure that affect how the virus interacts with the human immune system and spreads among the population (**Andre et al.**; **2023**). Therefore, it becomes essential to find a way which helps us to differentiate these variants so that patients can be treated accordingly. Several approaches

allow researchers to classify the variants of SARS-COV-2. One of them is by using more specialized Reverse Transcription Polymerase Chain Reaction (RT-PCR) genotyping assays or whole genome sequencing (WGS). RT-PCR genotyping assays use specific primers and probes designed to target known mutations in the virus genome associated with different variants. For instance, mutations such as N501Y, E484K, and P681R can be identified using these targeted assays to infer the presence of particular variants like Alpha, Beta, Gamma, or Delta (**Centers for Disease Control and Prevention**; **2024**). RT-PCR test can detect particular changes that set one virus variation apart from another by examining the genetic material of the virus in a sample (**Bray et al.**; **2024**). Recent advancements in computational genomics and deep learning have significantly enhanced the classification of SARS-COV-2 variants. Techniques such as transformer-based models, hybrid frameworks combining CNNs and LSTMs, and ensemble learning approaches have improved the analysis of complex genomic sequences, the identification of subtle mutations, and generalization across diverse datasets (**Dip**; **Sarkar**; **Setu**; **et al.**; **2023**). Besides, attention mechanisms have been crucial in underlining important genomic regions for better classification. These models are usually combined with domain knowledge from methods such as WGS and advanced genotyping assays to improve the accuracy of prediction (**Qin et al.**; **2024**). While the field has recently advanced with hybrid frameworks and ensemble methods, it still needs approaches that will integrate and prioritize features in a dynamic way, thus capturing both interdependencies and subtle variations within genomic data. Most of the currently available methods focus on independent feature extraction, which can miss important interactions between global and local patterns. By integrating features in a dynamic manner, the gaps could be addressed to reach a complete understanding of the genomic structure and its variations, particularly in the analysis of complex genomic data. Moreover, the challenges are further compounded by the presence of imbalanced datasets or, in some cases, small sample sizes, as observed in the classification of COVID-19 variants. These issues highlight the necessity for robust methodologies that can effectively handle limited and skewed data distributions while maintaining high accuracy and interpretability in genomic analyses. This work, therefore, proposes a novel approach to deal with such needs by incorporating DWPCA block within the well-known CNN architectures, namely ResNet-50, DenseNet, and XceptionNet. This block helps in dynamically interacting and coordinating features between models for capturing complex genomic patterns. During training, two models ran in parallel for feature extraction, considering the complementary strengths of the selected architectures out of the three models, which are ResNet-50, capturing global patterns through residual connections; DenseNet, providing local feature representations via densely connected layers; and XceptionNet, capturing spatial information using depthwise separable convolutions. The DWPCA mechanism integrates the extracted features dynamically for effective feature interaction and gives priority to the most relevant genomic pattern. This improves the detection of subtle genomic alterations, allowing for precise differentiation between the balanced dataset of Alpha, Delta, and Omicron variants. After feature fusion, the unified feature representation is fed into a fully connected layer for classification. This method significantly enhances the accuracy and robustness of the classification by tackling data complexity and variability. The contribution of the article is:

- Introduced a novel Dynamic Feature Fusion approach using a Weighted Pairwise Cross-Attention (DWPCA) block integrated within CNN architectures for enhanced genomic feature interaction.
- Leveraged the complementary strengths of ResNet-50, DenseNet, and XceptionNet to capture global, local, and spatial genomic patterns, enabling precise detection of subtle genomic alterations.
- Demonstrated improved classification accuracy and robustness in differentiating COVID-19 variants (Alpha, Delta, and Omicron), effectively addressing challenges of data complexity and imbalanced datasets.

1.1 Organization of Article

The article is structured as follows: Section 2 presents a review of related work. Section 3 details the methodology, providing an in-depth discussion of each component. Section 4 showcases the results, while Section 5 outlines future work and limitations.

2. Related Work

This section describes the literature related to our work, which uses CNN. Deep learning has created several opportunities for researchers to make progress in the exploration of various characteristics of the COVID-19 virus. One such approach is to use images of X-rays or CT scans and then apply CNN models for performing the classification task. This method is well-researched and widely adopted in the medical imaging field due to its proven effectiveness. However, our approach, which uses DNA sequences, diverges from this traditional method and is somewhat less explored in current research. Recent research has explored various ways to classify the variants of COVID-19. For instance, (**Hammad**; **Ghoneim**; **et al.**; **2023**) implemented FCGR which counts the number of occurrences of nucleotides of DNA and after plotting genomic sequences of Human coronavirus (H-COV) and SARS-COV-2, it converts into an image and then they

use the AlexNet model for extracting features, followed by KNN for classification. (**M. S. Hammad et al.**; **2023**) used an enhanced Deep Learning CNN for the classification of SARS-COV2 variants by first converting the DNA sequence into a vector of binary numbers and then applying a combination of CNN and LSTM. (**Mwanga et al.**; **2023**) first converted DNA sequences into numbers and then binarized them. They then applied MTCN for the classification of COVID-19 variants. (**Ullah et al.**; **2022**) developed a technique that converts DNA sequences into images using Recurrent Plot, GAF, and MTF, which captures the probability of transition between DNA nucleotides, followed by ResNet-34 for classification. (**Ali et al.**; **2022**) employed a deep learning model for feature selection from DNA sequences for COVID-19 variant classification and subsequently applied machine learning classifiers like SVM, KNN, MLP, and Random Forest for classification. (**Togrul and Arslan**; **2022**) proposed an alignment-free k-mer based LSTM deep learning model to classify different COVID-19 variants using genetic sequence data. (**Basu and Campbell**; **2022**) utilized an SCV filter, a deep hybrid model, employing embedding, attention residual networks, and LSTM components to accurately identify SARS-CoV-2 variant sequences. Moreover, (**Wang and Gao**; **2023**) and (**Muhammed S. Hammad et al.**; **2023**) utilized features from images generated from DNA sequences rather than using direct sequence data. These extracted features were then classified using KNN, as seen in (**Muhammed Sayed Hammad**; **Ghoneim**; **and Mabrouk**; **2021**). In another approach, (**Harikrishnan**; **Pranay**; **and Nagaraj**; **2022**) used CNN models for classification between COVID-19 and other viruses. (**Lopez-Rincon et al.**; **2021**) combined deep features with MLP for the classification of COVID-19 and influenza viruses. In the domain of medical imaging, (**Gomes et al.**; **2021**) employed the concept of a broad learning system, using an autoencoder for classification rather than conventional deep learning models, testing on datasets such as Chest-Xray 8, COVID-CT, and SARS-COV-2. Similarly, (**G. Wu and Duan**; **2022**) performed classification using Mask R-CNN to distinguish between COVID-19 and non-COVID-19 cases based on CT images of the chest, followed by few-shot learning using ResNet-50 for severity classification. Furthermore, (**D. and R.**; **2023**) developed a CNN+Variational Autoencoder-based system for detecting COVID-19 using smartwatch data on steps and heart rate. (**Abir et al.**; **2023**) proposed a multi-ECA attention mechanism with a federated learning system for enhancing X-ray image classification while preserving quality. (**Adjei-Mensah et al.**; **2024**) introduced a novel approach to COVID-19 detection based on cough sounds, utilizing two neural networks one for detecting coughs and another for analyzing their characteristicsto classify COVID-19 symptoms. In addition, there are certain other works, which are related to other diseases. (**Liyanarachchi et al.**; **2023**) had developed a novel algorithm for the treatment of Breast Cancer. They developed an ensemble learning approach by combining the UNeT model with SVM and Random Forest. Using an attention mechanism to highlight key locations with an attention map, the researcher presents a deep neural network framework in which every layer is a neural network in and of itself.

Moreover, another approach utilized Capsule Networks to enhance local feature recognition and resolve CNNs translation invariance problem. With classifiers such as SVM, Random Forest, and Logistic Regression, the model outperforms state-of-the-art techniques in Human Immunodeficiency Virus (HIV) integration site identification; a comparison with the two most popular approaches in the literature is also provided (**Munshi et al.**; **2024**). In addition, the authors of (**Boruah and Das**; **2024**) presented an automated approach to measure the number of inflammatory cells in hepatitis C-infected liver biopsies. Adaptive thresholding and clump decomposition were used to identify important regions in color-corrected images, which were then used to extract characteristics from those areas. Afterward, the sequential floating search technique and principal component analysis were used to reduce dimensionality. Using Gaussian parametric and mixed models optimized to maximize the area under the ROC curve, the system ability to categorize regions as inflammatory or healthy was assessed and ultimately tested using previously ranked photographs by observers with varying degrees of skill (**Hodgson**; **Harrison**; **and Cross**; **2006**).Table 1 demonstrates the comparison with other state-of-the-art techniques in terms of methodology, which is more correlated with the proposed one, and dataset size. The article (**Ali et al.**; **2022**) classifies the variants with a lower amount of data, whereas in (**M. S. Hammad et al.**; **2023**), (**Harikrishnan**; **Pranay**; **and Nagaraj**; **2022**), (**Lopez- Rincon et al.**; **2021**), and (**Gomes et al.**; **2021**), class imbalance is observed, which influenced their results. In the rest of the articles, classification is binary. Our approach outperforms other state-of-the-art methods primarily in its ability to accurately classify closely related viral sequences, a task where DNA sequences are very hard to classify since they are the mutations of the same virus. Moreover, we ensured that an equal amount of data is utilized to overcome the class imbalance issue, and a larger dataset has been used for classification to demonstrate the significance of the results in multi-class classification, ensuring that our approach does not have the limitations faced by other works described above.

Table 1. Comparison with State of the Art Methods

Sr No	Papers	Methodology	Dataset	Accuracy
1	Hamaad.M.et.al	FCGR + Pretrained AlexNet + KNN	HCOV: 2863 MERS-COV: 734 SARS-COV-2: 3700 SARS-COV-1: 64	99.71%
2	Hamaad.M.S.et.al	Standard features from gray-level images + KNN	SARS-CoV-1: 57 MERS-CoV: 258 COVID-19: 300	100%
3	Hamaad.M.S.et.al	Standard features from fourth-order FCGR images + KNN	COVID-19: 3700 Coronavirus: 3663	99.39%
4	Harikrishnan.N.et.al	Chaos features + SVM	COVID-19: 4498 SARS-COV-1: 101	99%
5	S.Ali.et.al	Recurrent plot + Deep Learning models, GAF + Deep Learning models, MTF + Deep Learning models	SARS-CoV-2 22 variants (7000 sequences)	82%
6	Lopez-Rincon.A.et.al	CNN	COVID-19: 66 Others: 487	98.75%
7	Gomes.J.C.et.al	Deep features + MLP	COVID-19: 171 Influenza: 347162	98%
8	**Proposed Method**	FCGR + DWPCA MTF + DWPCA	Alpha: 3000 Delta: 3000 Omicron: 3000	99.62%

3. Proposed Methodology

This section presents the proposed methodology and provides a detailed explanation of each component. The workflow of the method is illustrated in Figure 1. In the initial step, DNA sequences, each approximately 30,000 base pairs in length, were collected from the National Center for Biotechnology Information (Virus n.d.), a specialized database within the NCBI main portal that contains extensive information on viral genomics. This repository serves as a central source for various details about viruses, including taxonomy, genomic sequences, and related literature. The initial plan aimed to gather over 9,000 DNA sequences for each of the three variants; however, due to the limited availability of the Omicron variant (only 3,000 sequences), an equal number of sequences (3,000) was selected for each variant from the USA, Mexico, and Canada. This ensured a geographically diverse and balanced dataset, resulting in a final dataset size of 9,000 samples. To address redundancy in sequence data, a preprocessing step was applied, where headers were removed from the FASTA files, and all sequences were scanned for corruption or incompleteness. Incomplete sequences were identified and removed, ensuring that only unique and complete sequences were retained. This step was essential for preventing model bias and improving the generalizability of the classifiers. Next, sequences were converted into images using FCGR and MTF. These techniques were selected for their ability to represent genomic sequences in a grid or matrix format, making them highly suitable for CNN-based models. Alternative methods, such as direct nucleotide color mapping, DNA walk, and spectrogram plots, were initially considered but were not further analyzed in this study due to their lower

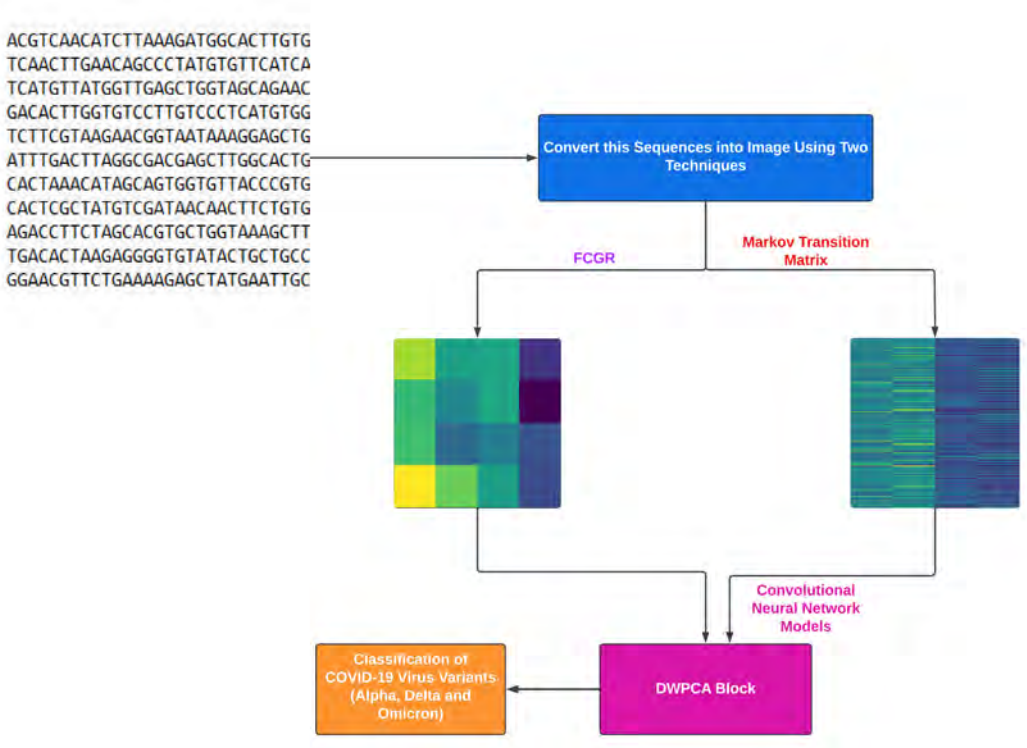


Figure 1. Proposed methodology

feature distinction and detail in preliminary experiments. Once the 9,000 images were generated, the dataset was split into training, validation, and testing sets. Care was taken to ensure that no class imbalance was introduced during the split. The training dataset consisted of 6,300 images, while the validation and testing datasets contained 1,350 images each. This balanced distribution ensured that model performance was not influenced by any dataset biases.The proposed architecture enhances classification performance by leveraging the complementary strengths of multiple models. During training, features were extracted in parallel using pairs of CNNs selected from ResNet-50, DenseNet, and XceptionNet. ResNet-50 captured global patterns through its residual connections, DenseNet provided detailed local features via its densely connected layers, and XceptionNet incorporated depthwise separable convolutions to capture additional spatial information. This parallel processing strategy maximized feature diversity and computational efficiency. The extracted features were then processed through a DWPCA layer, facilitating interaction between feature representations. The DW-PCA block dynamically assigned weights to features, prioritizing the most relevant patterns during training. Following this, a dynamic feature fusion step combined the interacted features into a unified representation. The final classification task was performed using a fully connected layer followed by a softmax activation function. Additionally, early stopping was implemented to halt training if validation loss did not improve for five consecutive epochs. After training, model performance was evaluated using various metrics. Table 2 provides details of the dataset used in this study.

Table 2. Properties of the Dataset

Properties of Dataset	Respective Values
Image Size	224x224 pixel
Nature of Image	3 Channel RGB Image
Number of Classes	3
Samples of Alpha Variant of SARS-COV-2	3000
Samples of Delta Variant of SARS-COV-2	3000
Samples of Omicron Variant of SARS-COV-2	3000

3.1 Genomic Image Processing

In this subsection, we introduce two distinct Genomic Image Processing techniques that convert raw DNA sequence data into images. The first technique, FCGR, counts the occurrences of nucleotides within a DNA sequence. After constructing the FCGR matrix, we visualize and save it as an image for various k-mers, as explained further. The second technique, MTF, captures the transition dynamics between different nucleotide states in the DNA sequence. We generate

MTF matrices of orders 2 and 3, plot them as images, and use them to enhance interpretability. By transforming genetic sequences into visual patterns, this approach facilitates a more intuitive assessment of genomic relationships and variant distinctions. Ultimately, it improves the transparency of deep learning model predictions.

3.1.1 Frequency Chaos Game Representation

FCGR method provides a two-dimensional grid, where each nucleotide in the DNA sequence: adenine (A), thymine (T), cytosine (C), and guanine (G) is recorded with coordinates. In general, A is assigned (0, 0), T is assigned (0, 1), C is assigned (1, 0), and G is assigned (1, 1) (**Anitas**; **2022**).

To construct the Chaos Game Representation, we start from the center of the grid, positioned at (0.5, 0.5). For each nucleotide in the sequence, we move halfway towards the coordinates assigned to that nucleotide. If the current position is (x_i, y_i) and the nucleotide's coordinates are (x_n, y_n), the new position (x_{i+1}, y_{i+1}) is calculated as (**LÂąÂğochel and Heider**; **2021**):

$$(x_{i+1}, y_{i+1}) = \left(\frac{x_i + x_n}{2}, \frac{y_i + y_n}{2}\right) \tag{1}$$

The iterative process of moving from the current position to the new position by averaging the coordinates with those of the nucleotide is described in Equation (1). Every nucleotide in the sequence goes through this procedure once more, producing a set of dots that represent the structure of the DNA sequence. An investigation of frequency is the last step in which DNA sequences are examined in more depth due to k-mers, which are contiguous sequences of k nucleotides. K-mers are all potential substrings of length k for a DNA sequence S of length n. The biological importance of these patterns stems from the depiction of k-mer diversity across distinct viral strains. Smaller k-mers (e.g., k=2, k=3) reveal conserved viral functions, whereas bigger k-mers (e.g., k=5, k=6) highlight areas of mutation and divergence, particularly in Omicron. These alterations could correspond to major functional changes in the virus, impacting replication, immunological evasion, or transmissibility (**Messaoudi**; **Oueslati**; **and Lachiri**; **2014**). A mathematical definition for these k-mers is provided in Equation (2):

$$m_i = S[i : i + k - 1] \tag{2}$$

By combining the idea of k-mers with FCGR, DNA sequence visualization becomes more detailed and granular. Then the frequency analysis is carried out by splitting the grid into smaller parts and measuring the number of times each region is visited once the positions for each k-mer have been determined. A color gradient is then used to visualize this frequency data, with higher frequencies corresponding to darker or more intense colors (**Rizzo et al.**; **2016**). After being enhanced with k-mer information, the resulting FCGR image shows the distribution and frequency of particular k-mer, exposing patterns and motifs that aid in the comprehension of the genetic structure and function. Here K-means that we have utilized are 2,3,4,5,6. When calculating the FCGR for DNA sequences, the output image is generated by plotting the FCGR matrix. For K=2, all possible 2-mers of the DNA sequence are considered, resulting in a 4x4 image. Increasing K to 3 involves all possible 3-mers, producing an 8x8 image with more detail compared to the K=2 image. Similarly, at K=4, all possible 4-mers are considered, creating a 16x16 image with even more features than the 2-mer and 3-mer images. As the value of K increases, the images generated from the FCGR matrix plots contain more details, which leads to higher accuracy in analysis.

At K=5, all possible 5-mers have been considered and the size of the image now is 32x32. As compared to the previous we get a more accurate classification rate. At last, K=6 all possible 6-mers have been counted and from that 6-mers frequency of characters has been calculated and here the order of the matrix that is used to generate the image is 64x64. Figure 2 displays the FCGR images for the Alpha Delta and Omicron variants from left to right with each image corresponding to a different k-mer value of 2 to 6 from top to bottom.

3.1.2 Markov Transition Field

DNA sequences can be represented as pictures using MTF, which captures the patterns of transition between various states in a sequence. These states are associated with the nucleotides (A, C, G, and T) in DNA sequences. First, the transition probabilities between these states must be computed, and a matrix representing these probabilities must then be created. The sequence patterns can be visually analyzed by viewing this matrix as an image (**Zakarczemny and Zajecka**; **2022**). The DNA sequence must first be mathematically encoded to produce an MTF. Assigning a distinct integer (such as A=0, C=1, G=2, or T=3) to every nucleotide is a popular encoding technique. We compute the transition probabilities between every pair of states after encoding. The transition matrix P contains these probabilities which are given by (**Ryabko and Usotskaya**; **2008**):

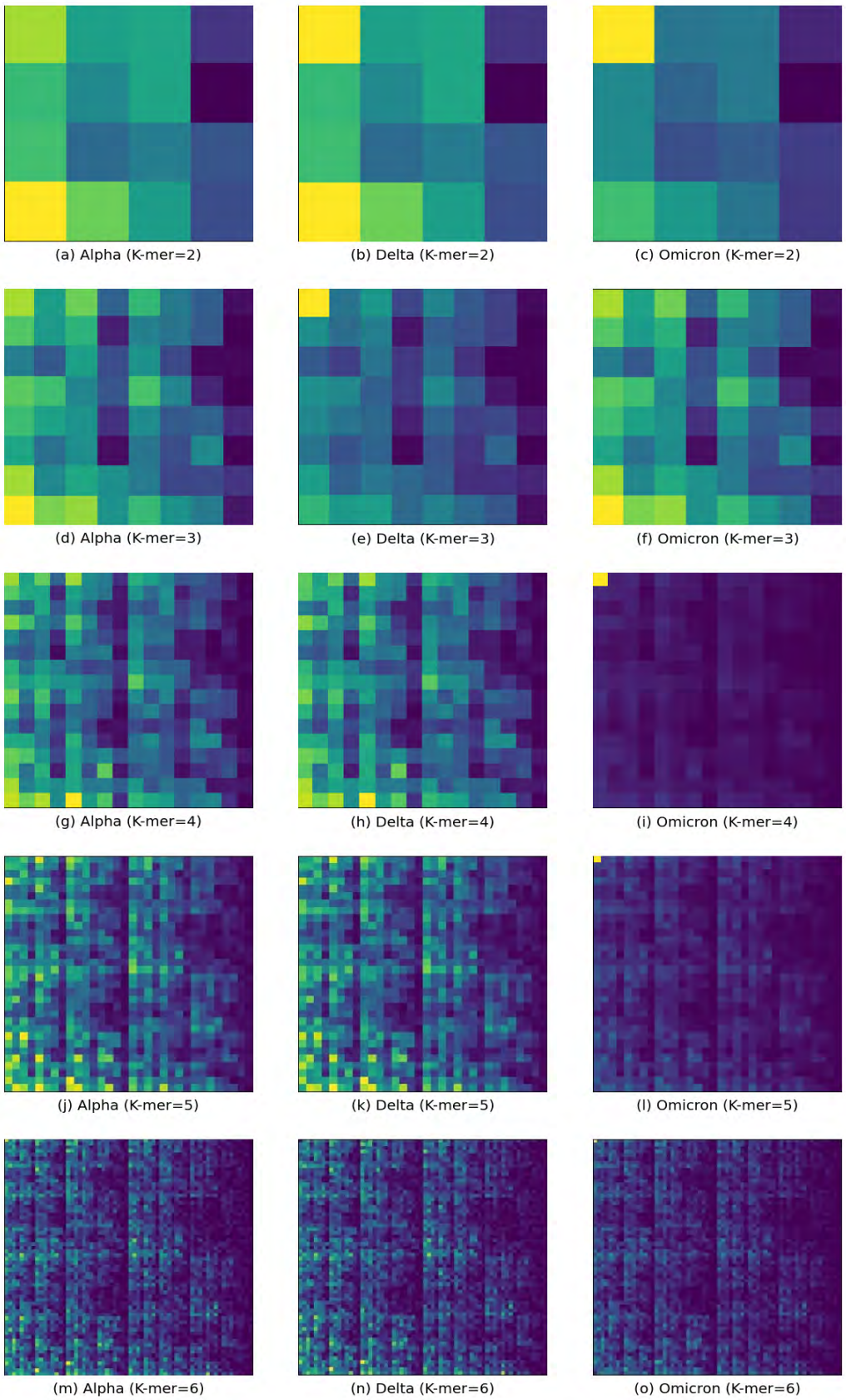


Figure 2. DNA sequences of Alpha, Delta, and Omicron variants converted to images using FCGR with k-mers from 2 to 6

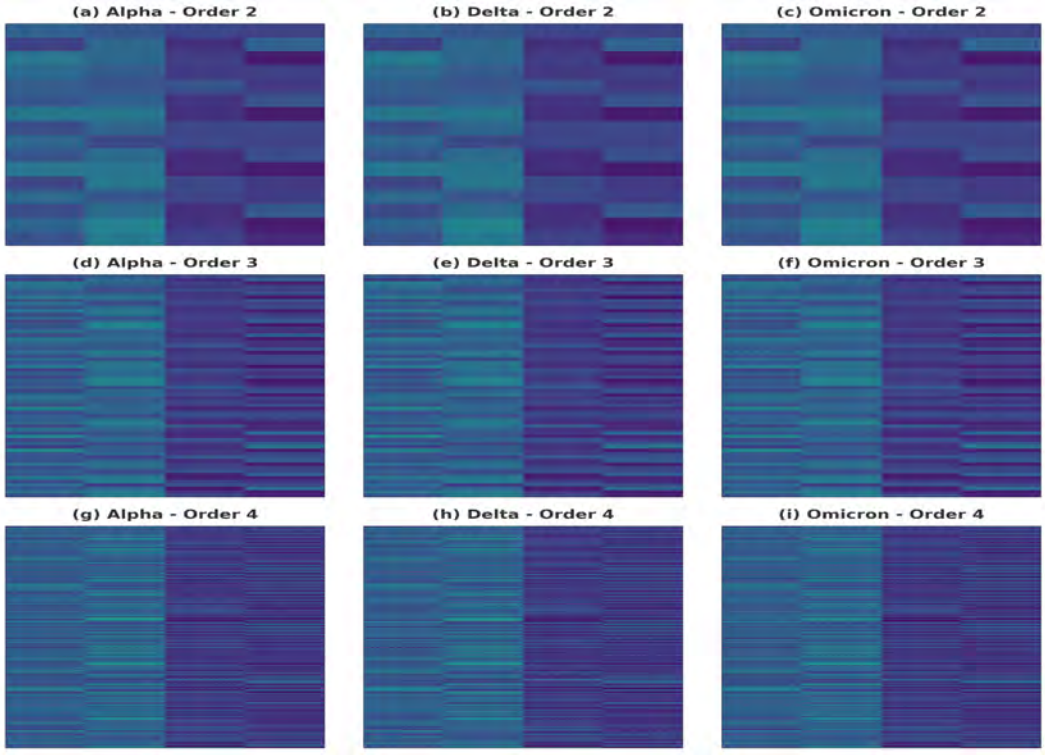


Figure 3. DNA sequences of Alpha, Delta, and Omicron variants converted to images using Markov with orders from 2 to 4

$$P_{ij} = \frac{n_{ij}}{\sum_{k=1}^{m} n_{ik}} \tag{3}$$

where n_{ij} is the number of transitions from state i to state j, the denominator is the total number of transitions from state i, m is the Markov order matrix and k is the total number of states.These probabilities are mapped onto a grid that corresponds to the places in the original DNA sequence to form the MTF, which is done after the transition matrix is built. The likelihood that two nucleotides will transit between them at various points during the sequence is represented by each element in the field. The matrix order in an MTF is determined by the number of distinct states in the sequence. Here we have taken order number 2 and 3. For Markov order 2 both the current and prior states affect the likelihood of a state transition in an order Markov process. This is computing the likelihood that each nucleotide will appear after each potential pair of nucleotides that came before it in DNA sequences Later on, we develop a third-order Markov transition matrix to estimate the transition probabilities, where each triplet of nucleotides in the sequence must be counted to accomplish this. The transition probability P_{ijk} specifically defined as the probability of reaching state k, assuming that the prior states were i and j. The biological relevance of these patterns reflects the evolutionary and mutational distinctions between the Alpha, Delta, and Omicron types. At lower orders (Order = 2), the conserved patterns indicate that these variants share genetic characteristics critical to their survival (**Yoon**; **2009**). However, when the order grows (Order = 3, 4), the patterns become more different, showing areas of genetic diversity that may be linked to the variations differences in immune evasion, transmission rates, or toxicity. These discrepancies could be due to changes in viral proteins that influence how each variation functions biologically.

By utilizing the current state and the two stages prior, the third-order Markov transition matrix is determined. To estimate the probabilities of transition, we build a third-order Markov transition matrix by counting the occurrences of each triplet of nucleotides in the sequence. Stated differently, the transition probability P_{ijkl} represents the likelihood of switching from states i, j, and k to state l. Figure 3 illustrates MTF Matrix from order 2 to 4.

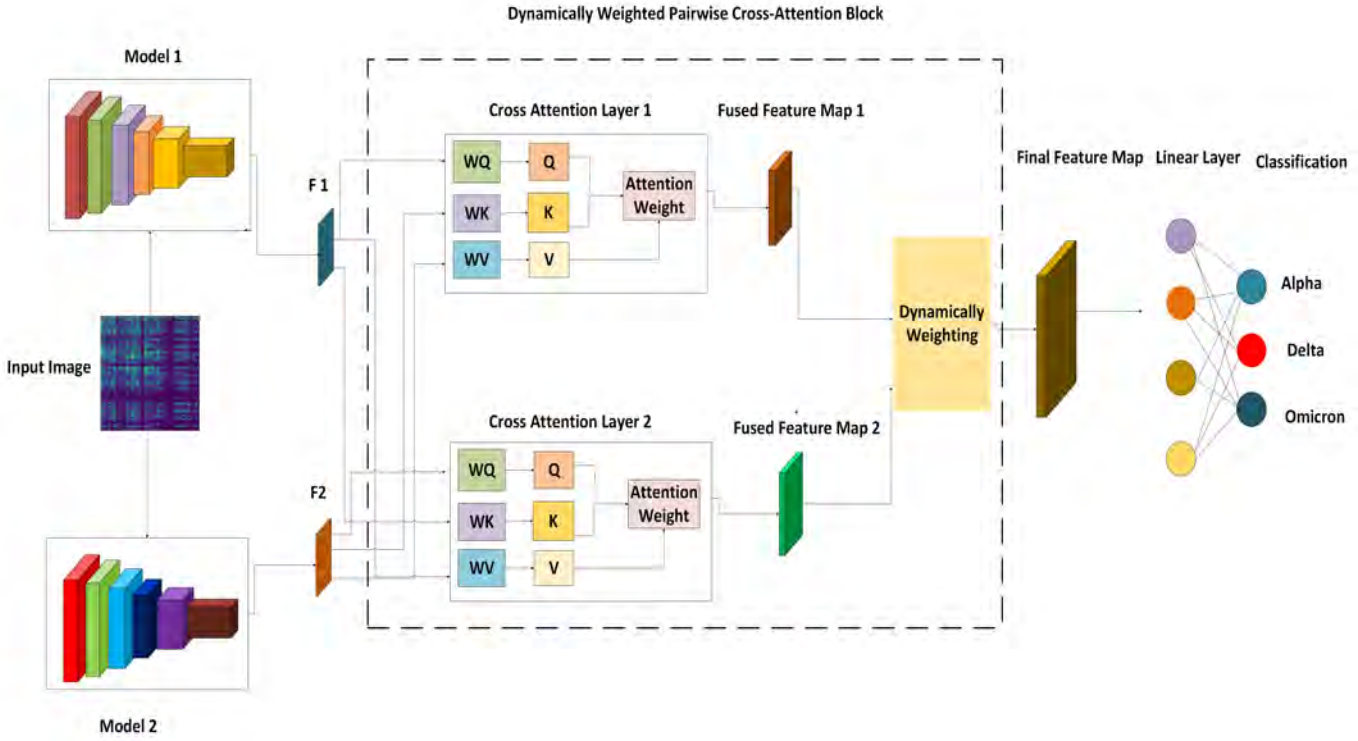


Figure 4. Proposed Architecture

3.2 Proposed Deep Learning Architecture

The proposed architecture, shown in Figure 4, is designed for robust classification tasks, specifically aimed at distinguishing between classes such as Alpha, Delta, and Omicron. The architecture leverages the complementary strengths of multiple models by integrating two parallel feature extraction pipelines, cross-attention layers for feature interaction, and dynamic weighting for effective feature fusion. This design ensures improved accuracy, robustness, and adaptability for diverse datasets. The architecture follows a modular approach, allowing flexibility in selecting feature extraction models and attention mechanisms. This modularity also facilitates the exploration of different combinations of models to find optimal performance configurations for specific data sets. During the training phase, a single input image is processed to extract structural or spatial patterns relevant to the classification task. While in the feature extraction stage, two parallel models, for instance, ResNet-50 and DenseNet, extract hierarchical feature representations. ResNet captures global patterns with residual learning, while DenseNet provides local, densely connected feature maps. These feature maps are then passed through our proposed DWPCA block. This block allows the interaction between the features extracted by these two models through attention. It calculates attention-based feature maps and fuses those with learnable weights. Learnable weights allow dynamic adjustments in the contributions of the fused feature maps during training, giving precedence to the most relevant information. Finally, the feature maps obtained after fusion feeds through global pooling and a fully connected layer to get the class probabilities. We designed the architecture to address the challenges posed by highly variable data, where individual models may miss subtle patterns. By integrating multiple models and utilizing attention mechanisms, the architecture ensures comprehensive feature extraction and robust classification. In the further subsections, we will discuss each component of the architecture in detail.

3.2.1 Parallel Feature Extraction

Feature extraction models are chosen for better performance of the architecture. Various combinations of state-of-the- art architectures for Model 1 and Model 2 were used to capture diverse feature representations as in (**H. Zhu et al.**; **2023**). DWPCA-RDNet is based on ResNet and DenseNet for Model 1 and Model 2, respectively. Residual learning and dense connectivity have been well-balanced in capturing global and local features effectively. DWPCA-DXNet, uses DenseNet for Model 1 and Xception for Model 2, using dense feature propagation and depthwise separable convolutions for efficient and detailed feature extraction. DWPCA-XRNet, applies Xception for Model 1 and ResNet for Model 2, leveraging the strength of residual learning and efficient spatial feature extraction to ensure robust feature representation. These combinations provide diverse and complementary features, significantly enhancing the per- formance of the models. In

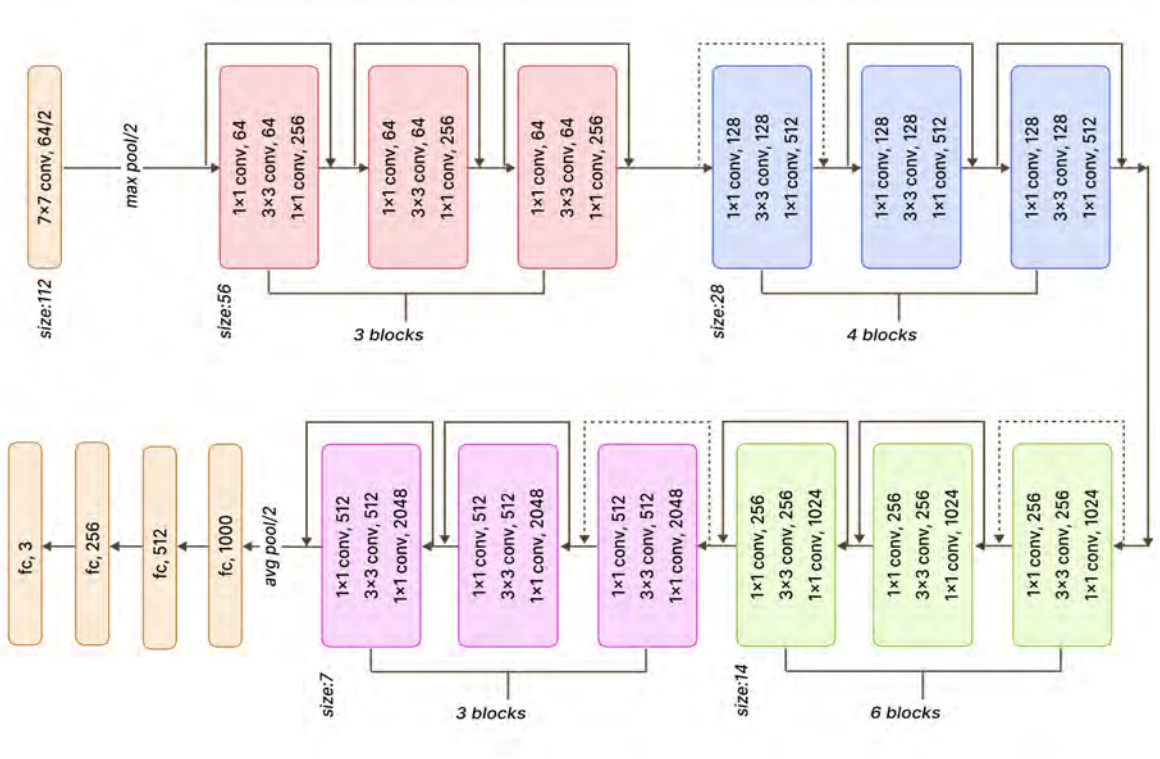


Figure 5. Architecture of ResNet-50

the next section, we will delve into the specific CNN models employed and their contributions to the overall architecture.

3.2.2 ResNet-50

ResNet is a deep Convolutional Neural Network. There are many deeper versions of ResNet available. For feature extraction in our study. The model has 50 layers and the advantage of using this architecture is that they have residual blocks in them. These blocks address the problem of vanishing gradients faced by deep CNNs with the help of skip connections, which subsequently allow the training of very deep networks. Figure 5 demonstrates the architecture of the ResNet-50 model. This can be mathematically formulated as (**Wei et al.**; **2020**):

Given an input $\mathbf{x}$, a residual block computes the output $\mathbf{y}$ as follows (**Lin et al.**; **2022**):

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x} \tag{4}$$

where $\mathbf{x}$ is the input. $\mathcal{F}(\mathbf{x}, \{W_i\})$ is the residual function representing the series of transformations (weight layers, batch normalization, and ReLU activations); $\mathbf{y}$ is the output. The residual function $\mathcal{F}(\mathbf{x}, \{W_i\})$ typically consists of two or three layers, depending on the specific design of the block. For a two-layer residual block (**S. He et al.**; **2023**), it is designated by:

$$\mathcal{F}(\mathbf{x}) = W_2\sigma(\mathrm{BN}(W_1\mathbf{x})) \tag{5}$$

where W_1 and W_2 are the weights of the convolutional layers, BN denotes batch normalization, a n d σ is the ReLU activation function. After the feature extraction part, the output feature maps, which have been through several convolutional and residual layers, are three-dimensional (i.e., height, breadth, and depth). These features are then flattened into a one-dimensional vector before being fed into a dense layer. The dense layer finally transforms these features linearly. This output is usually routed via a softmax activation to generate a probability for classification.

3.2.3 DenseNet

DenseNet is also a type of deep CNN. It is one of the most widely used models for image classification. Similarly to ResNet, there are many versions of DenseNet available with varying depths and layers. In our study, we have used DenseNet, which is a 121-layered CNN. The main reason behind the wide usage of DenseNet architecture is that it includes dense connections. These connections ensure maximum information flow between layers and address the vanishing

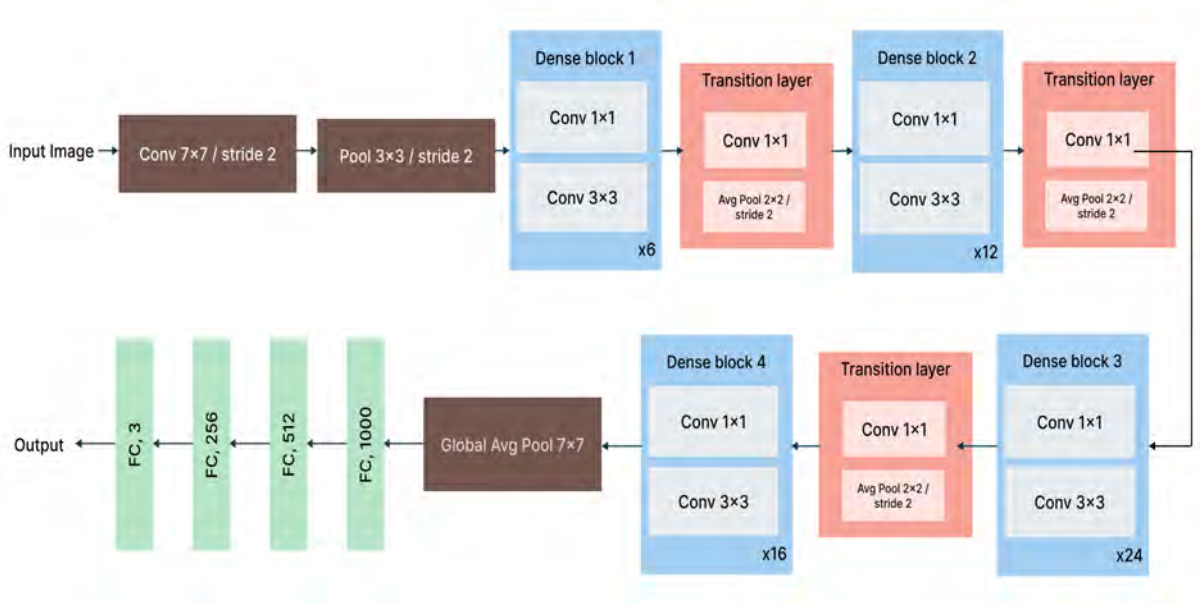


Figure 6. Architecture of DenseNet-121

gradient problem, allowing the training of very large and deep networks (**Shen et al.**; **2024**). Figure 6 demonstrates the architecture of the DenseNet model.

Given an $\mathbf{x}$, a dense block computes the output $\mathbf{y}$ by concatenating the input feature maps with the output feature maps of each preceding layer within the same block (**Usama et al.**; **2019**):

$$\mathbf{y}_l = H_l([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{l-1}]) \tag{6}$$

where $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{l-1}$ are the feature maps from the preceding layers; $H_l(\cdot)$ is the composite function of batch normalization (BN), ReLU, and convolution (Conv). $\mathbf{y}_l$ is the output of the l-th layer. The composite function $H_l(\cdot)$ can be expressed as (**Habib and Qureshi**; **2022**):

$$H_l(\mathbf{x}) = \text{Conv}(\sigma(\text{BN}(\mathbf{x}))), \tag{7}$$

where Conv represents the convolutional layer. BN denotes batch normalization, and σ is the ReLU activation function.

Between two contiguous dense blocks, a transition layer is introduced to perform down-sampling. This is achieved by a convolutional layer followed by an average pooling layer. The transition layer can be formulated as (**Habib and Qureshi**; **2022**):

$$\mathbf{y}_{\text{trans}} = \text{AvgPool}(\text{Conv}(\sigma(\text{BN}(\mathbf{x})))) \tag{8}$$

where AvgPool represents the average pooling layer, Conv represents the convolutional layer, BN denotes batch normalization, σ is the ReLU activation function and $\mathbf{y}_{\text{trans}}$ is the output of the transition layer. After passing through multiple dense blocks and transition layers, the output feature maps, which are typically three-dimensional (height, width, and depth) are subjected to the flattening process. This process converts the 3D feature maps into a 1D feature vector. This flattened vector is then passed into the dense layer for classification.

3.2.4 XceptionNet

Figure 7 demonstrates the architecture of the XceptionNet model. In XceptionNet, a standard convolution is factorized into a depthwise convolution and a pointwise convolution via the depthwise separable convolution. This not only boosts the performance of the model but also reduces the computational costs and model size. The depthwise convolution applies a single filter per input channel, and the pointwise convolution combines these outputs to create new features. By adding depthwise separable convolutions, XceptionNet is a deep convolutional neural network that expands on the concepts of Inception. This architecture produces a model that performs better and is more efficient. We have employed the XceptionNet model, a CNN with 48 layers. The benefit of XceptionNet design is that it can train very deep networks by more efficiently capturing spatial correlations across channels by utilizing depthwise separable convolutions (**Farag et al.**; **2021**).

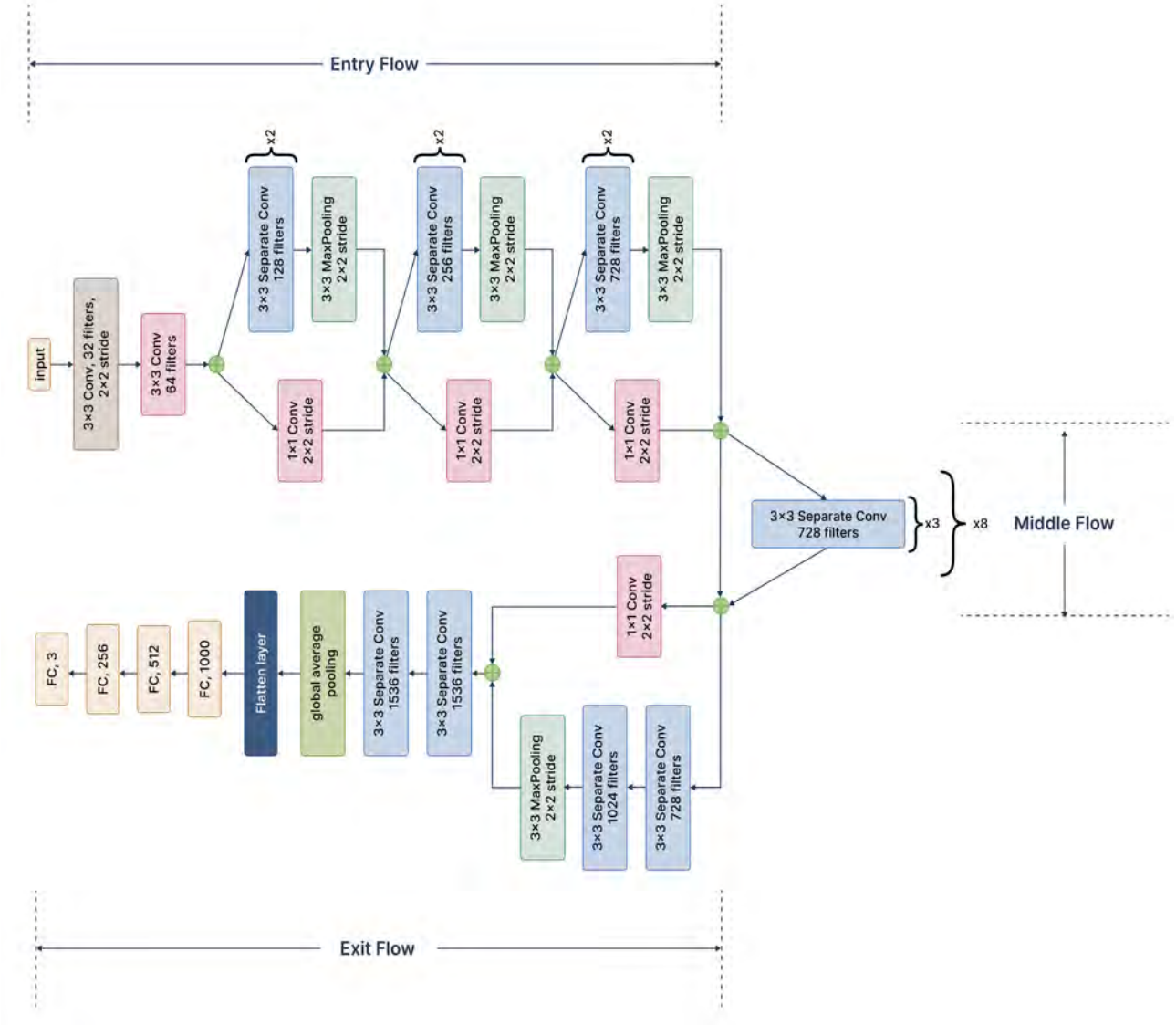


Figure 7. Architecture of XceptionNet

The Entry Flow, Middle Flow, and Exit Flow are the three primary flows that comprise the XceptionNet architecture (**Deng et al.**; **2009**). Depthwise separable convolutions are a key component of these flows, which are engineered to process input images through a sequence of convolutional and pooling layers in an efficient way. The Entry Flow main job is to quickly downsample the input image while retaining all of its important properties. While the network depth and ability to learn more complicated features are increased, the feature maps spatial dimensions remain unchanged as a result of this part. These layers are applied repeatedly, which aids in capturing complex patterns and hierarchies in the data. This flow raises the depth while drastically decreasing the spatial dimensions. Regardless of the input dimensions, a global average pooling layer is used after the convolutions to provide a fixed-size vector. The output is then obtained by flattening and passing this vector through fully connected layers. The output is then fed through a softmax activation to produce probabilities for each class for the classification tasks (**K. He et al.**; **2016**).

3.2.5 Dynamically Weighted Pairwise Cross Attention Block

The DWPCA block is responsible for fusing features extracted by two parallel deep learning models. It computes the interaction between features using a cross-attention mechanism and combines the resulting feature maps (**H. Wu et al. n.d.**). Let the feature maps extracted by Model1 and Model2 be $F_R \in \mathbb{R}^{B\times C\times H\times W}$ and $F_D \in \mathbb{R}^{B\times C\times H\times W}$, respectively, where (**Xu**; **Fu**; **and D. Zhu**; **2023**): B: Batch size; C: Number of channels; H, W: Height and width of the feature maps. For both feature maps, the DWPCA block computes the Query (Q), Key (K), and Value (V) matrices (**Xu**; **Fu**; **and D. Zhu**; **2023**):

$$Q_R = F_R W_Q^R, \quad K_R = F_R W_K^R, \quad V_R = F_R W_V^R \tag{9}$$

$$Q_D = F_D W_Q^D, \quad K_D = F_D W_K^D, \quad V_D = F_D W_V^D \tag{10}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{C\times C_r}$ are learnable weight matrices, and C_r is the reduced dimensionality of the attention space. The cross-attention mechanism computes interaction scores between the feature maps. For example Model1 and Model2 feature maps, the attention weights are designated by(**Amin et al.**; **2022**):

$$A_{RD} = \text{Softmax}\left(\frac{Q_R K_D^\top}{\sqrt{C_r}}\right), \quad A_{DR} = \text{Softmax}\left(\frac{Q_D K_R^\top}{\sqrt{C_r}}\right) \tag{11}$$

The attended feature maps are (**X. Li et al.**; **2020**) denoted by:

$$F_{RD} = A_{RD} V_D, \quad F_{DR} = A_{DR} V_R \tag{12}$$

Then the attended feature maps are fed to the dynamic weighting mechanism, which combines the outputs of the cross-attention layers (F_{RD} and F_{DR}) to produce a unified representation. Unlike softmax-based normalization, the weights α and β are directly learned as independent parameters during training. The final fused feature map F_{fused} is computed as (**Cinar**; **Ozcan**; **and Kaya**; **2022**):

$$F_{\text{fused}} = \alpha \cdot F_{RD} + \beta \cdot F_{DR} \tag{13}$$

where α and β are learnable parameters initialized to 0.5 and updated during training allowing the architecture to independently adjust the contributions of F_{RD} and F_{DR}.

The final fused feature map is passed through a global average pooling layer, followed by a fully connected layer for classification (**Zhou et al.**; **2022**):

$$F_{\text{pool}} = \text{GlobalAvgPool}(F_{\text{fused}}) \tag{14}$$

$$P = \text{Softmax}(W F_{\text{pool}} + b), \tag{15}$$

where W and b are the weights and bias of the fully connected layer, and $P \in \mathbb{R}^{B\times C}$ represents the predicted probabilities for C classes.

The DWPCA block provides a robust solution for feature fusion by dynamically integrating feature maps from two parallel deep learning models using a cross-attention mechanism. This enables effective modeling of interactions between global and local patterns, resulting in richer representations. Its learnable weighting mechanism (α, β) adapts the contributions of each feature map during training, ensuring flexibility and robustness to data variability. The reduced dimensionality of the attention space enhances computational efficiency, while the fused feature map significantly improves classification performance. Overall, the DWPCA block is a scalable, efficient, and adaptable solution for multi-source feature integration.

4. Results

This section provides a thorough analysis of our experiment findings, showing how well our models per- formed according to a number of measures, such as Receiver Operating Characteristic(ROC) curves, F1 scores, confusion matrices, precision, recall, and accuracy. The ROC curve offers a graphical representation of the trade-off between true positive rate (TPR) and false positive rate (FPR) across different classification thresholds, highlighting the model ability to distinguish between SARS-CoV-2 variants.

4.1 *Comparitive Analysis for Different k-mers of FCGR*

We determined each model precision, recall, and F1 score in order to measure performance. The definition of these measures is designated by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{19}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{20}$$

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \tag{21}$$

The performance metrics for the three models DWPCA-RDNet, DWPCA-DXNet, and DWPCA-XRNet applied using the FCGR approach for feature fusion and assessed across various k-mer sizes are summarized in Table 3. With accuracies ranging from 39.33% to 44.23% and comparable precision, recall, and F1-scores indicating similar constraints, all three models perform pretty poorly with a k-mer size of 2. This implies that smaller k-mers dont offer sufficient granularity to identify significant patterns for efficient categorization. The models show a noticeable improvement in performance as the k-mer size grows. Raising the k-mer size to 3 improves DWPCA-XRNet accuracy to 95.55%. Moving from k-mer size 3 to size 4 results in even higher gains, with accuracies over 98% for all models. DWPCA-XRNet outperforms other models with increasing k-mer sizes, achieving precision, recall, and F1-scores of 99.62% and a maximum accuracy of 99.62% by k-mer size 5. DWPCA-RDNet and DWPCA-DXNet work well at greater k-mer sizes, achieving accuracies up to 99%. Interestingly, k-mer sizes of four or more are related to near-optimal performance, implying that increased feature resolution improves the models capacity to discriminate between classes. Among the three models, DWPCA-XRNet demonstrates the most consistent and superior performance, followed closely by DWPCA-RDNet and DWPCA-DXNet. Overall, the table underscores the importance of selecting an appropriate k-mer size, with larger sizes yielding significantly better results.

Table 3. Summary of Performance Metrics for FCGR Technique for Feature Fusion

Sr. No	Model	k-mer	Accuracy	Precision	Recall	F1-Score
1	DWPCA-RDNet	2	42.30%	43.58%	42.30%	42.86%
2	DWPCA-DXNet	2	39.33%	29.33%	39.33%	33.50%
3	DWPCA-XRNet	2	44.23%	29.38%	44.66%	35.24%
4	DWPCA-RDNet	3	92.59%	92.61%	92.59%	92.60%
5	DWPCA-DXNet	3	95.40%	95.41%	95.40%	95.42%
6	DWPCA-XRNet	3	95.55%	95.60%	95.55%	95.58%
7	DWPCA-RDNet	4	98.66%	98.66%	98.66%	98.66%
8	DWPCA-DXNet	4	98.00%	98.00%	98.00%	98.00%
9	DWPCA-XRNet	4	98.74%	98.73%	98.74%	98.74%
10	DWPCA-RDNet	5	99.48%	99.49%	99.48%	99.48%
11	DWPCA-DXNet	5	99.32%	99.32%	99.33%	99.32%
12	DWPCA-XRNet	5	99.62%	99.62%	99.62%	99.62%
13	DWPCA-RDNet	6	99.33%	99.34%	99.33%	99.33%
14	DWPCA-DXNet	6	98.59%	98.61%	98.59%	98.60%
15	DWPCA-XRNet	6	99.10%	99.12%	99.10%	99.10%

Figures 8 to 15 provide additional insight into the performance of DWPCA-RDNet summarized in Table 3, demonstrating model behavior across different k-mer sizes for classification tasks using the FCGR approach. The loss and accuracy curves, as shown in Figures 8 and 9, reflect the performance of the model for training and validation, respectively, over epochs. For k = 2, the training and validation accuracies are low and inconsistent, with obvious variations, while the loss does not converge efficiently. In the case of k=3, the accuracy stabilizes while the loss drops in a much more consistent fashion. From k=4 onwards, the accuracy quickly converges to 0.99 within a few epochs, and the loss curves converge to their minimum, which shows that increasing k-mer size increases the model capacity to learn and generalize well. Furthermore, the ROC curves, as shown in Figures 10 11, and 12 illustrate the true positive rate versus the false positive rate for each class (Alpha, Delta and Omicron) with k-mer sizes ranging from 2 to 6. At k=2, the AUC values are rather low (Class Alpha: 0.63, Class Delta: 0.63, Class Omicron: 0.73), indicating poor performance as shown in the table. As k grows, the AUC improves dramatically. At k=3, AUC values approach near-optimal levels (about 0.99 for Class Alpha and Delta, 0.98 for Omicron), and by k=4, all classes achieve perfect or near-perfect AUC values of 1.00. This aligns with the sharp improvement in accuracy, precision, recall, and F1-score observed in the table for higher k-mer sizes. In addition, the confusion matrices in Figures 13,14,and 15 represent the class-wise distribution of the predictions. At k=2, there is a large-scale mis classification among the three classes; for example, Alpha is often misclassified as Delta or Omicron. At k = 3, misclassifications reduce enormously and diagonal dominance shines through clearly. From k=4 to k=6, almost perfect classification is attained by the near-zero off-diagonal values. This corroborates with the table that higher k-mer size results in near-perfect accuracy metrics.

In Figures 16 and 17, the training and validation metrics for DWPCA-DXNet show its learning progression over epochs for different k-mer sizes. At k=2, both training and validation accuracies are low and unstable, with big fluctuations; the loss does not converge well, indicating poor learning. At k=3, accuracy increases smoothly, and the loss drops more linearly, indicating better generalization. At k=4 and beyond, accuracy goes to near-perfect levels almost (1.0) very early in training, and the loss stabilizes at minimal values. The convergence of the training and validation loss curves indicates effective learning with minimal overfitting. Moreover, the ROC curves in Figures 18,19, and 20 provides insight into the classification performance of DWPCA-DXNet for k-mer sizes ranging from 2 to 6. For k=2, AUC values are relatively low, indicating poor discriminative ability for all classes (e.g., Alpha: 0.57, Delta: 0.59, Omicron: 0.65). The curves stay far from the top-left corner to reflect a high rate of false positives. At k=3, AUC improves massively with values very close to 1.0 for most classes: Alpha and Delta reach 1.0, and Omicron gets to 0.99. This suggests a vast improvement in the separation between classes. Starting from k=4, all models will have a perfect or near-perfect AUC score of 1.0 for all classes, reflecting good classification performance. The ROC curves in all cases follow the diagonal-to-top-left trajectory, indicating the best predictions with almost zero false positives. Furthermore, Confusion matrices in Figures 21,22 and 23 demonstrates the distribution of true and predicted labels for different k-mer sizes for DWPCA-DXNet. At k=2, there is much misclassification across all classes, with high off-diagonal values; for example, Alpha and Delta are mostly misclassified as Omicron. At k=3, diagonal dominance is much stronger, with fewer misclassifications; most predictions correspond correctly to true labels. From k=4 to k=6, almost there is perfect classification in the sense that the diagonal dominance is very strong, with negligible values for the off-diagonals. This agrees with the table and other metrics, showing high accuracy and precision for higher k-mer sizes.

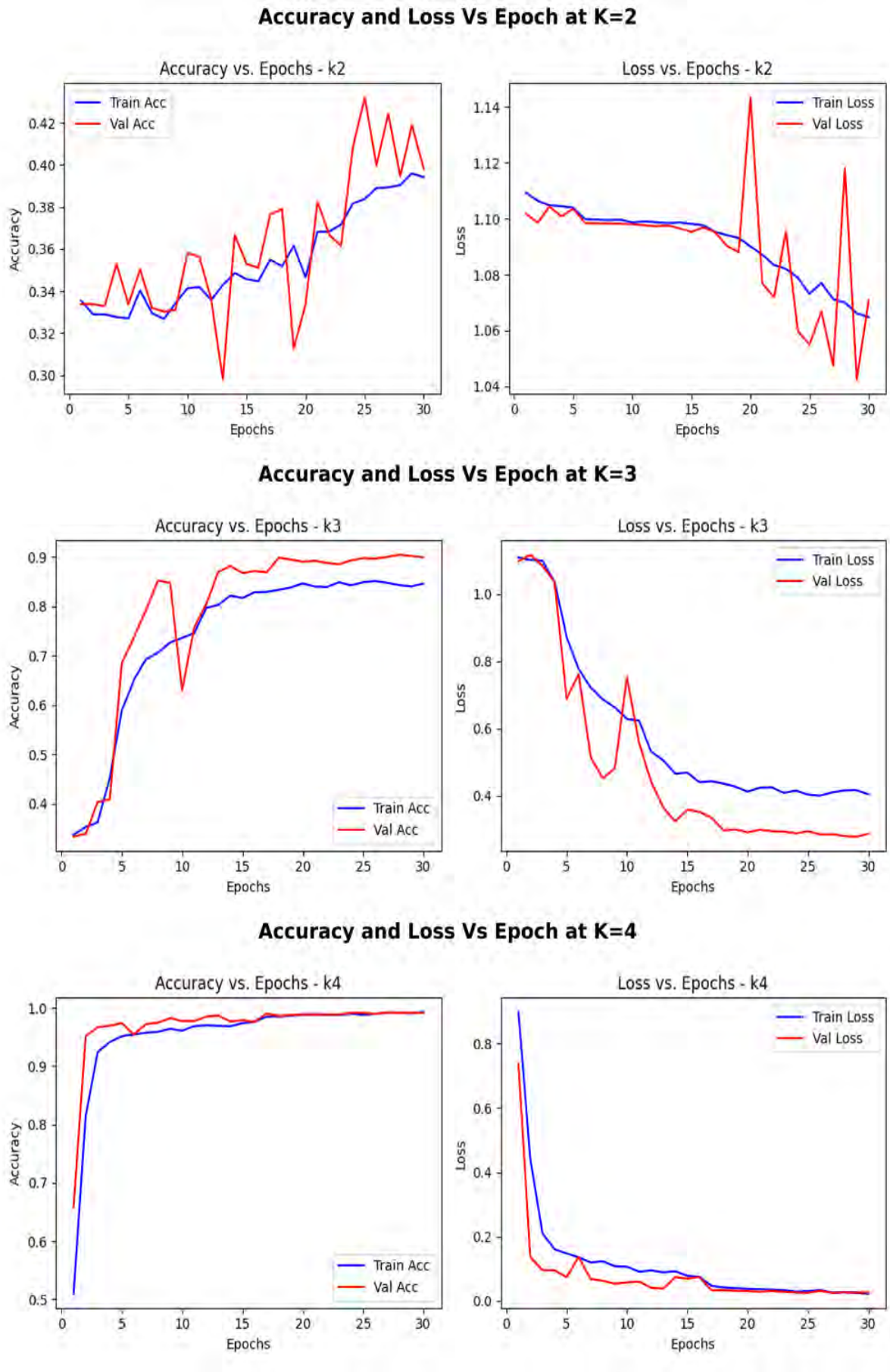


Figure 8. Accuracy and Loss Graphs for DWPCA-RDNet Performance for different k-mer Values

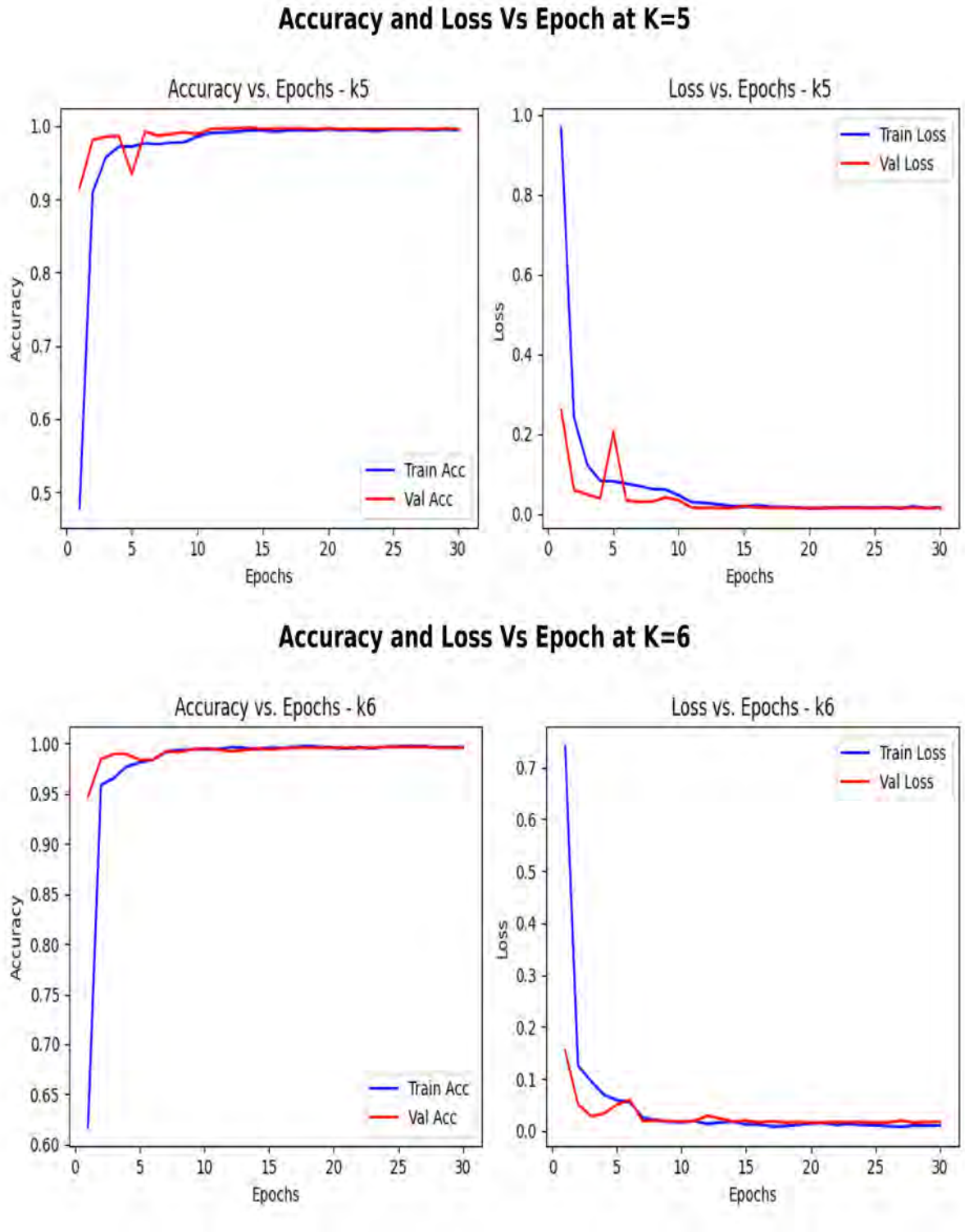
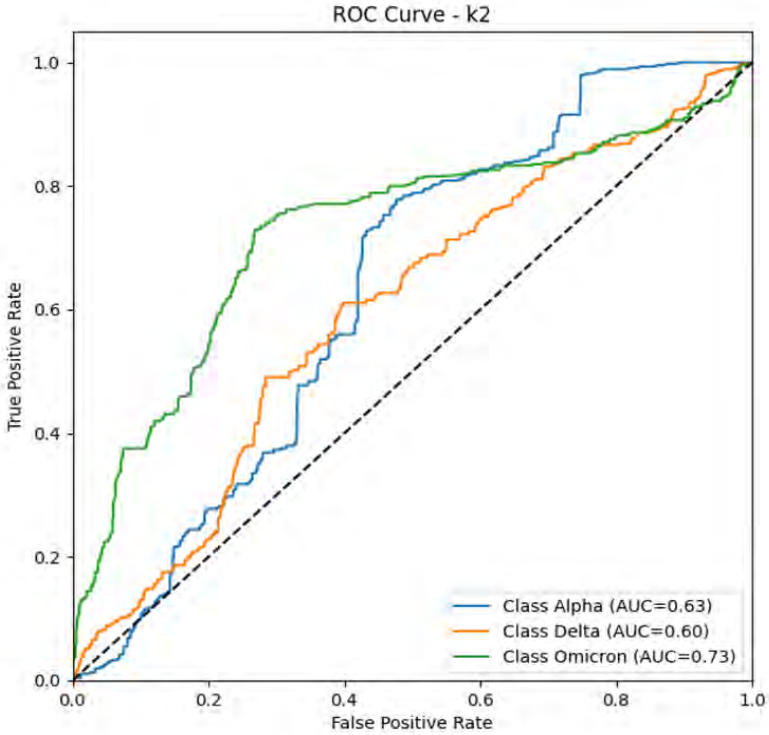


Figure 9. Accuracy and Loss Graphs for DWPCA-RDNet Performance for different k-mer Values

ROC at K=2



ROC at K=3

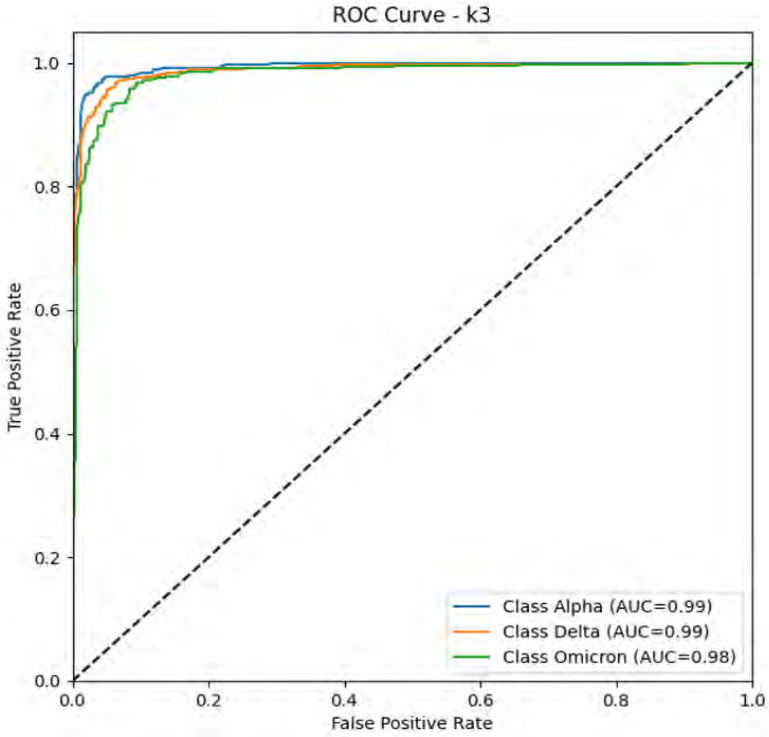


Figure 10. ROC graphs for DWPCA-RDNet Performance Across Different k-mer Values

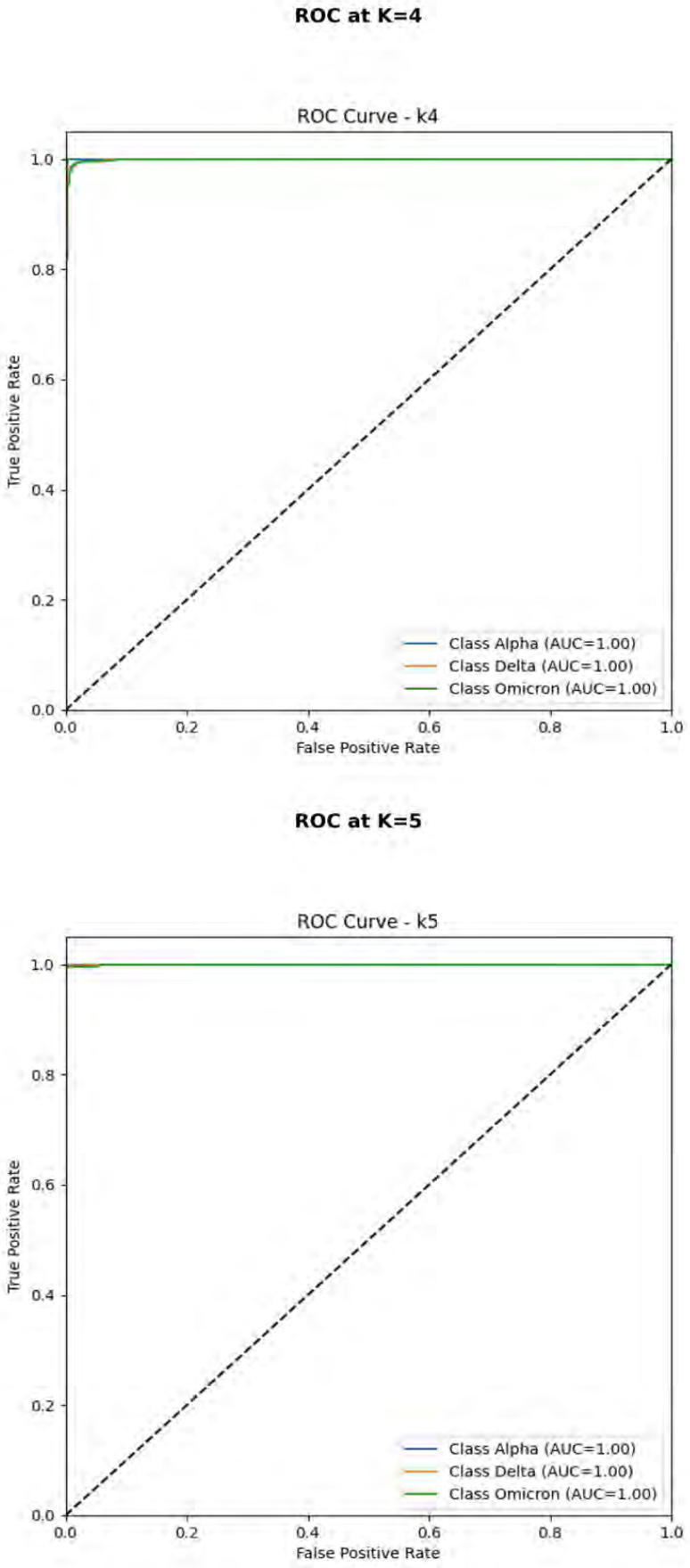


Figure 11. ROC graphs for DWPCA-RDNet Performance Across Different k-mer Values

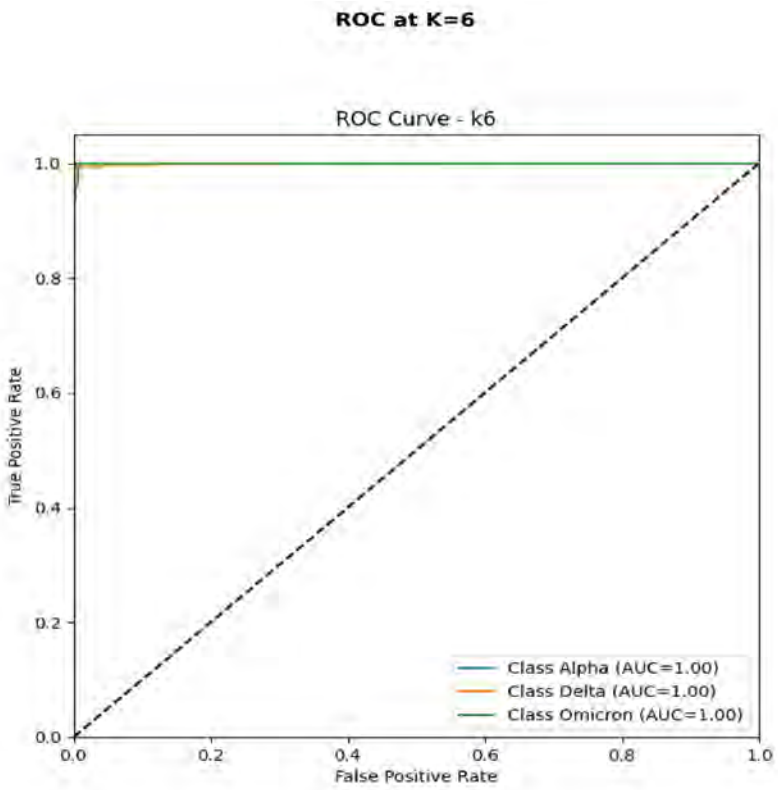


Figure 12. ROC graphs for DWPCA-RDNet Performance Across Different k-mer Values

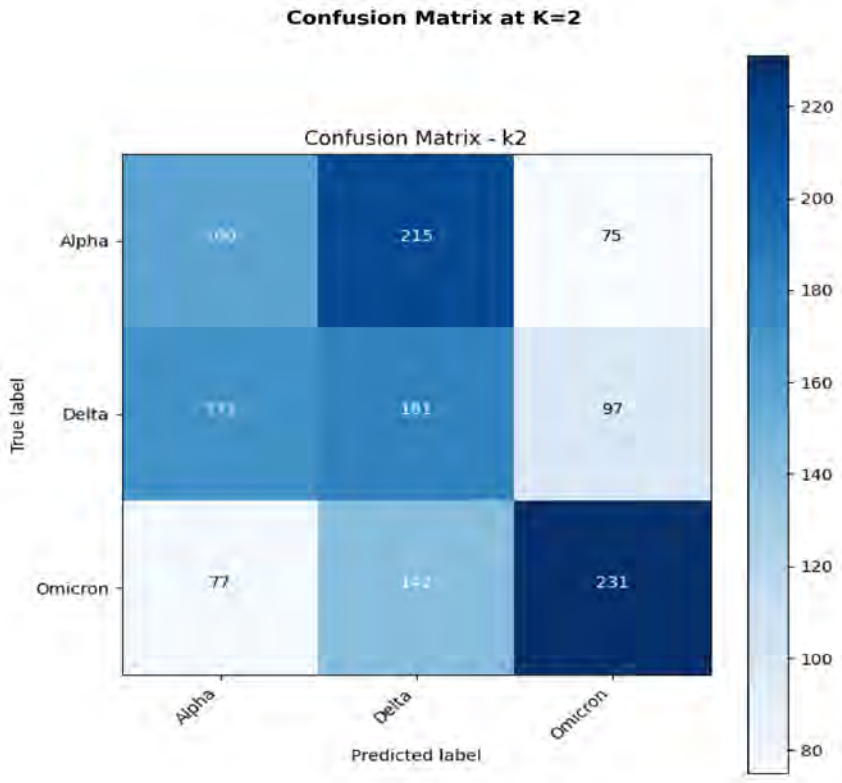


Figure 13. Confusion Matrices for DWPCA-RDNet Performance Across Different k-mer Values

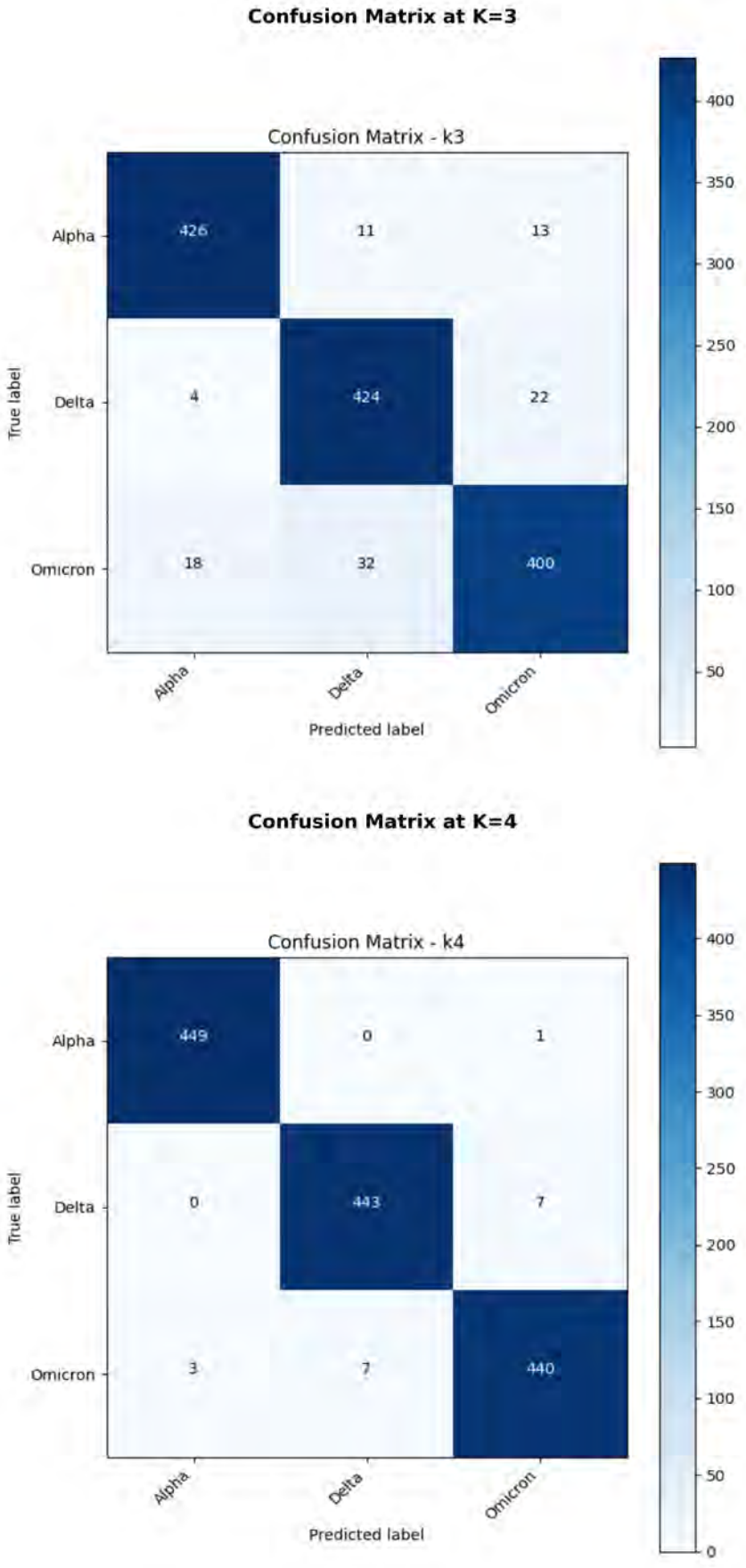


Figure 14. Confusion Matrices for DWPCA-RDNet Performance Across Different k-mer Values

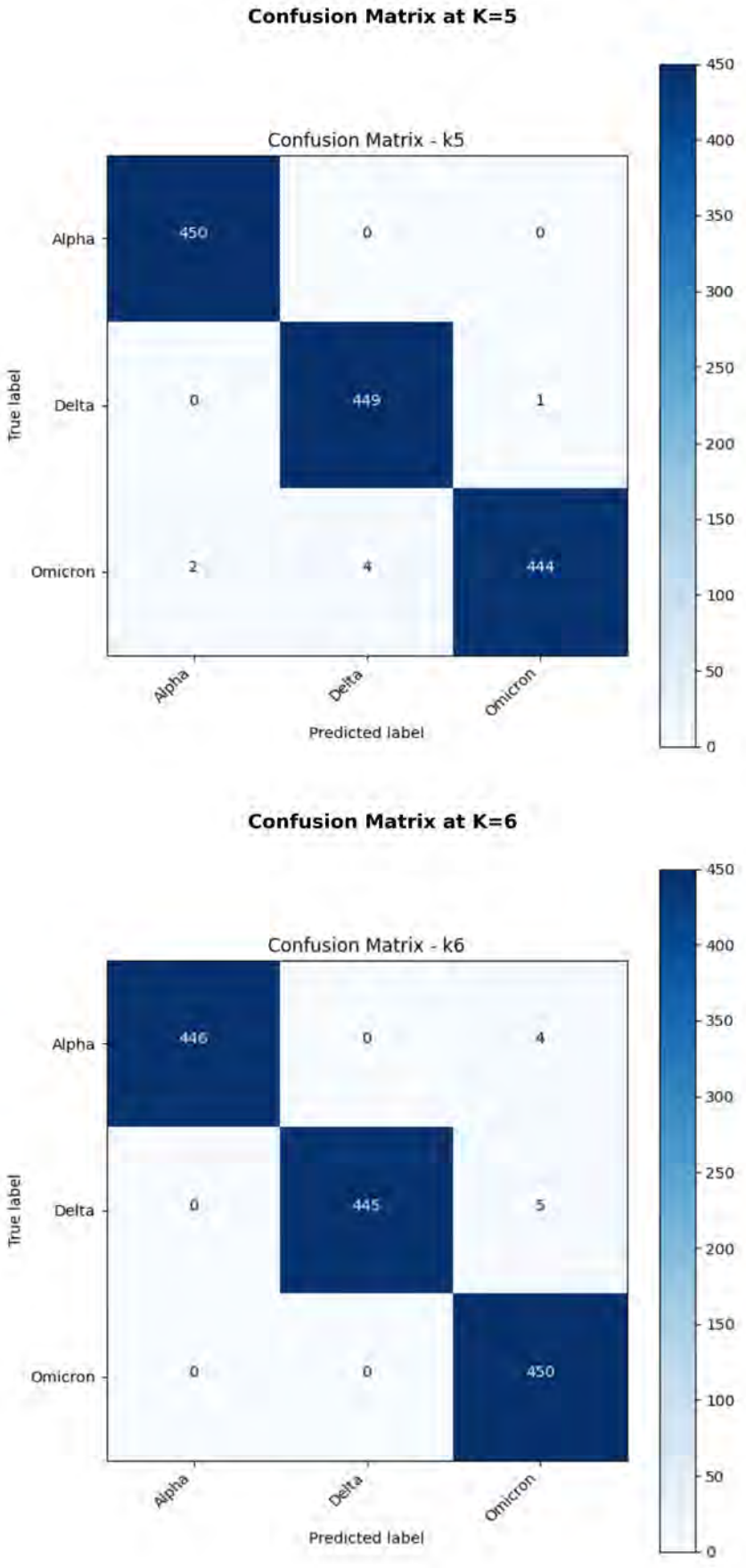


Figure 15. Confusion Matrices for DWPCA-RDNet Performance Across Different k-mer Values

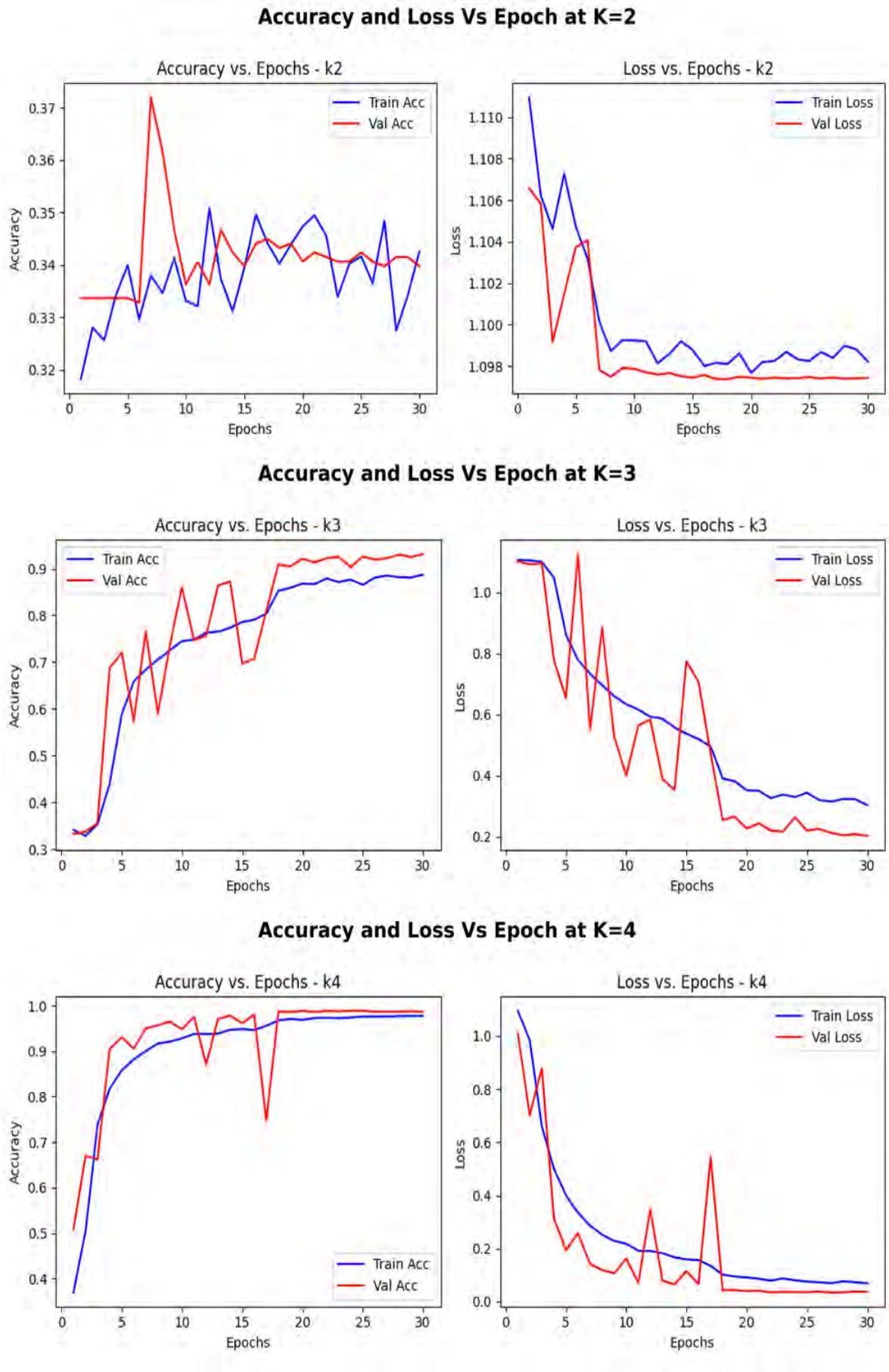


Figure 16. Accuracy and Loss Graphs for DWPCA-DXNet Performance for various k-mer Values

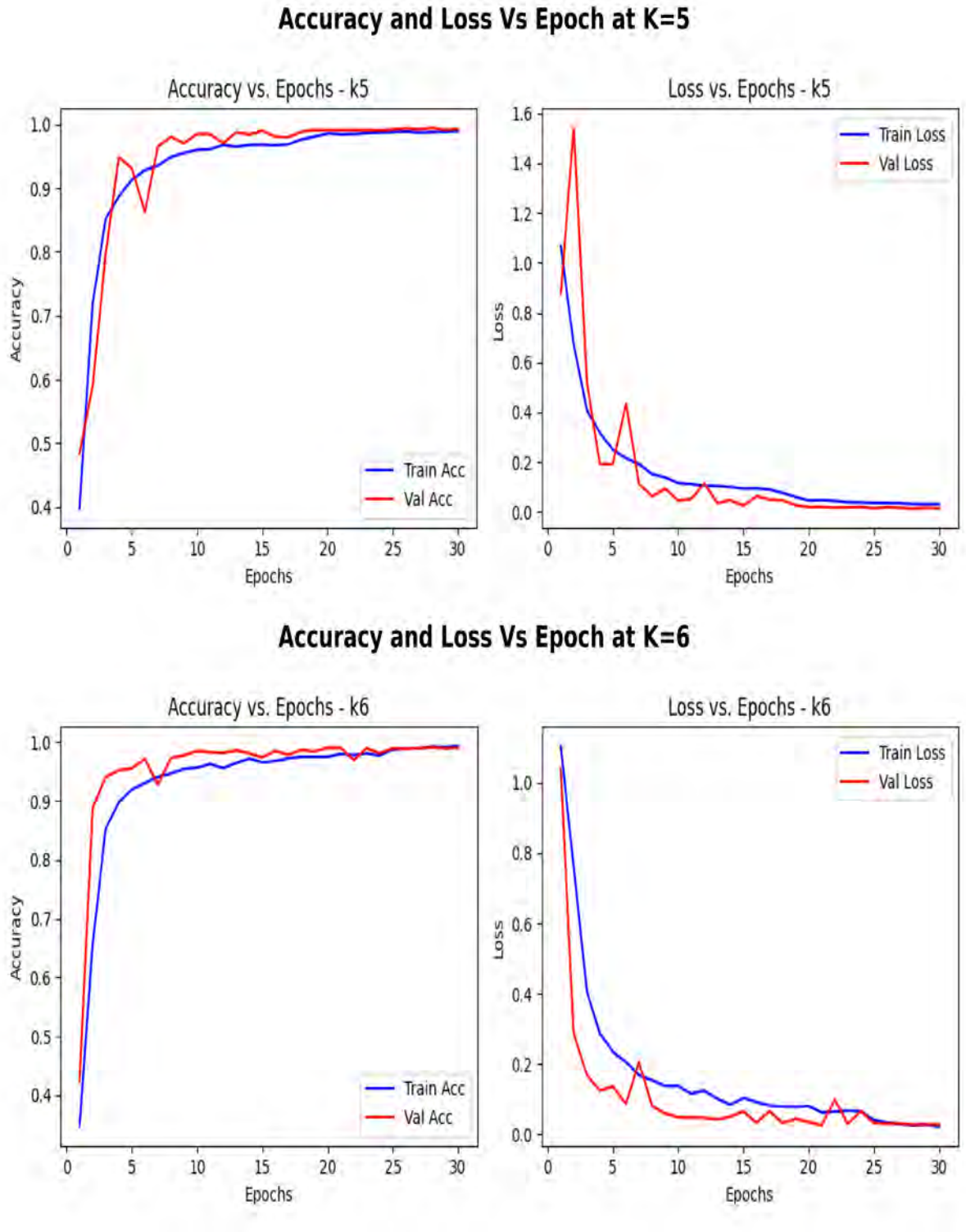


Figure 17. Accuracy and Loss Graphs for DWPCA-DXNet Performance for various k-mer Values

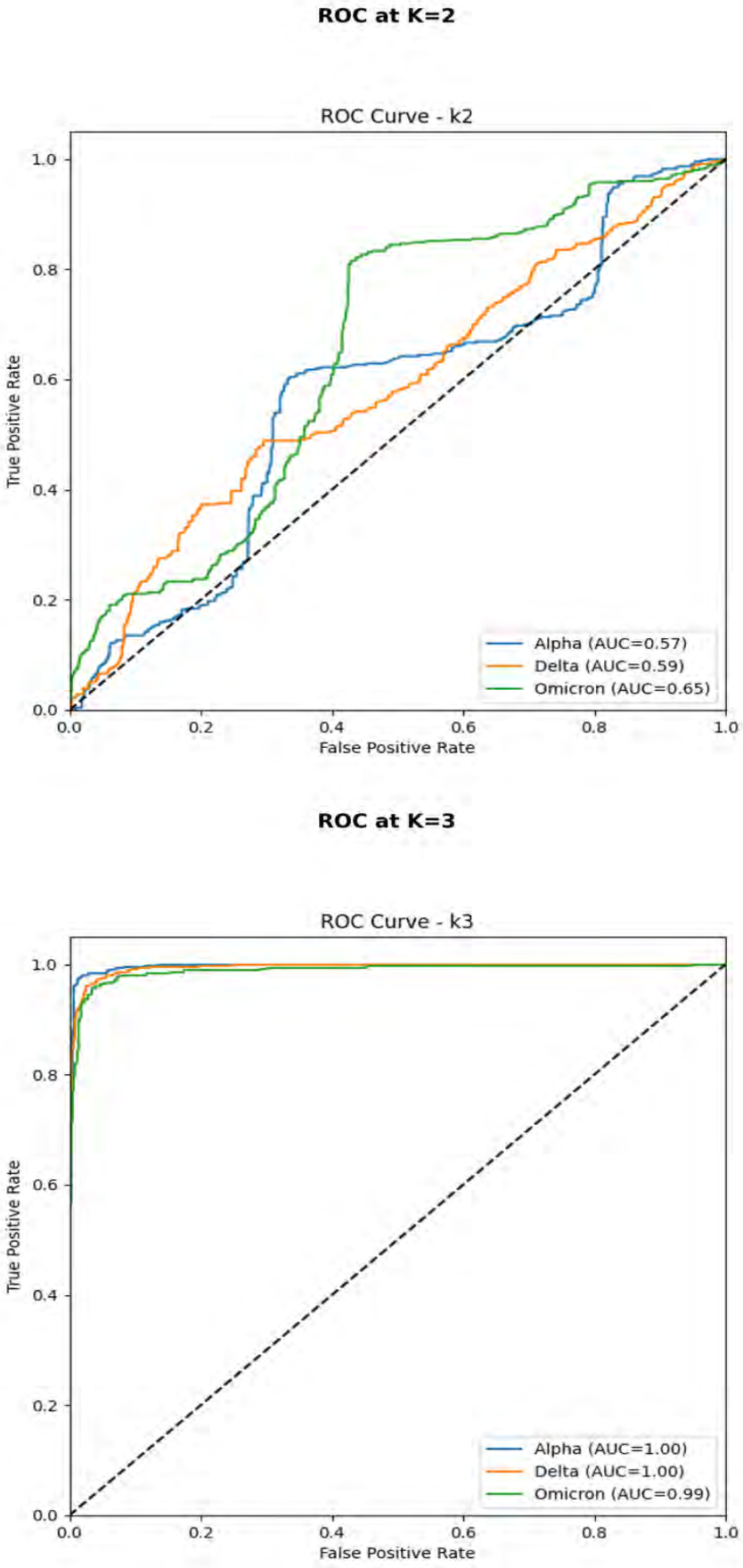


Figure 18. ROC graphs for DWPCA-DXNet Performance Across Different k-mer Values

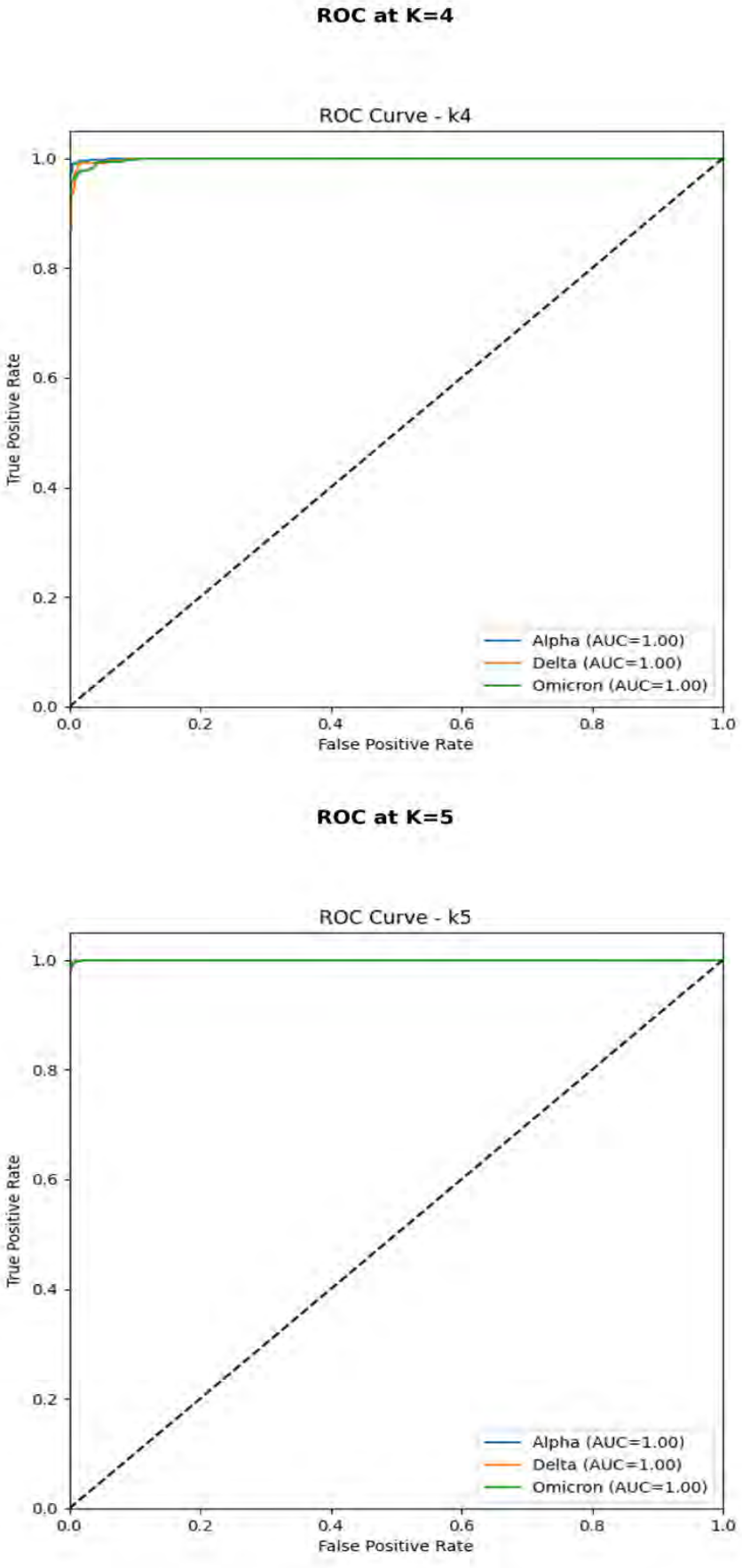


Figure 19. ROC graphs for DWPCA-DXNet Performance Across Different k-mer Values

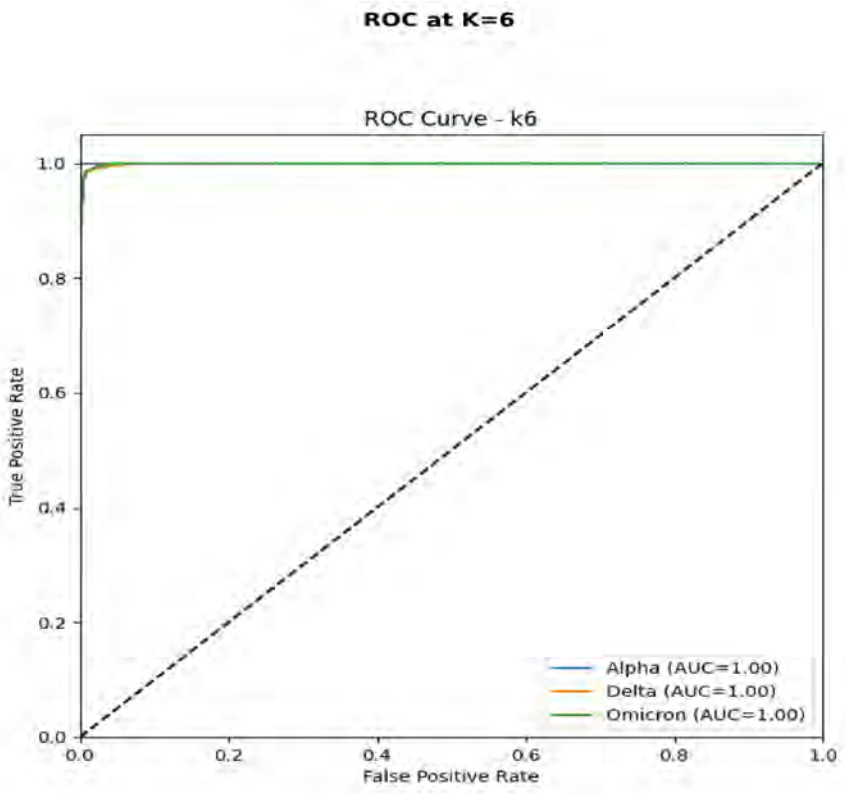


Figure 20. ROC graphs for DWPCA-DXNet Performance Across Different k-mer Values

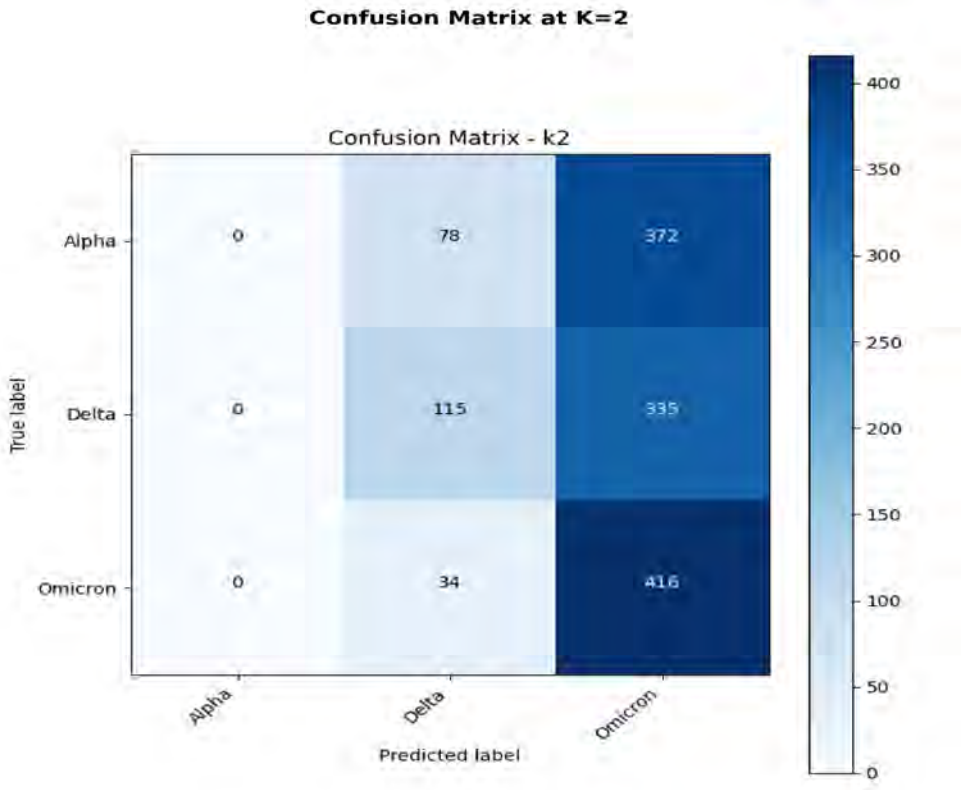


Figure 21. Confusion Matrices for DWPCA-DXNet Performance Across Different k-mer Values

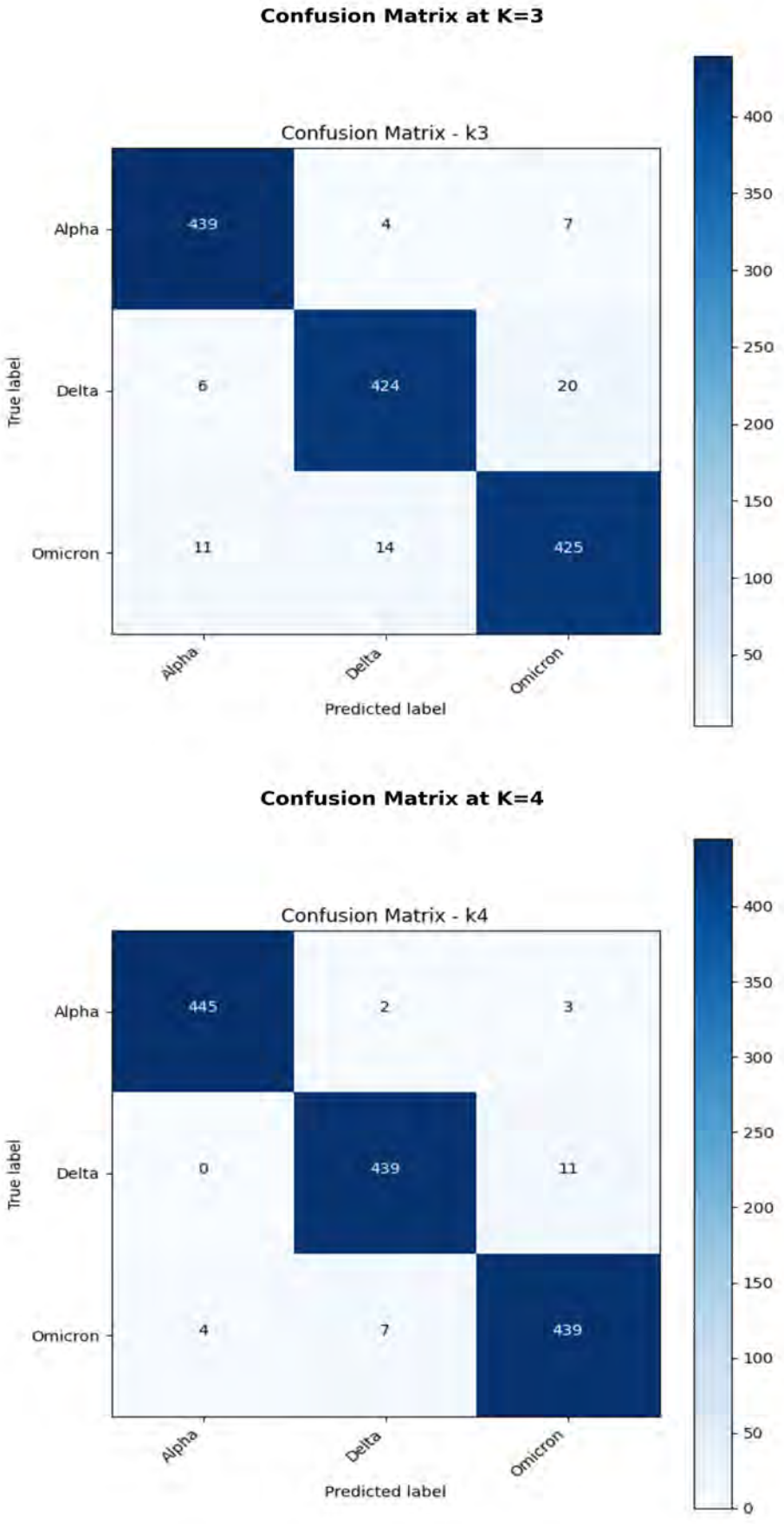


Figure 22. Confusion Matrices for DWPCA-DXNet Performance Across Different k-mer Values

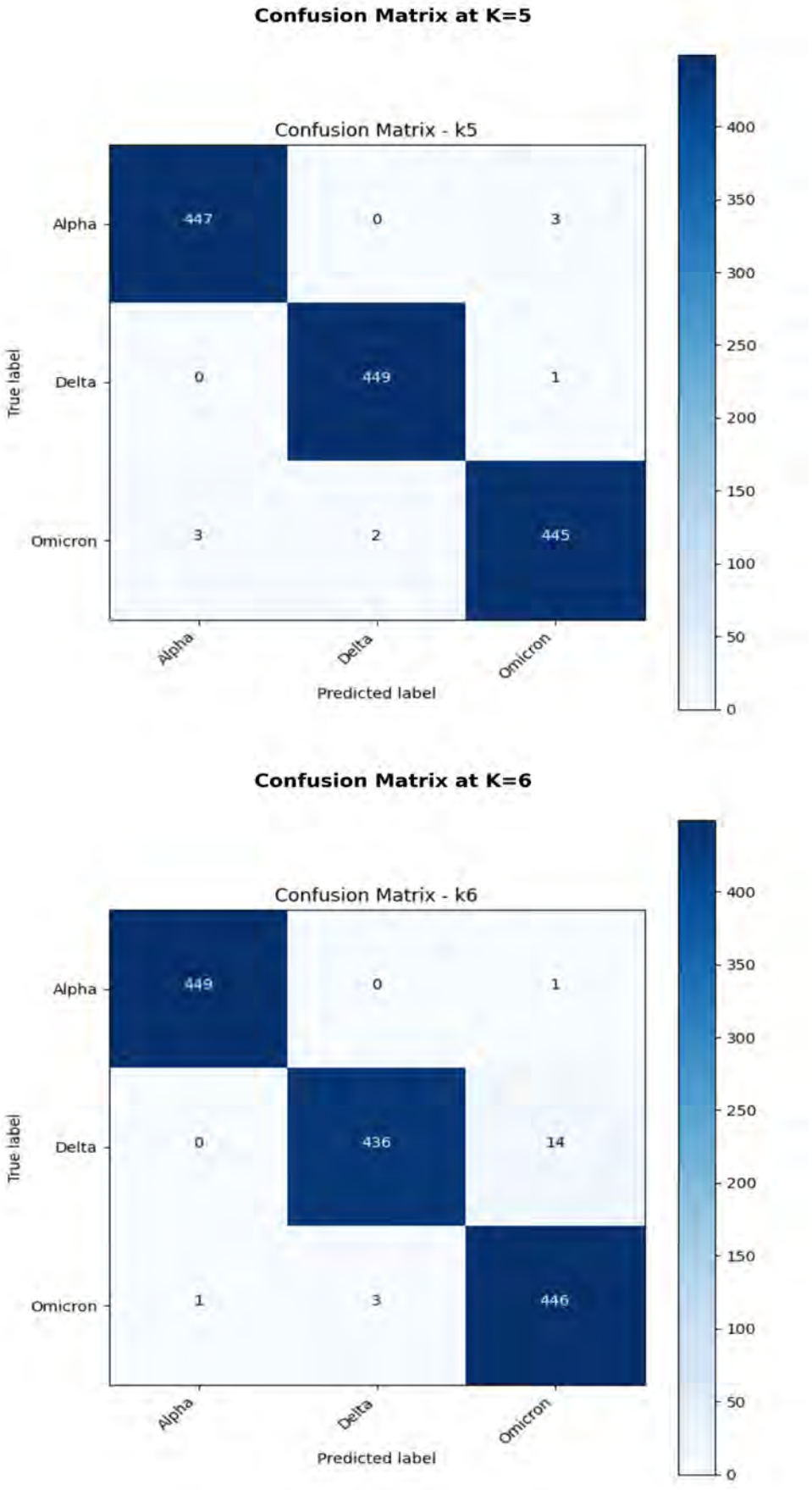


Figure 23. Confusion Matrices for DWPCA-DXNet Performance Across Different k-mer Values

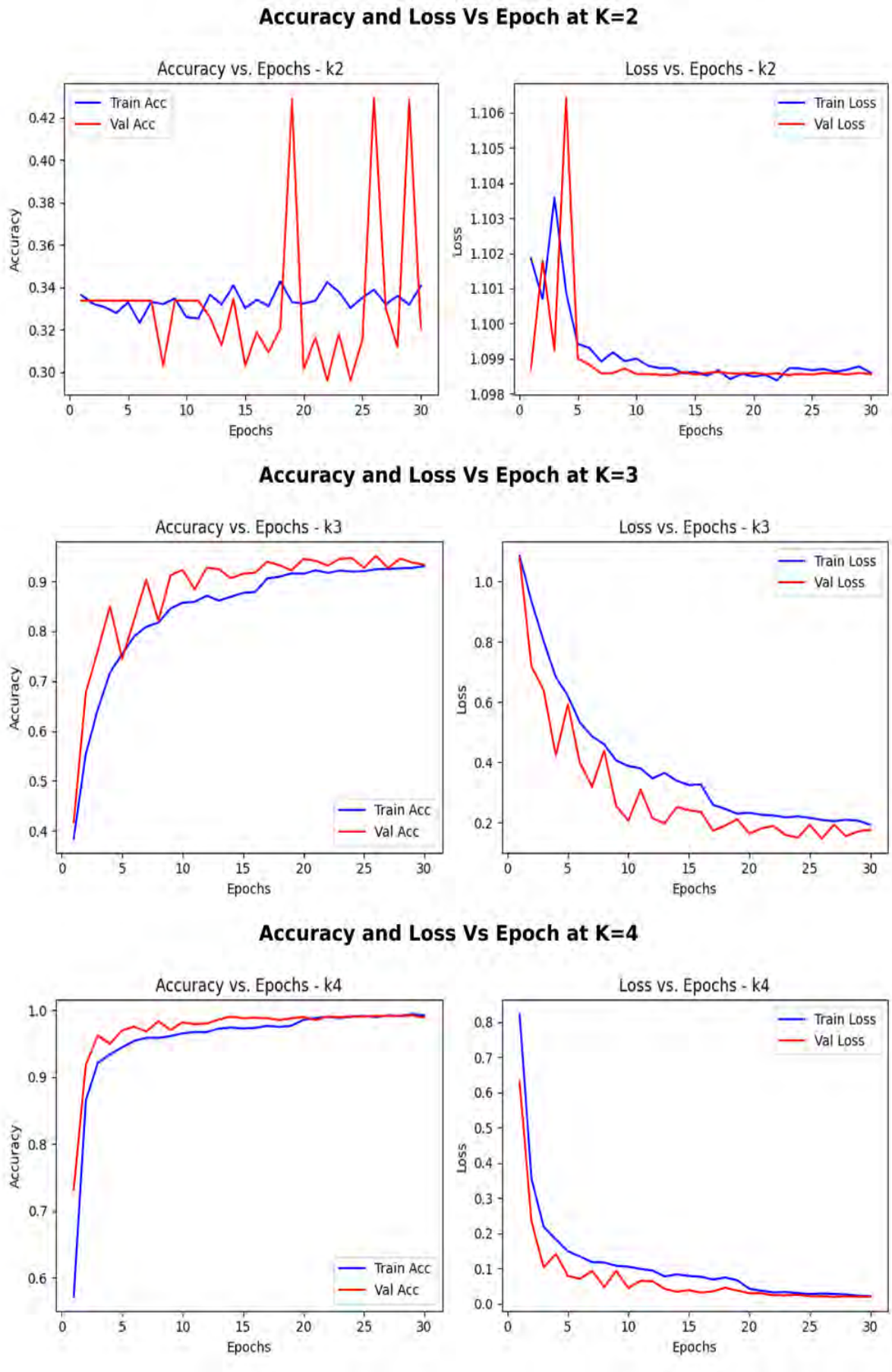


Figure 24. Accuracy and Loss Graphs for DWPCA-XRNet Performance for various k-mer Values

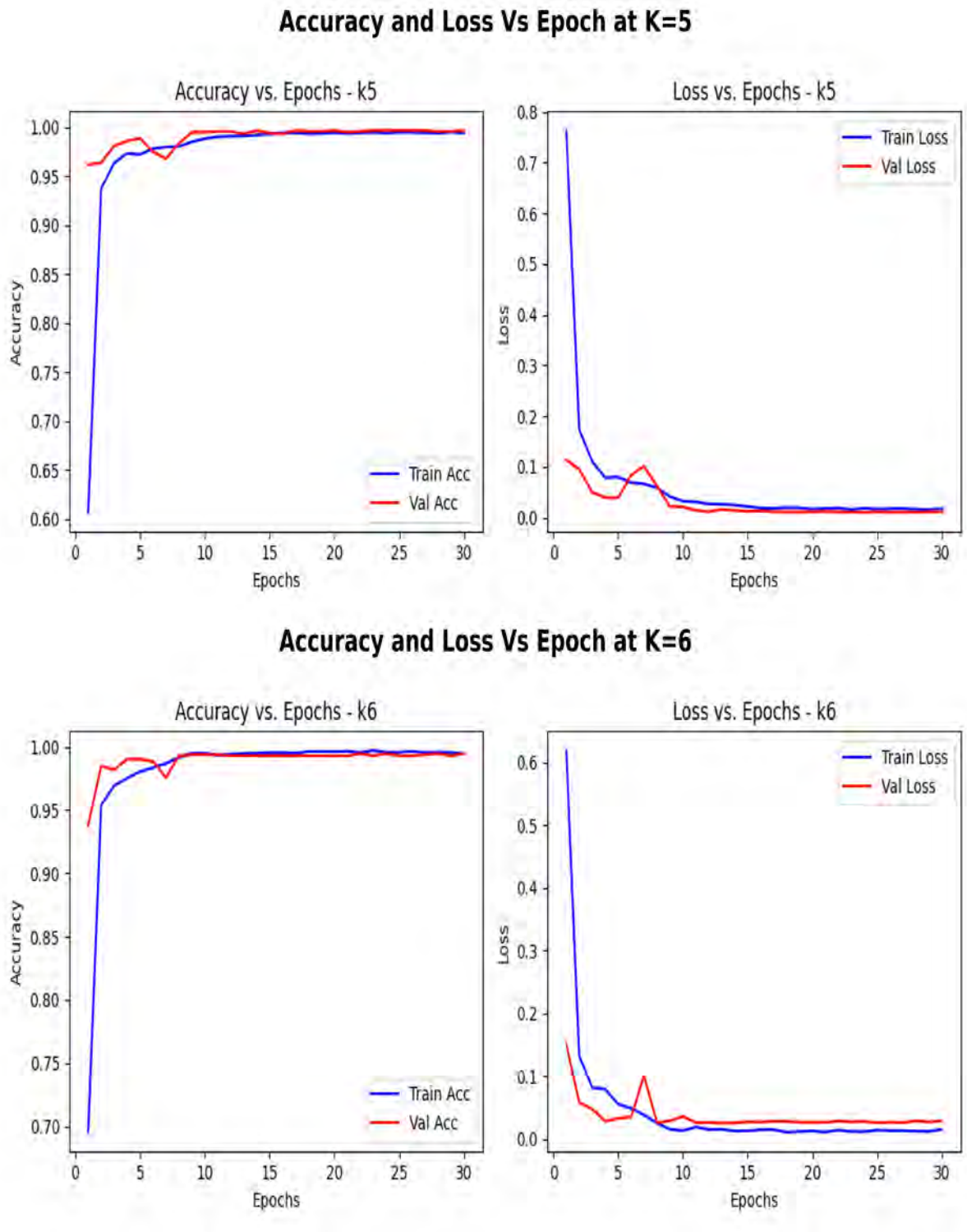


Figure 25. Accuracy and Loss Graphs for DWPCA-XRNet Performance for various k-mer Values

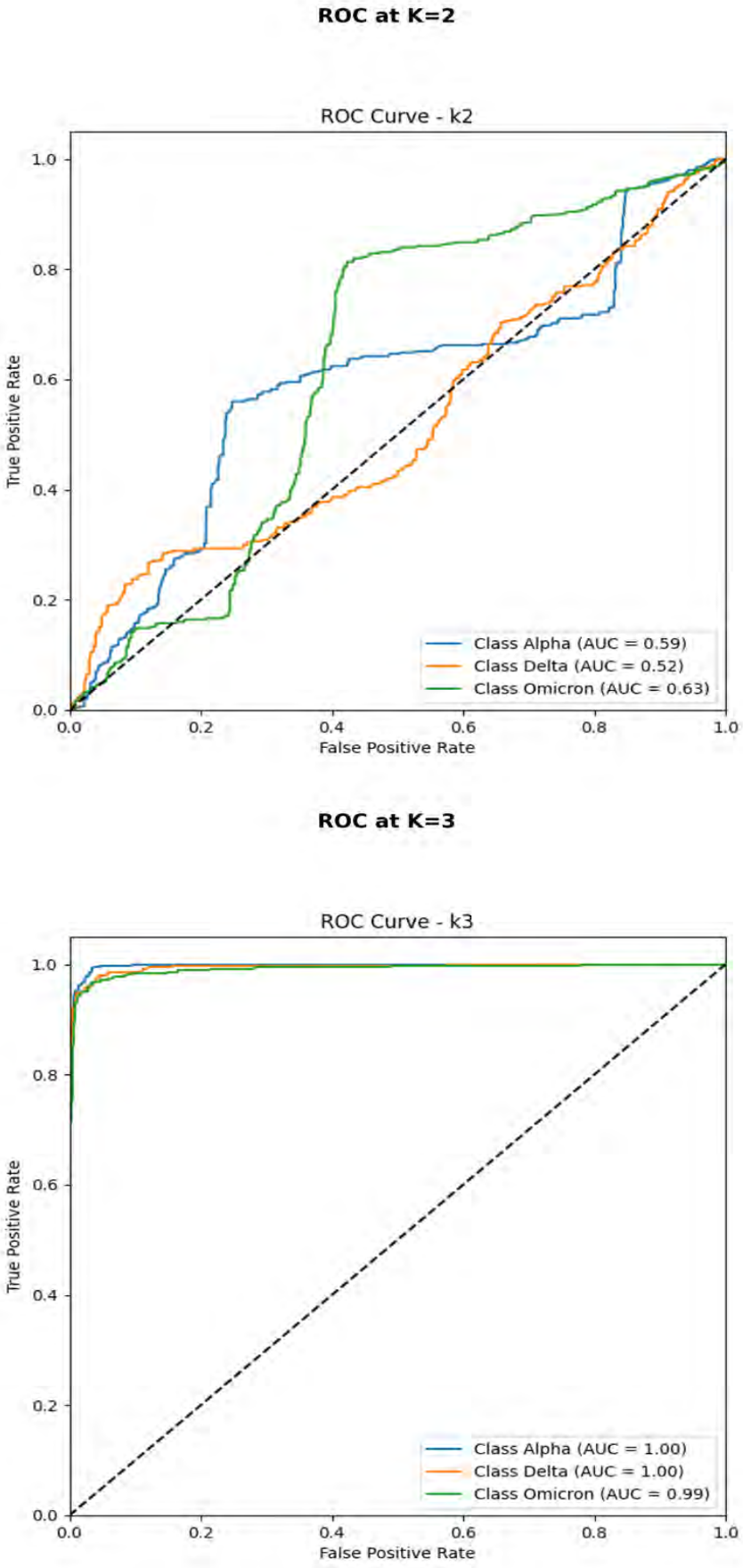


Figure 26. ROC graphs for DWPCA-XRNet Performance Across Different k-mer Values

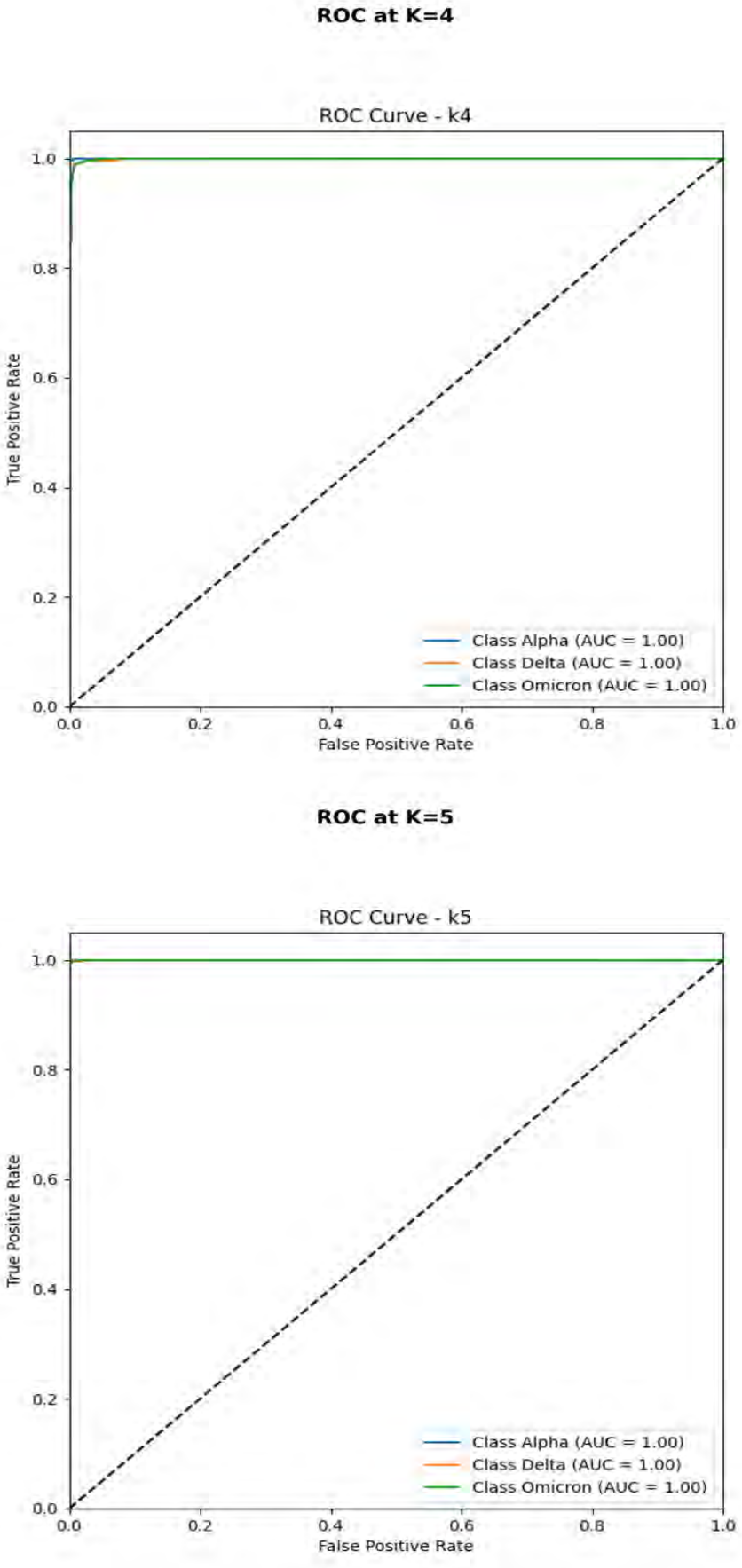


Figure 27. ROC graphs for DWPCA-XRNet Performance Across Different k-mer Values

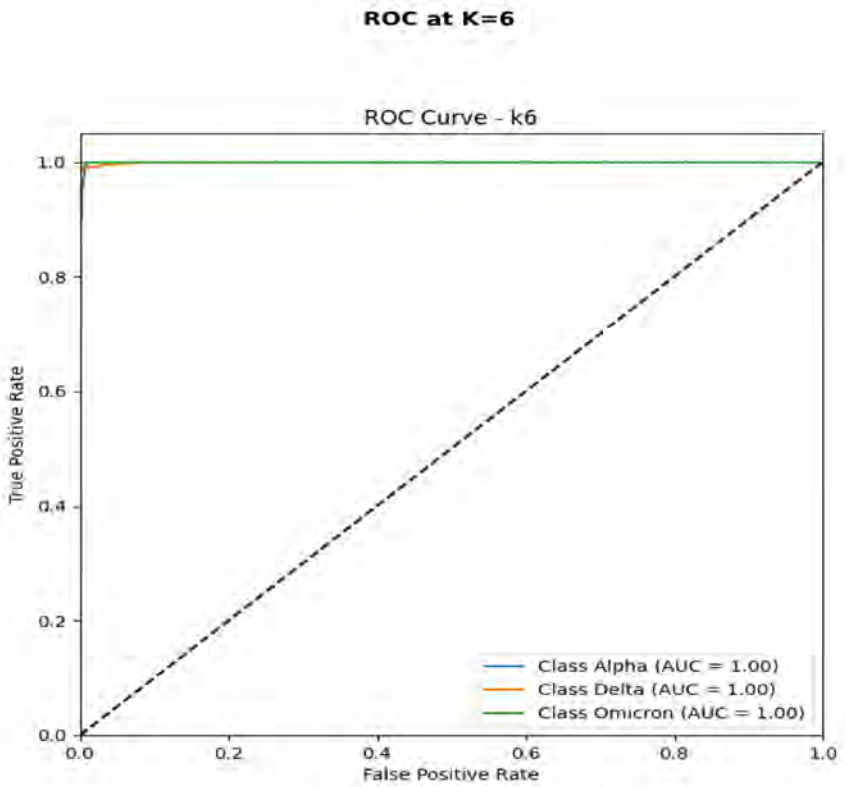


Figure 28. ROC graphs for DWPCA-XRNet Performance Across Different k-mer Values

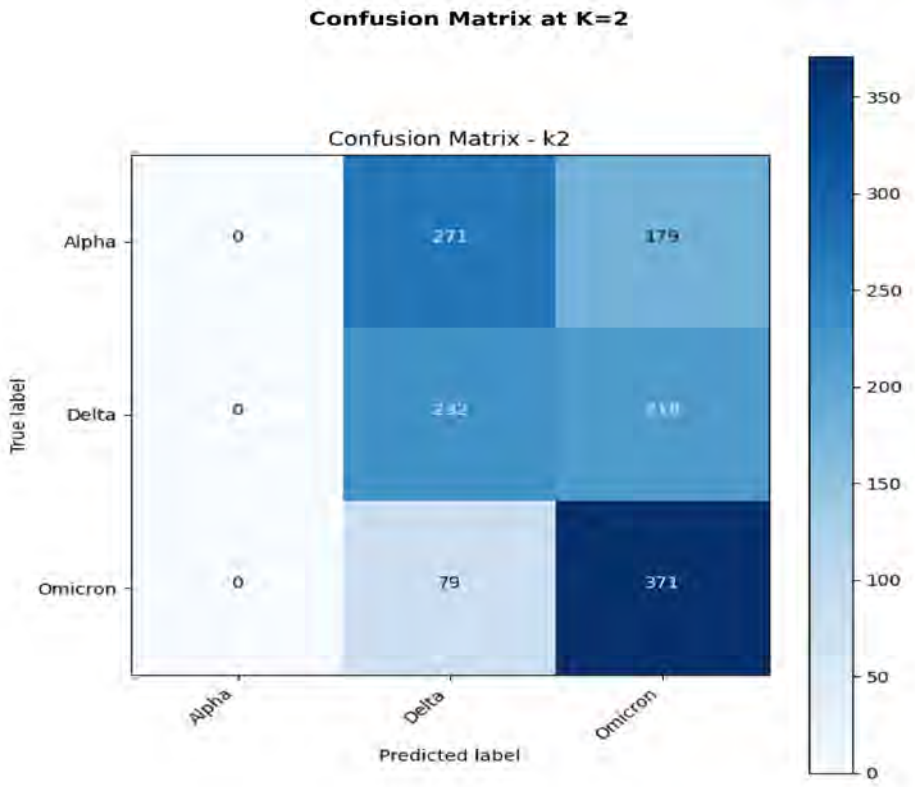


Figure 29. Confusion Matrices for DWPCA-XRNet Performance Across Different k-mer Values

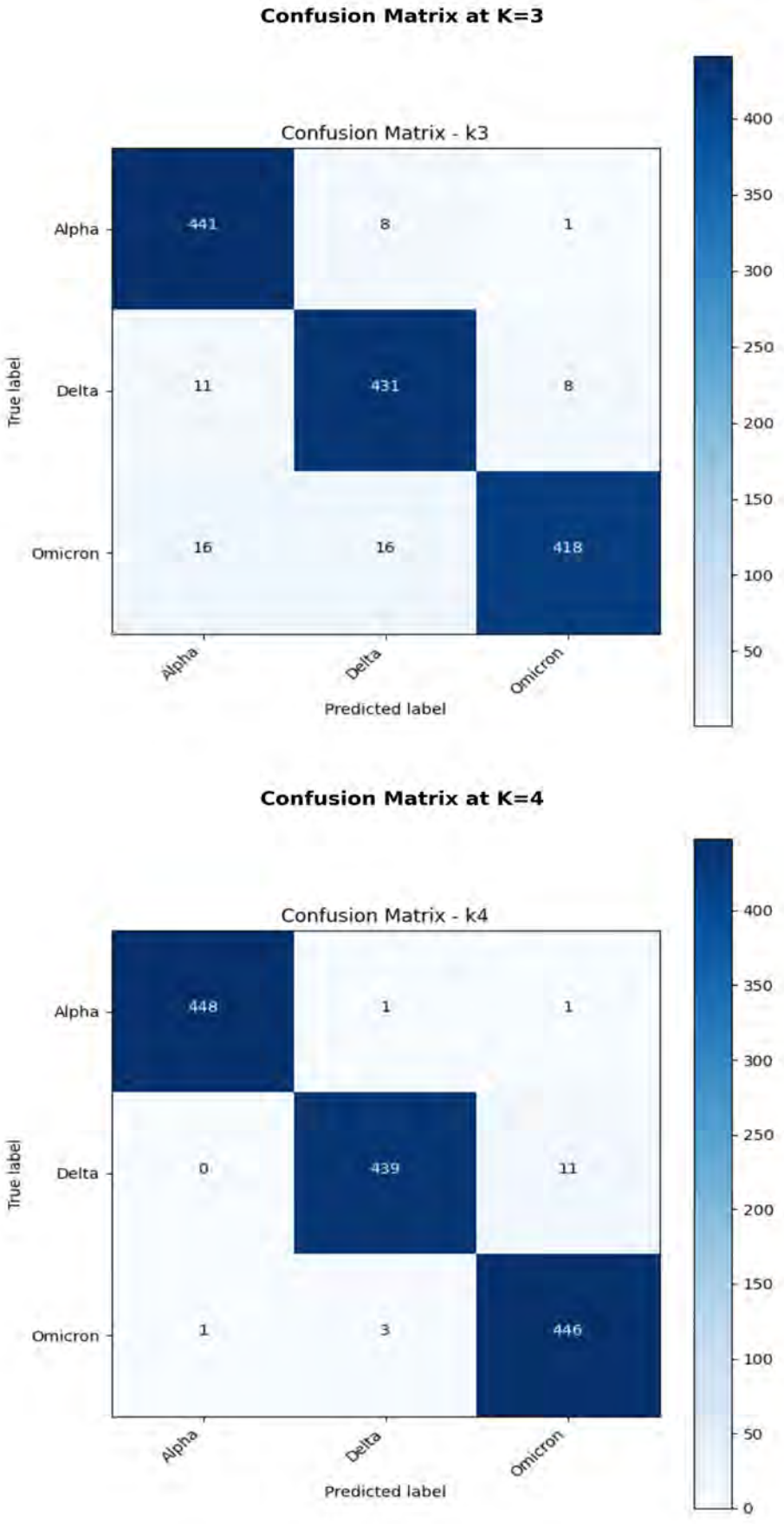


Figure 30. Confusion Matrices for DWPCA-XRNet Performance Across Different k-mer Values

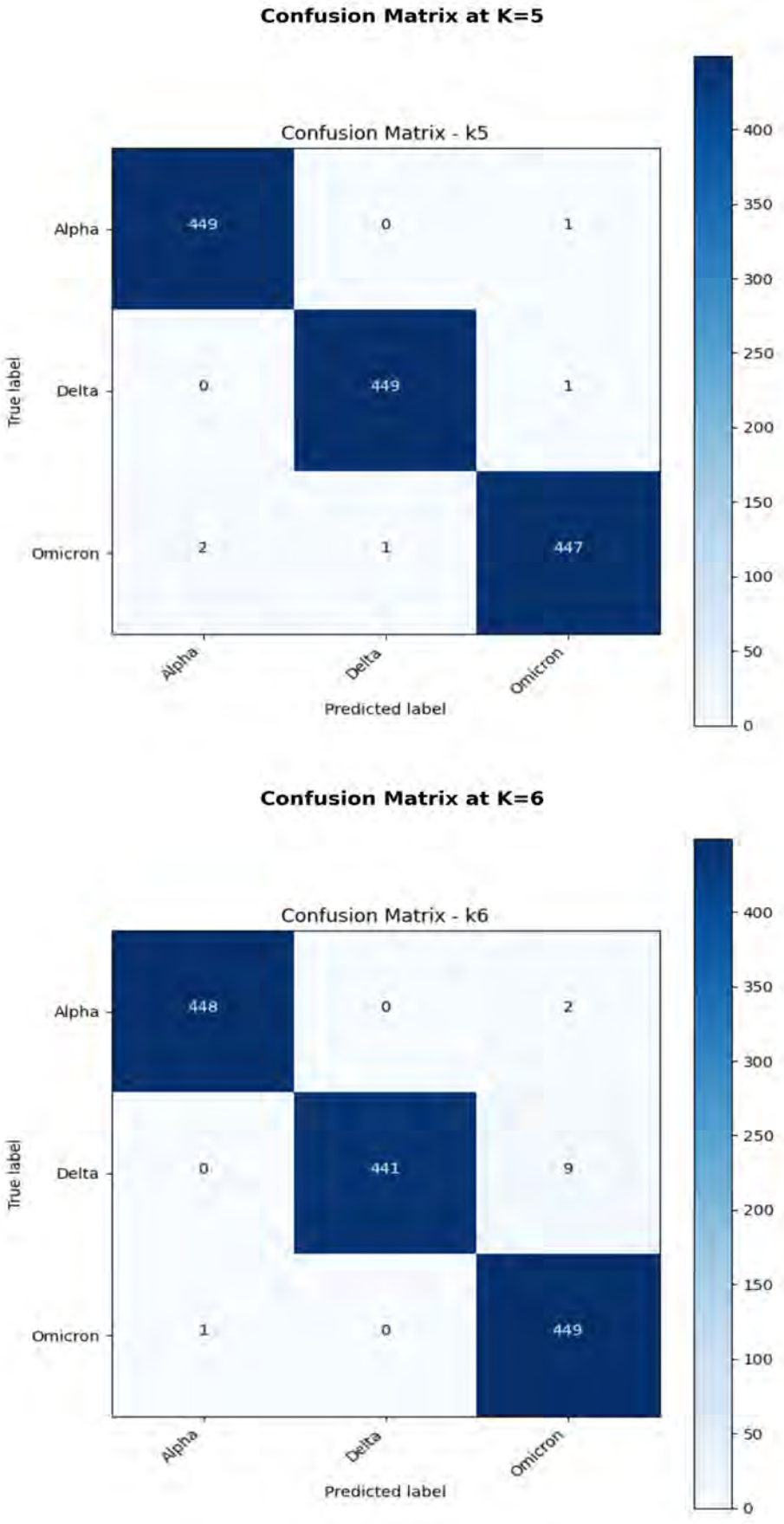


Figure 31. Confusion Matrices for DWPCA-XRNet Performance Across Different k-mer Values

In Figures 24 and 25, the loss and accuracy curves show very unstable training and validation processes for DWPCA-XRNet. The validation accuracy fluctuates much, and the loss values do not converge to a certain value, which means poor generalization. At k=3, one can see some stabilization in the training process: the accuracy steadily increases to high values and the loss consistently decreases, indicating better learning and generalization. At k=4 to k=6, the models achieve near-perfect accuracy within a few epochs, with validation and training curves converging closely. Loss values drop to near-zero, indicating optimal performance and minimal overfitting. Moreover, at k=2, ROC curves for DWPCA-XRNet in Figures 26,27,and 28 depicts suboptimal performance, where all classes values of AUC Alpha: 0.59, Delta: 0.52, and Omicron: 0.63 indicate poor separability. The curves are a long way off the top-left corner, which hints at a high false- positive rate. At k=3, the AUC improves dramatically, reaching near-perfect values (Alpha and Delta: 1.00, Omicron: 0.99). The curves now closely follow the top-left trajectory, which reflects a better discriminative power. At k=4 and above (k=5, k=6), the AUCs achieve perfect scores (1.00) for all classes, demonstrating that the models are able to distinguish classes perfectly. The ROC curves are ideal, with zero false-positive rates across all thresholds. Besides this, at k=2, as shown in Figures 29, there is a lot of misclassification across all classes as can be seen by strong off-diagonal values; for example, many Alpha and Delta instances are misclassified as Omicron. At k=3, misclassifications decrease significantly and diagonal dominance becomes apparent, meaning better class separability. At k=4 to k=6, the confusion matrices display near-perfect classification, with minimal off-diagonal values. Most instances are correctly classified, as shown by the strong diagonal dominance. This trend is observed for all the models, where they perform increasingly well with larger k-mer sizes, specifically k=4 and above, which are near perfect. The ROC curves, loss metrics, and confusion matrices all show that these models are learning strong representations, which give way to accurate predictions and very few misclassifications at higher k-mer sizes.

4.2 Comparitive Analysis for Different Orders of MTF

Table 4. Summary of Performance Metrics for MTF of Feature Fusion

Sr. No	Model	Order	Accuracy	Precision	Recall	F1-Score
1	DWPCA-RDNet	2	92.14%	92.23%	92.18%	92.19%
2	DWPCA-DXNet	2	88.00%	88.36%	88.00%	88.18%
3	DWPCA-XRNet	2	92.14%	92.14%	92.14%	92.14%
4	DWPCA-RDNet	3	98.44%	98.45%	98.44%	98.45%
5	DWPCA-DXNet	3	98.00%	98.01%	98.00%	98.01%
6	DWPCA-XRNet	3	98.88%	98.90%	98.88%	98.88%
7	DWPCA-RDNet	4	99.33%	99.33%	99.33%	99.33%
8	DWPCA-DXNet	4	99.48%	99.47%	99.47%	99.48%
9	DWPCA-XRNet	4	98.81%	98.82%	98.81%	98.81%

Table 4 illustrates an overview of performance for MTF while comparing the different orders for DWPCA-RDNet, DWPCA-DXNet, and DWPCA-XRNet by accuracy, precision, recall, and F1-score. For Order=2, the models have good per-formances, with the accuracies between 88.00% (DWPCA-DXNet) and 92.14% (DWPCA-RDNet and DWPCA-XRNet). Both DWPCA-RDNet and DWPCA-XRNet have the same outcomes on all metrics, so they are of the same performances, while DWPCA-DXNet is a bit weaker in terms of metrics by about 88%, which means it has slightly weaker generalization ability at this order. Starting from an order up to 3 and above, there is a large performance gain in the models. DWPCA-XRNet has the best with 98.88% accuracy and corresponding 98.88% or more precision, recall, and F1-scores; DWPCA-RDNet and DWPCA-DXNet also perform fairly well at this order in terms of accuracy, by 98.44% and 98.00%, respectively implying a firmly consistent good classification performance beginning with this order. The models come to almost perfect classification metrics at Order=4: DWPCA-DXNet is 99.48%, and DWPCA-RDNet trails with 99.33%. DWPCA-XRNet, in turn, shows slightly lower metrics than what it has shown in Order=3, namely 98.81%, but it also evidences high efficacy in the task. The obtained results confirm one more time the efficiency of higher-order MTF representations in improving the performance of classification.

From Table 4, it is generally shown that the models increase in efficacy as the MTF order increases, with Order=4 showing the best results for most of the metrics. DWPCA-XRNet performed best at Order=3, while DWPCA-DXNet showed the highest performance at Order=4, reflecting their ability to capture meaningful patterns for robust classification. At Order=2, the accuracy curves in Figure 32 show a steady improvement for both training and validation, with the DWPCA-RDNet reaching high accuracy by the end of training. The loss curves drop smoothly and consistently, meaning that learning is effective and convergence is assured. At Order=3, both training and validation accuracy are at near-perfect levels (1.0) within a few epochs. The loss reduces rapidly and stabilizes at minimal values, showing efficient learning and at Order=4, the model achieves near-perfect training and validation accuracy very early in training. Loss curves stabilize at almost zero, which implies optimal learning with no signs of overfitting. Furthermore, at Order=2, the ROC curves in Figure 33 also show very promising performance in terms of AUC values very close to 0.99 for all classes: Alpha,

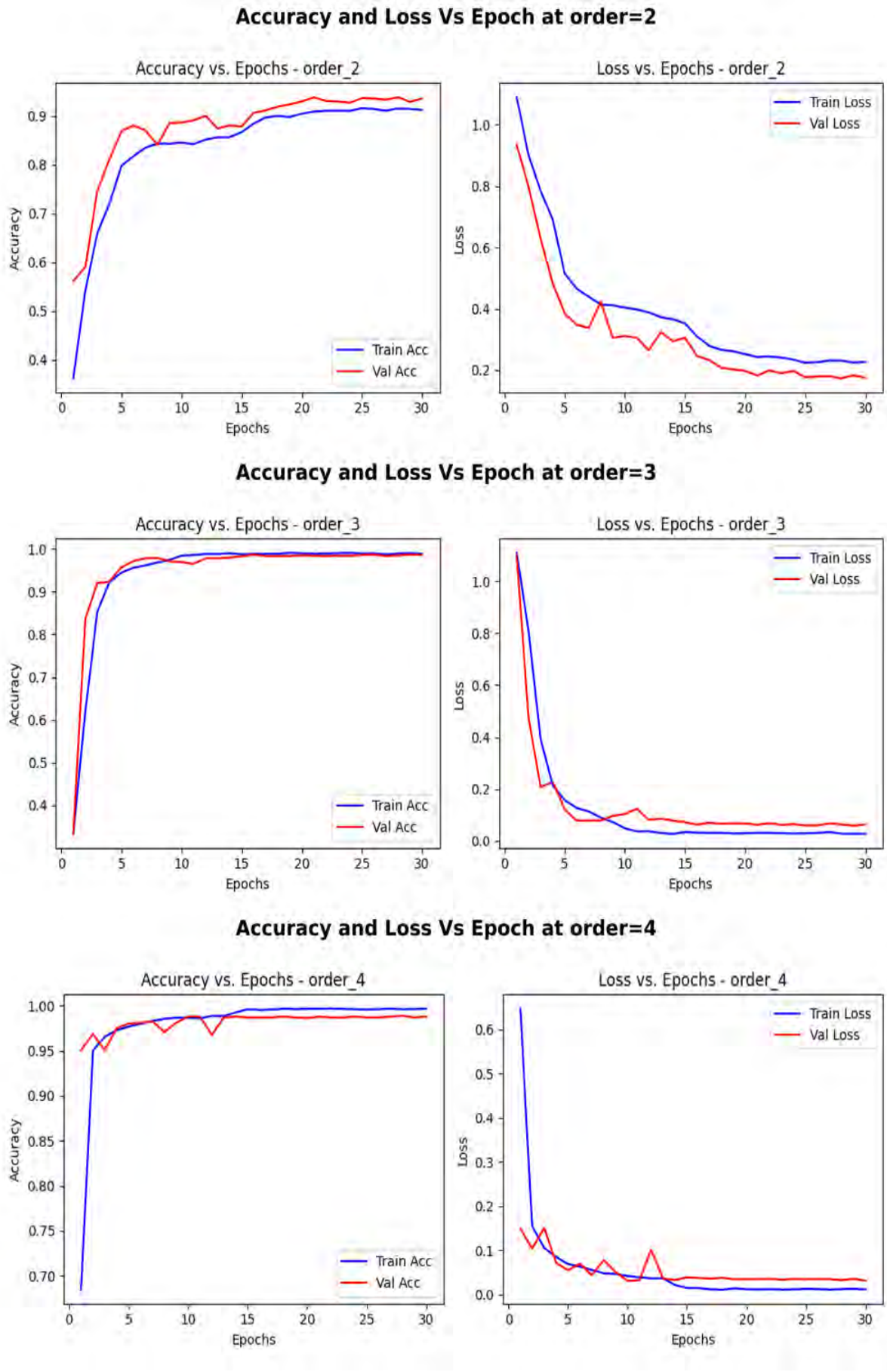


Figure 32. Accuracy and Loss Graphs for DWPCA-RDNet Performance Across Different MTF Order Values

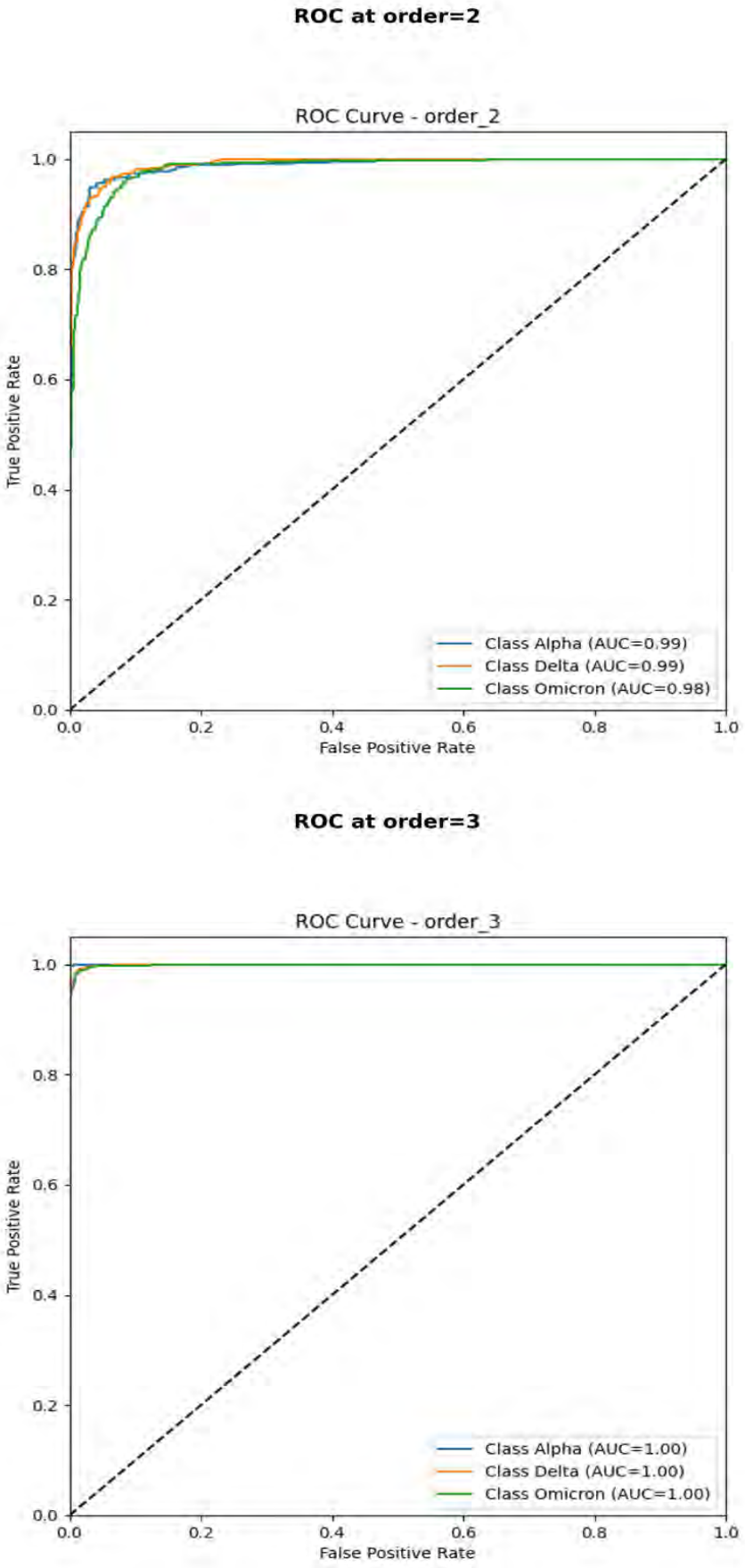


Figure 33. ROC graphs for DWPCA-RDNet Performance Across Different MTF Order Values

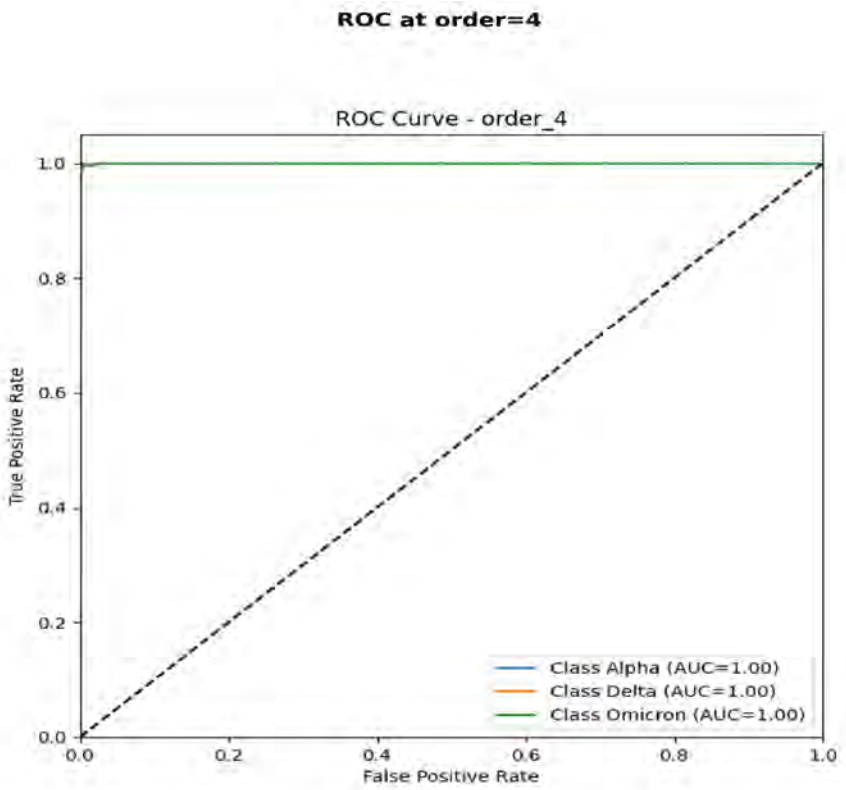


Figure 34. ROC graphs for DWPCA-RDNet Performance Across Different MTF Order Values

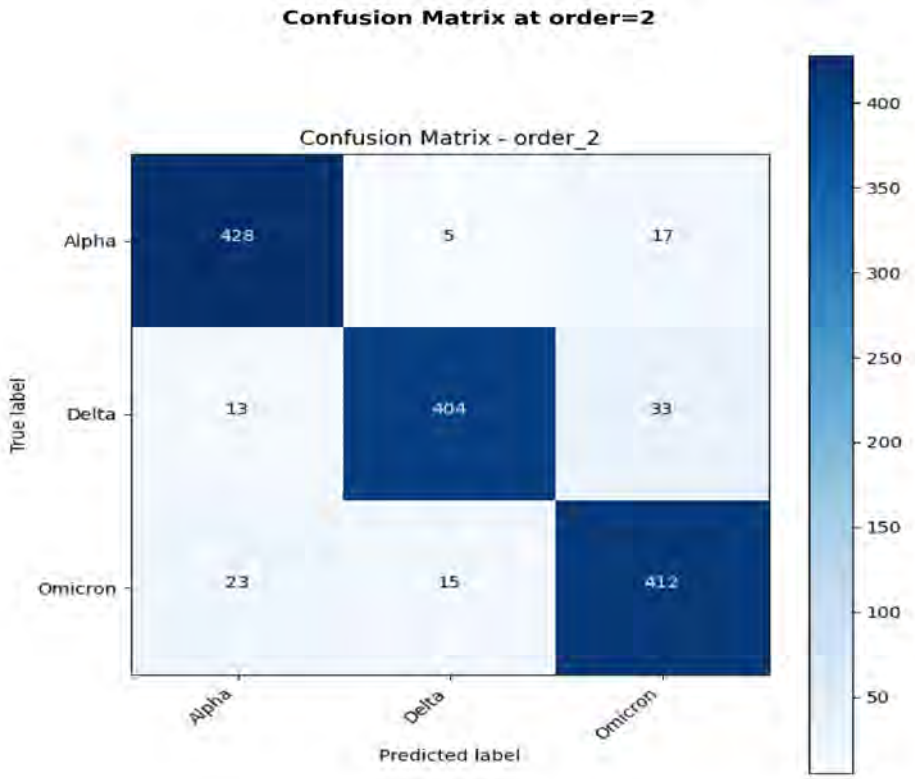


Figure 35. Confusion Matrices for DWPCA-RDNet Performance Across Different MTF Order Values

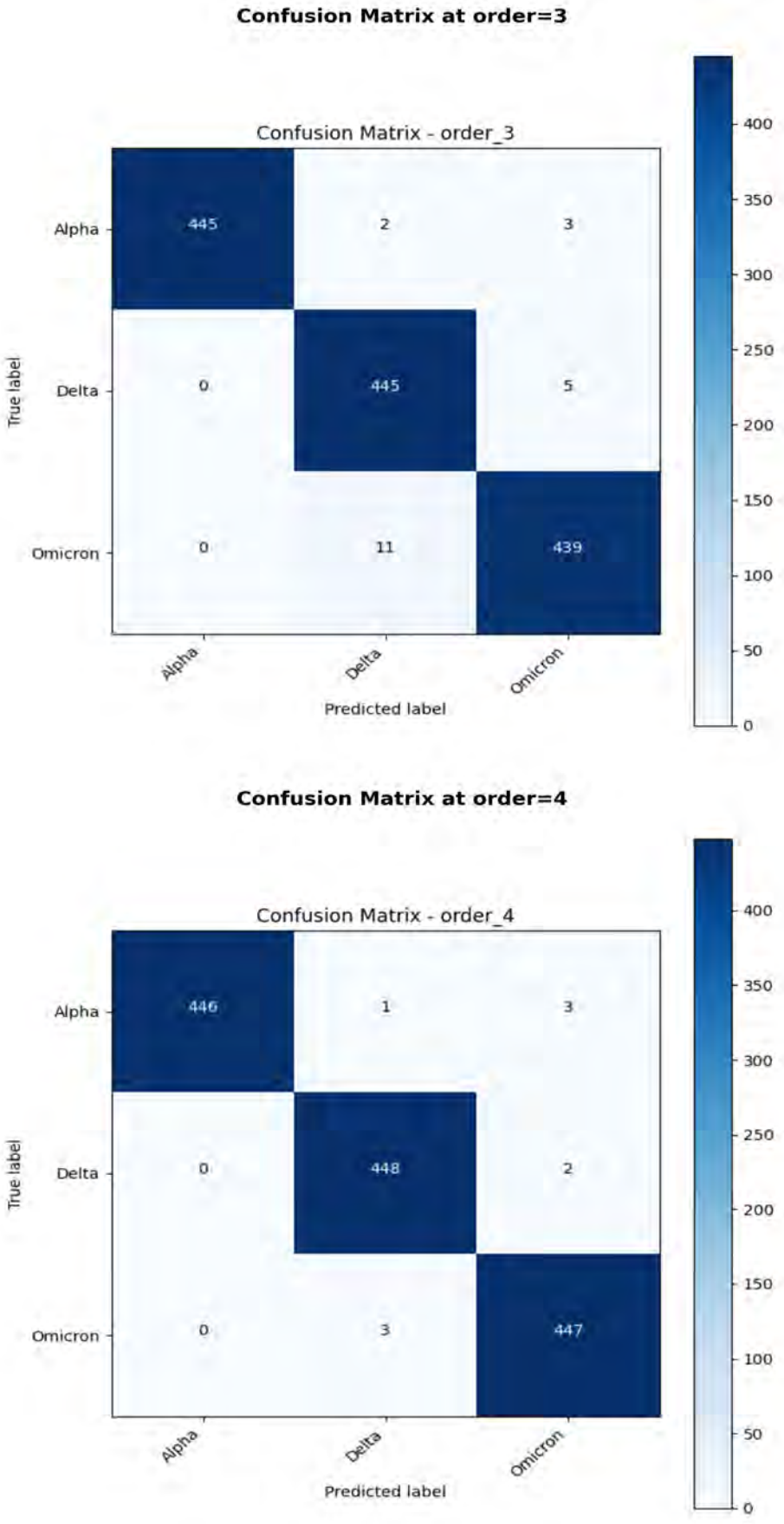


Figure 36. Confusion Matrices for DWPCA-RDNet Performance Across Different MTF Order Values

Delta, and Omicron. The results thus indicate good class discrimination abilities of the method, although there are still some false positives. For Order=3, the AUC values improve further to reach perfect scores of AUC=1.00 for all classes, indicating excellent separability with very minimal classification errors. In the case of Order=4, the best AUC values are maintained (AUC=1.00 for all classes), with the ROC curves showing perfect behavior in being very close to the top-left corner, indicating perfect classification performance. In addition, the confusion matrix in Figure 35 for order=2 is diagonally dominant but does contain some non-zero off-diagonal elements that imply misclassifications occurred. For example, Alpha has been misclassified as Omicron and vice versa. At Order=3, diagonal dominance improves with very few misclassifications; the vast majority of the instances are well-classified. For the case of Order=4, the confusion matrix shows close-to-perfect classification with almost negligible off-diagonal values; the majority of the predictions are equal to the actual labels.

In the case of DWPCA-DXNet at Order=2, training and validation accuracy, as shown in Figure 37, gradually improve but are slightly lower compared to higher orders. The loss decreases considerably, but stabilizes at a higher value than those observed at orders 3 and 4, which would suggest minor misclassifications. At Order=3, both the training and validation accuracy increase to almost perfect levels very early in the epochs (1.0). Its loss curves drop to near-zero, indicating efficient and consistent learning and for order=4, the model achieves perfect training and validation accuracy very quickly. The loss is minimal and converges smoothly, which shows no overfitting and a highly effective learning process. Besides this, the ROC curves in Figure 38 for order=2 show strong performance at AUC values of 0.97 for Alpha, 0.98 for Delta, and 0.97 for Omicron. This suggests a high degree of separability of the classes, although there is a small number of false positives. At Order=3, the AUC values are perfect AUC=1.00 for all classes: Alpha, Delta, and Omicron. This indicates perfect classification, where the curves are hugging the top-left corner. At Order=4, the models continue to maintain their perfect AUC values each is 1.00 for all classes also showing they are correctly classified without any mistakes. Their ROC curves also remain ideally uncurved. In Figure 40, at Order=2, the confusion matrix shows good diagonal dominance but with some large off-diagonal values. Indeed, misclassifications have occurred across all classes, for example, Alpha being predicted as Omicron or Delta. At Order=3, misclassifications are drastically reduced. In fact, most of the predictions perfectly match with the actual labels. The off-diagonal values are very low and the diagonals completely dominate. At Order=4, the confusion matrix shows almost perfect classification, with nearly all instances classified correctly. Off-diagonal values are negligible, meaning the model performs exceptionally well at this order.

In Figure 43, for Order=2, the ROC curves are powerful with the AUC values of 0.99 for Alpha and Delta and 0.98 for Omicron, reflecting strong class separability. Minor misclassifications still occur as the curves do not completely hug the top-left corner. For the case of Order=3, the AUC values are perfect AUC=1.00 for all classes which means perfect classification with zero false positives and complete separation among the classes. At Order=4, the models still have perfect AUC values of 1.00 for all classes. The curves are ideally aligned with the top-left corner, proving the consistency of the optimal performance. Furthermore, in Figure 45, the confusion matrix at Order=2 shows strong diagonal dominance but with small off-diagonal values. For instance, Alpha is sometimes predicted as Delta or Omicron and vice versa. At Order=3, the off-diagonal values are almost all cleared, and most of the predictions agree with the true labels correctly, demonstrating a huge gain in classification accuracy. At Order=4, the confusion matrix shows almost perfect classification with nearly zero misclassifications. All the predictions are concentrated along the diagonal, showing the outstanding precision and recall of the model. In can be seen that the models trained with MTF images yield very strong classification performances over a range of Markov orders. The models achieve near-perfect accuracy, minimal loss, and flawless ROC-AUC values as the order increases. From Order=3 onwards, the models exhibit optimal learning and very good class separability, as depicted by the ROC curves and confusion matrices. These results confirm that the Markov-based feature representations are powerful in extracting class-specific patterns for robust classification.

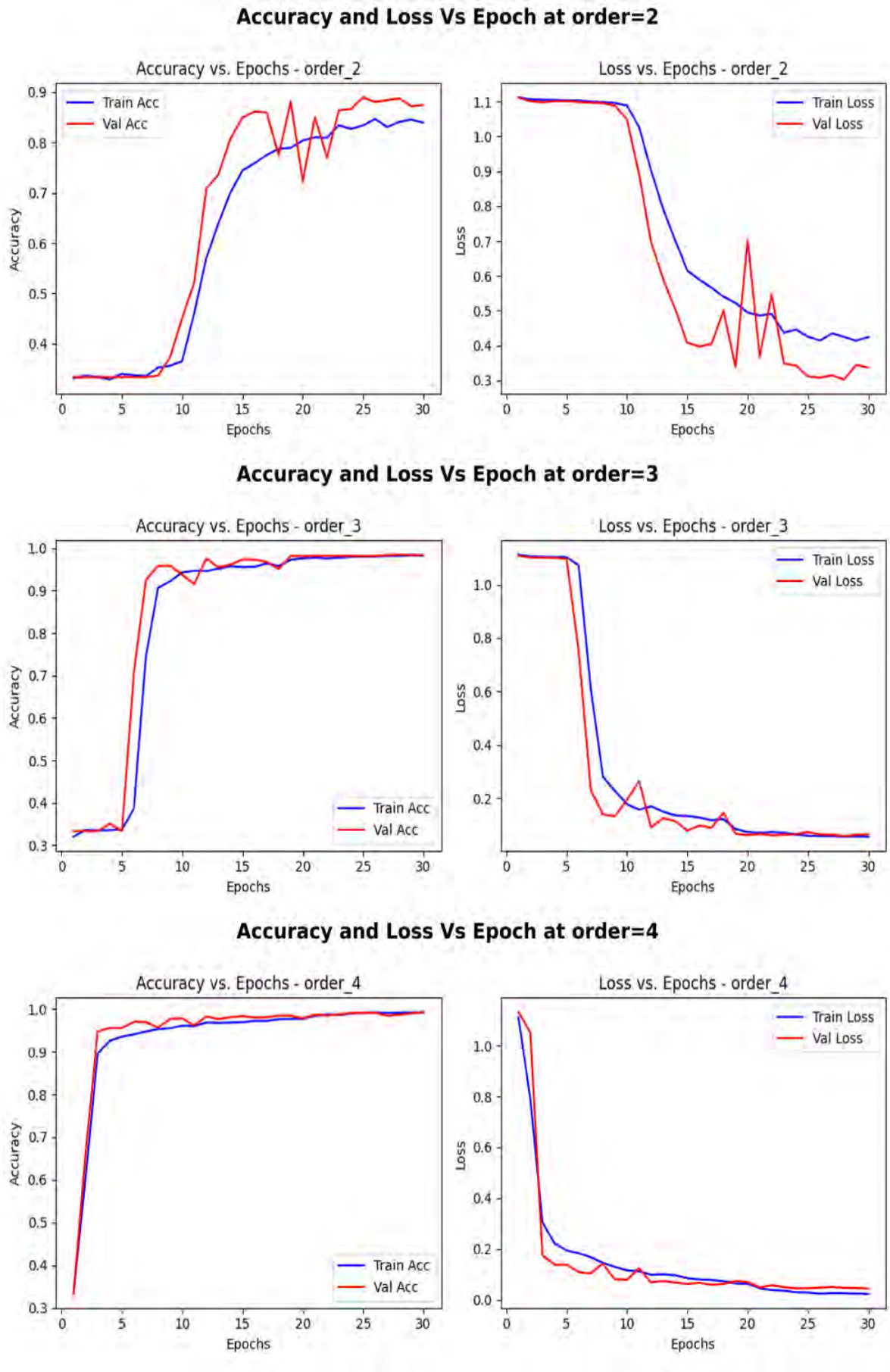


Figure 37. Accuracy and Loss Graphs for DWPCA-DXNet Performance Across Different MTF Order Values

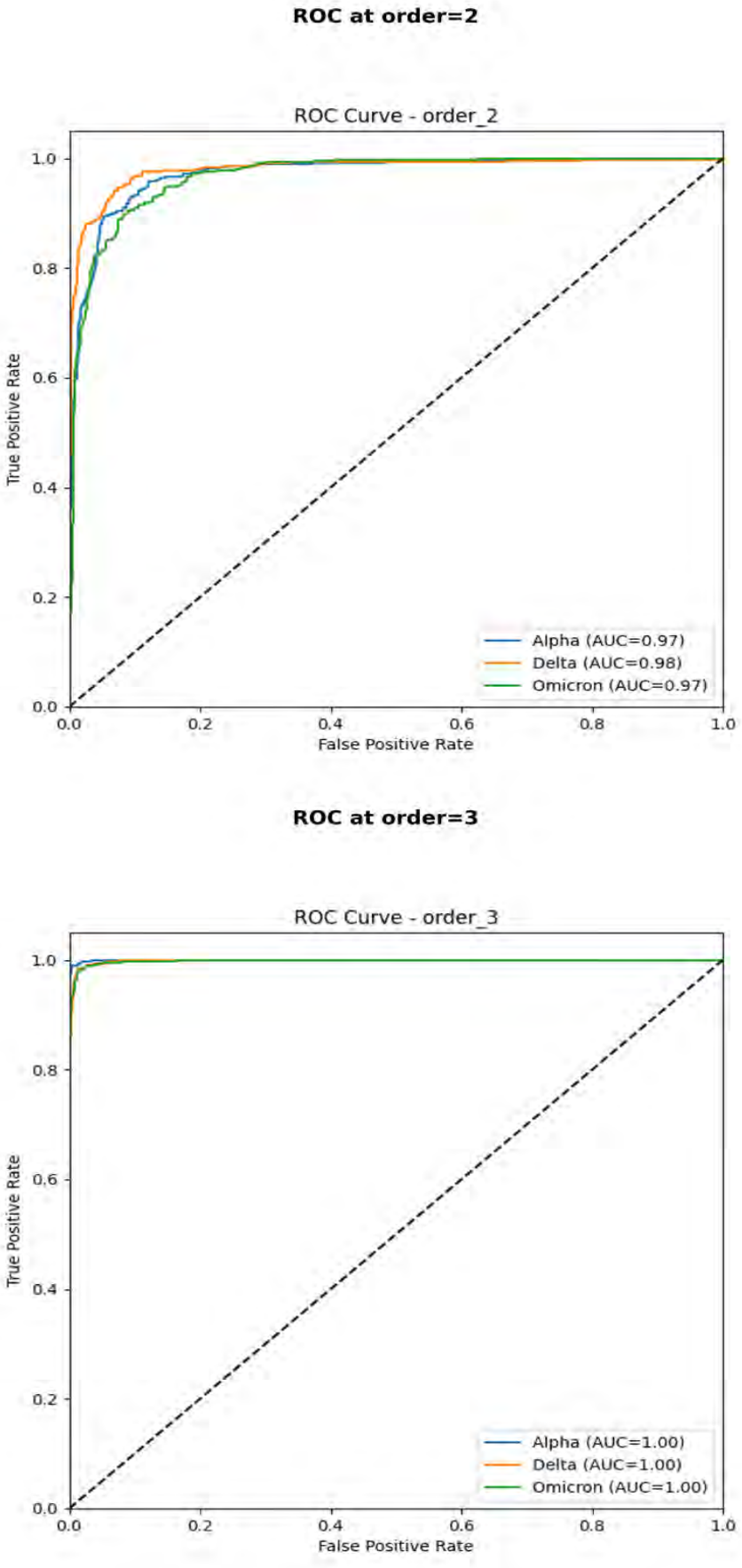


Figure 38. ROC graphs for DWPCA-DXNet Performance Across Different MTF Order Values

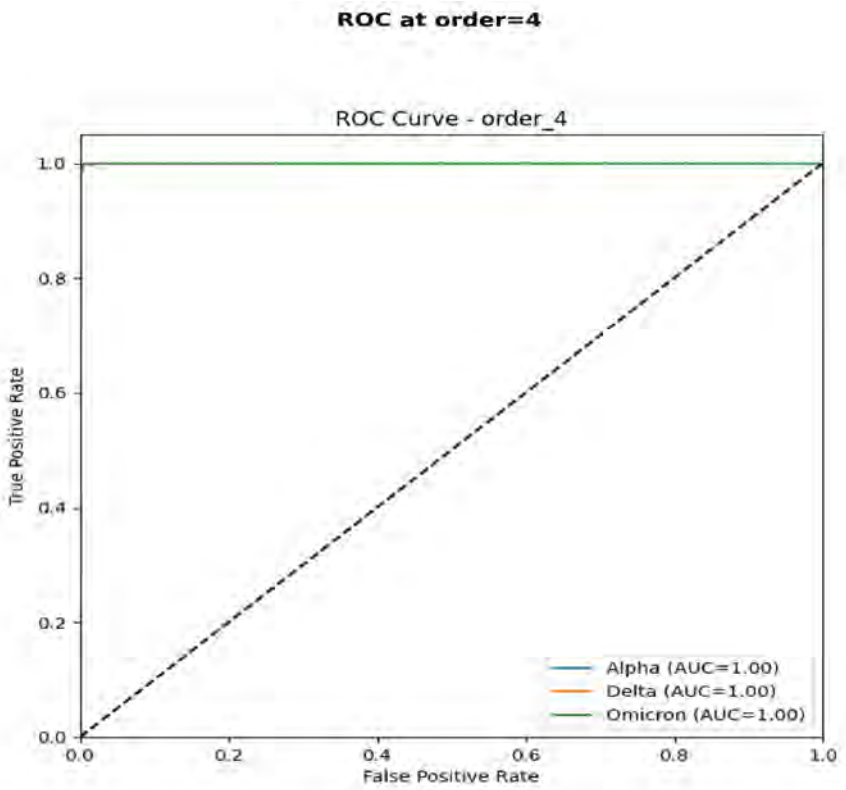


Figure 39. ROC graphs for DWPCA-DXNet Performance Across Different MTF Order Values

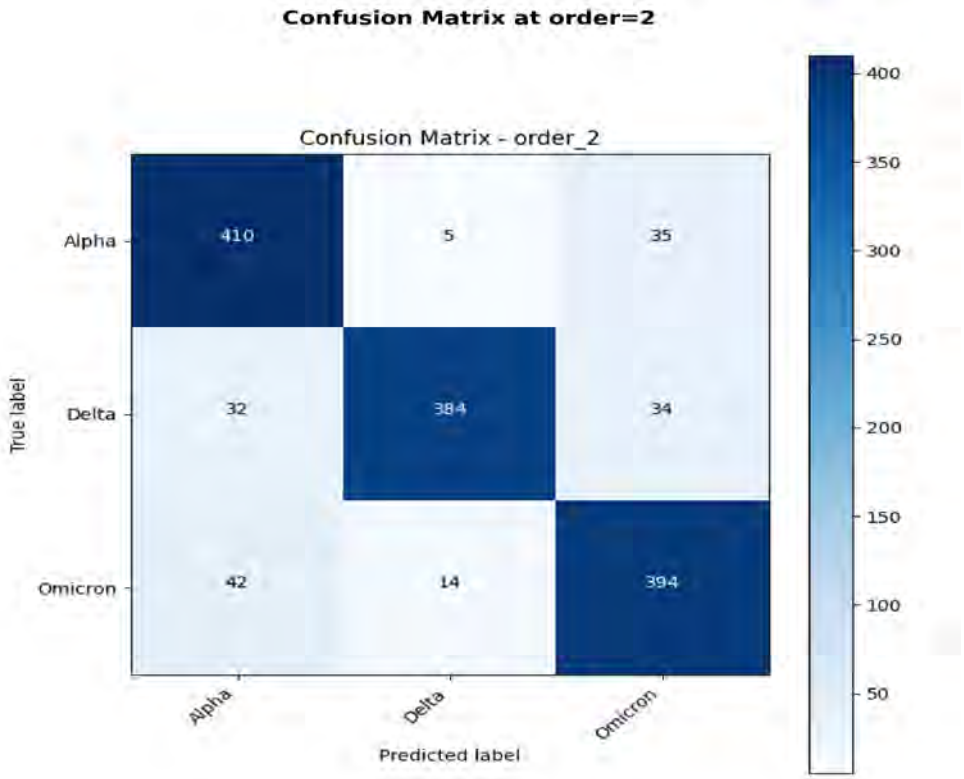


Figure 40. Confusion Matrices for DWPCA-DXNet Performance Across Different MTF Order Values

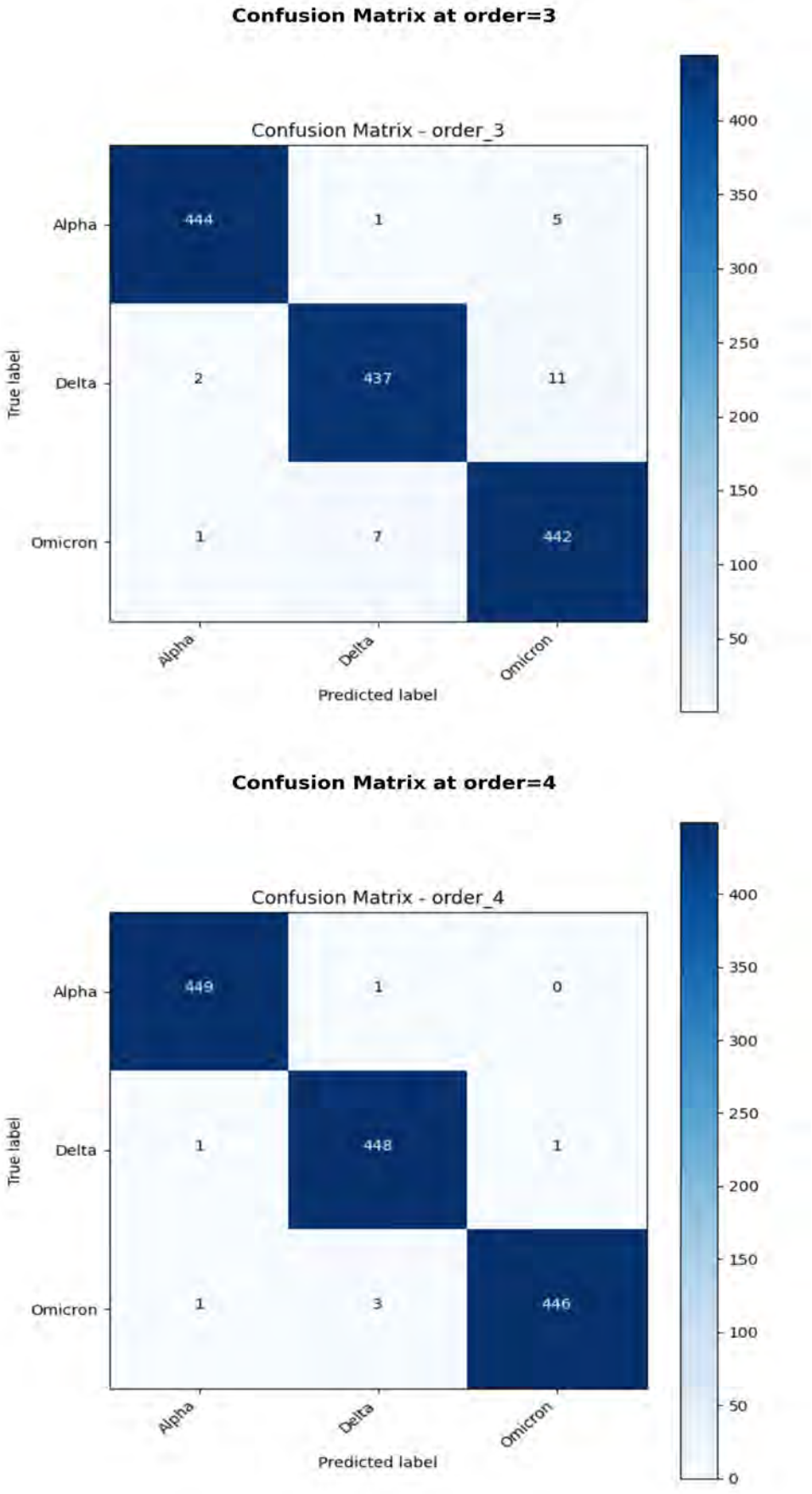


Figure 41. Confusion Matrices for DWPCA-DXNet Performance Across Different MTF Order Values

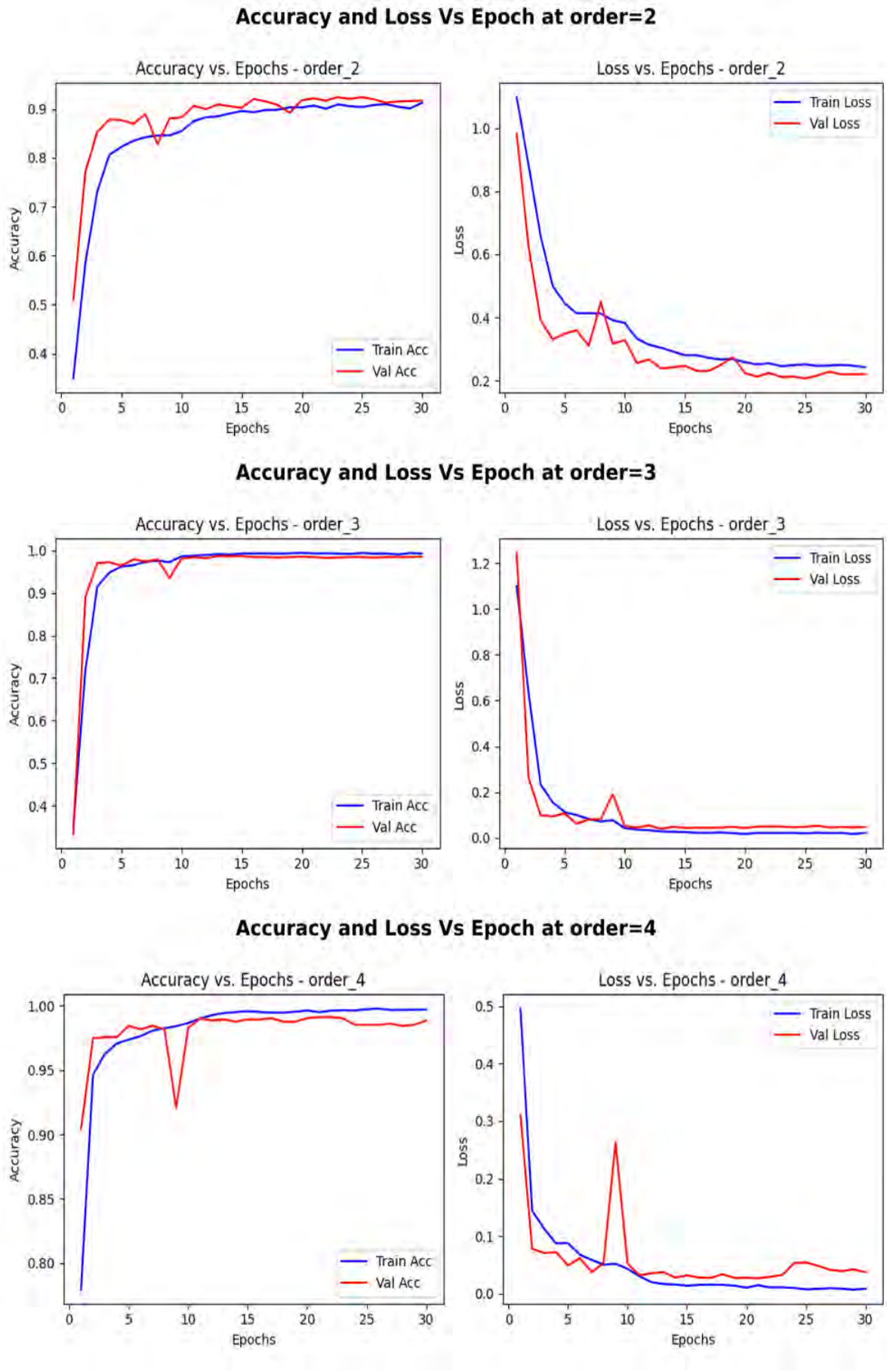


Figure 42. Accuracy and Loss Graphs for DWPCA-XRNet Performance Across Different MTF Order Values

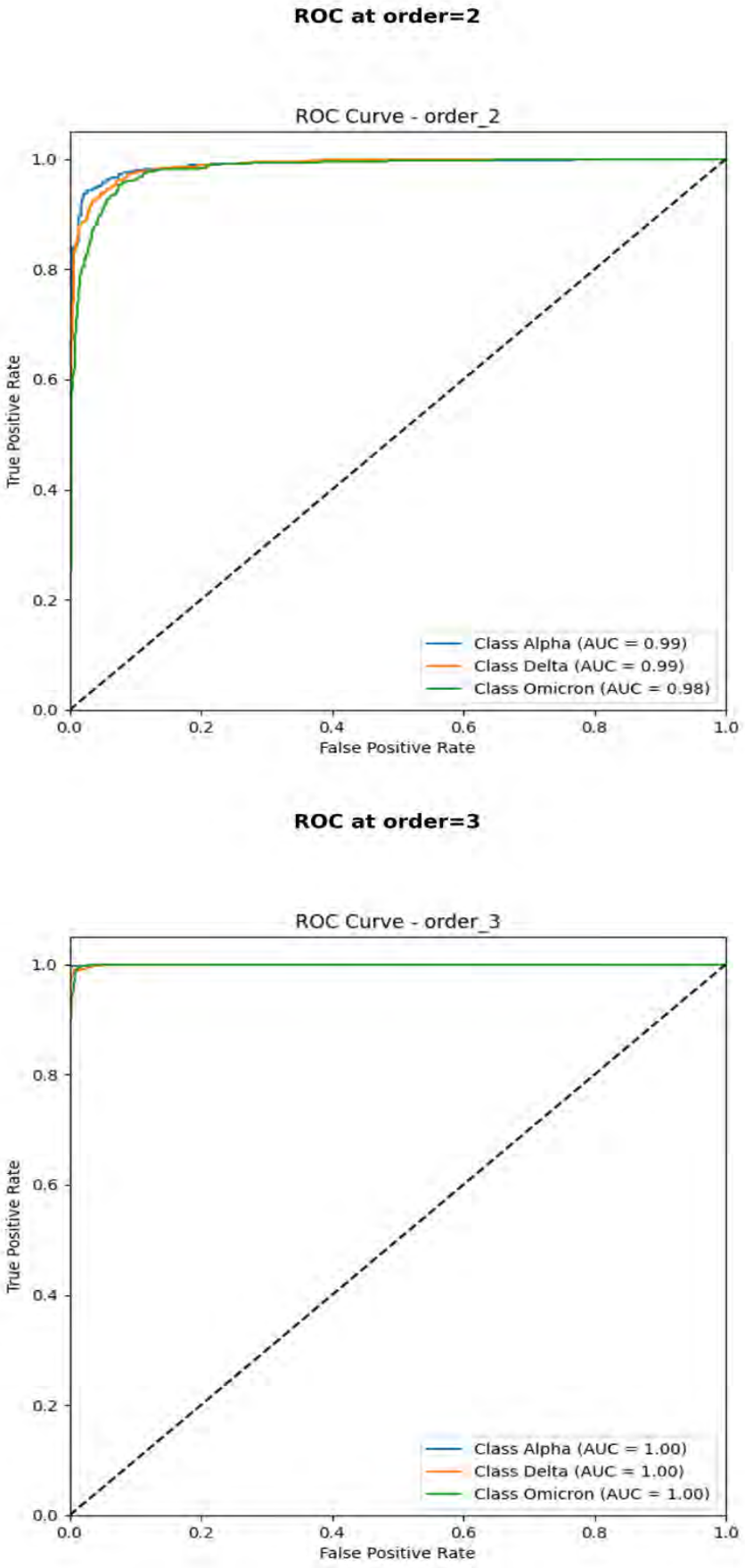


Figure 43. ROC graphs for DWPCA-XRNet Performance Across Different MTF Order Values

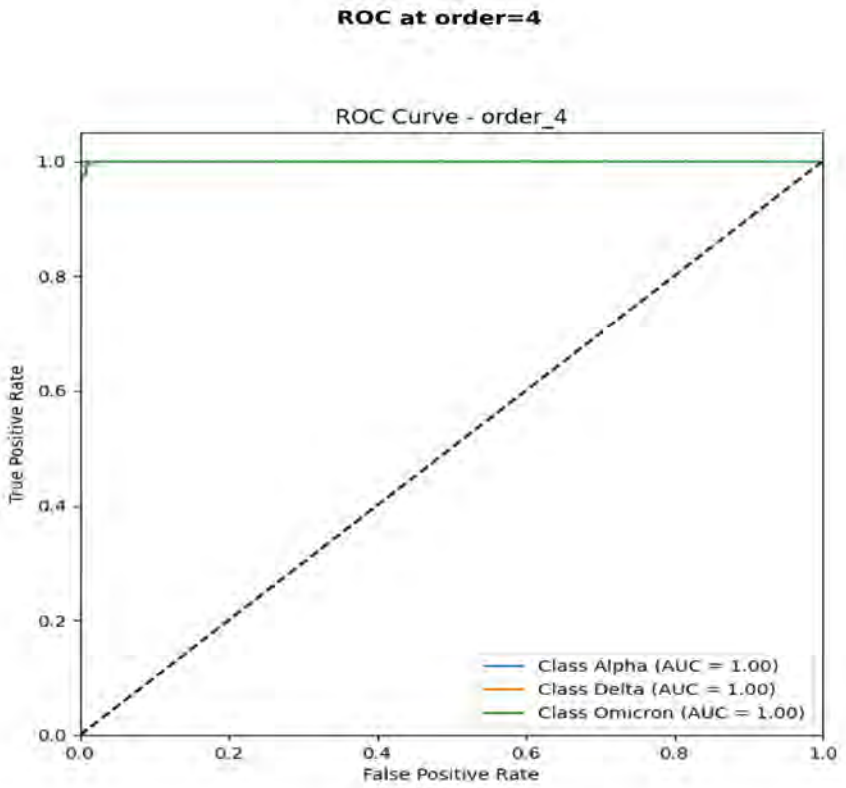


Figure 44. ROC graphs for DWPCA-XRNet Performance Across Different MTF Order Values

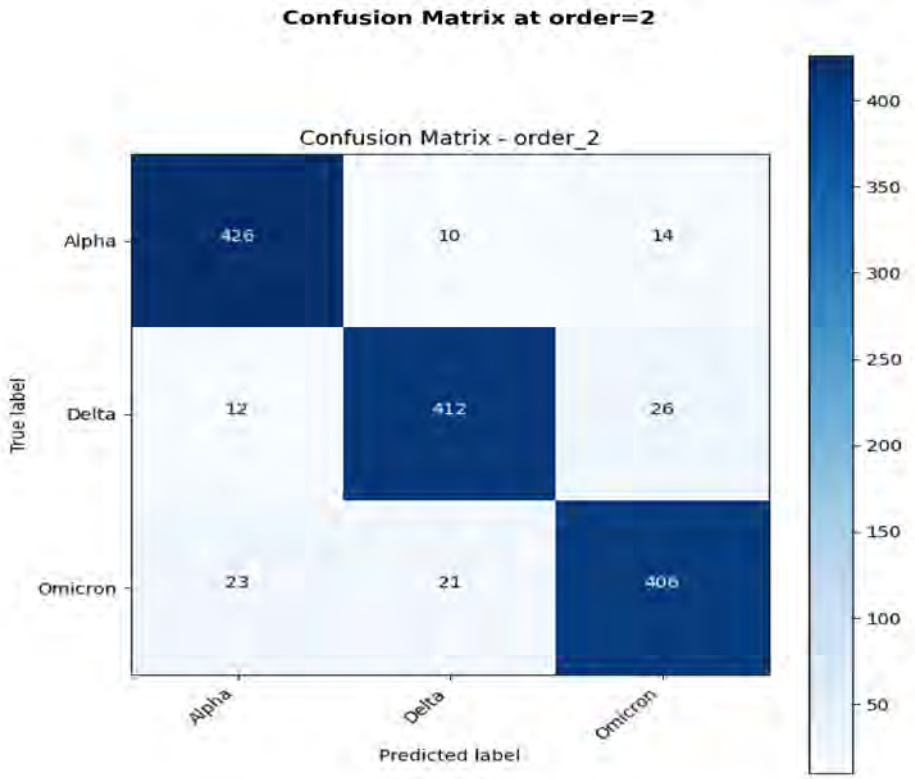


Figure 45. Confusion Matrices for DWPCA-XRNet Performance Across Different MTF Order Values

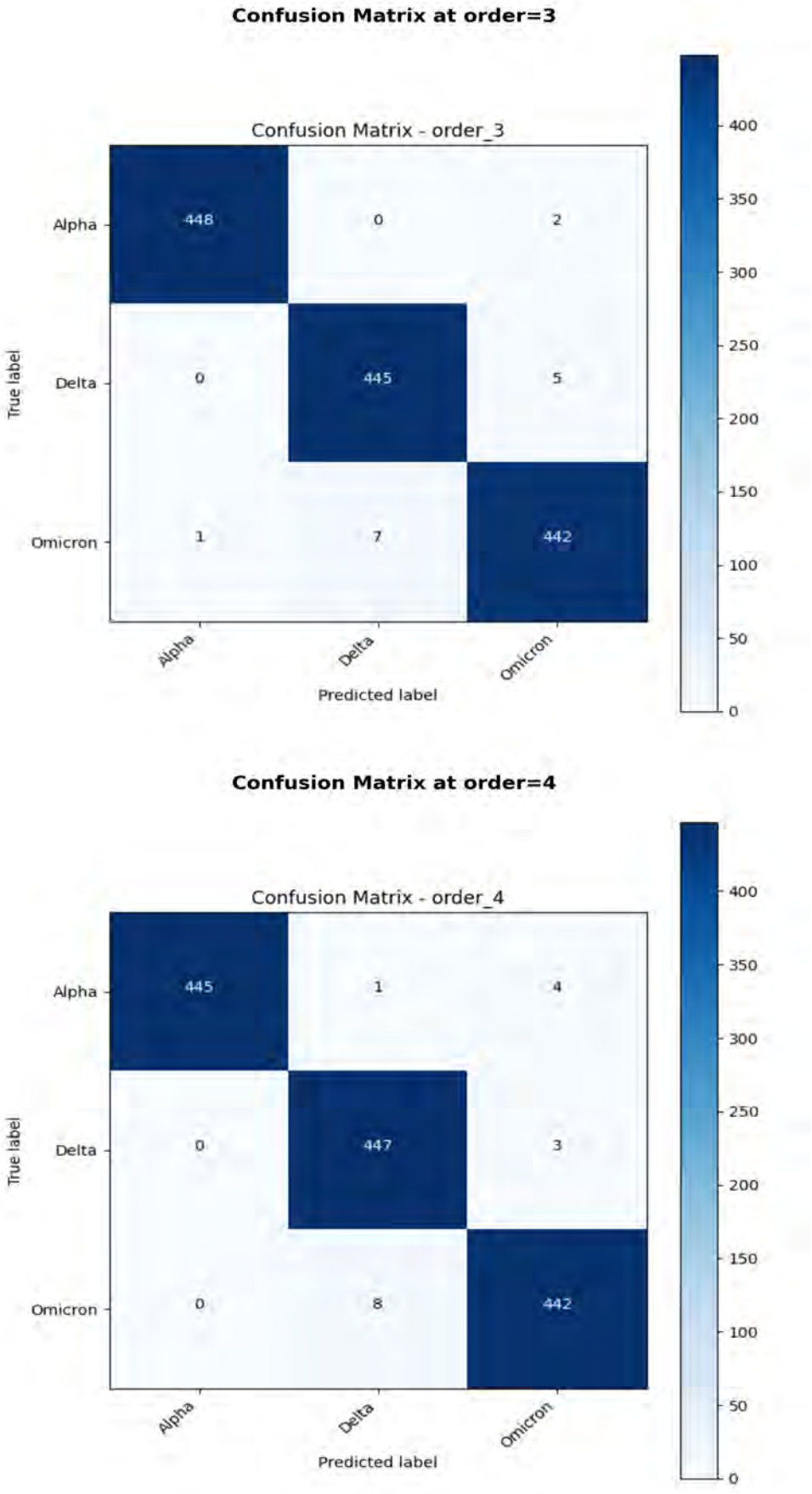


Figure 46. Confusion Matrices for DWPCA-XRNet Performance Across Different MTF Order Values

Table 5. Statistics of Deep Learning Model Performance

Sr No	Model Name	Average Accuracy	Standard Deviation	Worst Accuracy	Best Accuracy
1	DWPCA-RDNet	89.93%	19.69%	42.30%	99.48%
2	DWPCA-DXNet	88.84%	20.36%	39.33%	99.32%
3	DWPCA-XRNet	89.82%	19.84%	44.23%	99.62%

Table 6. Statistics of Genomic Image Processing Techniques

Sr No	GIP Technique	Average Accuracy	Standard Deviation	Worst Accuracy	Best Accuracy
1	FCGR (kmer = 2)	42.62%	2.02%	39.33%	44.66%
2	FCGR (kmer = 3)	94.51%	1.19%	92.59%	95.55%
3	FCGR (kmer = 4)	98.46%	0.33%	98.00%	98.74%
4	FCGR (kmer = 5)	99.47%	0.12%	99.32%	99.62%
5	FCGR (kmer = 6)	99.01%	0.37%	98.59%	99.33%
6	Markov (Order 2)	92.09%	2.32%	88.00%	92.14%
7	Markov (Order 3)	98.44%	0.48%	98.00%	98.88%
8	Markov (Order 4)	99.21%	0.25%	98.81%	99.48%

Table 5 provides statistical insights into the performances of three deep learning models: DWPCA-RDNet, DWPCA-DXNet, and DWPCA-XRNet, based on their accuracy metrics. Accuracy average, standard deviation, worst accuracy, and best accuracy are included to give a comprehensive assessment of each model performance consistency and extremes. Moreover, the DWPCA-XRNet reaches a maximum accuracy of 99.62%, followed by DWPCA-RDNet with 99.48%, and DWPCA-DXNet with 99.32%. Comparatively, the worst accuracies for all models are an expression of their variability in particular conditions; the worst stands at 39.33% for DWPCA-DXNet, whereas the most relatively reliable is DWPCA-XRNet, though poor, at 44.23%. In terms of average accuracy, DWPCA-RDNet and DWPCA- XRNet have very similar performances: 89.93% and 89.82%, respectively; DWPCA-DXNet also follows with 88.84%. The standard deviation values vary between 19.69% and 20.36%, which indicates there is somewhat high variability in performance through different test scenarios, pointing out a moderate level of unreliability in model behavior. In summary, DWPCA-XRNet is the most robust model with the highest best accuracy coupled with a good average performance, while DWPCA-RDNet presents comparable robustness. DWPCA-DXNet shows slightly lower average performance with higher variability. Table 6 shows an overview of performance statistics for genomic image processing techniques, including FCGR and MTF methods, evaluated over a range of configurations. Metrics include average accuracy, standard deviation, worst accuracy, and best accuracy. For FCGR, performance increases monotonically as k-mer values are increased. For example, at k=2, the average accuracy is only 42.62%, with a large variability (standard deviation of 2.02%), but with the best accuracy of 44.66%. As k increases, there is a large improvement in the metrics. At k=4, the average accuracy increases to 98.46%, and the standard deviation reduces to 0.33%. The best performance is obtained at k=5, where an average accuracy of 99.47% and low variability (standard deviation of 0.12%) is observed. For MTF, the performance generally improves with increasing order: at Order 2, the average accuracy is 92.09% with a standard deviation of 2.32%; at Order 4, the models perform near-optimally with an average accuracy of 99.21%, a best accuracy of 99.48%, and low variability (standard deviation of 0.25%). In general, higher configurations perform better with both methods. The highest accuracy and consistency are achieved by FCGR at k=5, which is very closely followed by Markov at Order 4. At the lower configurations FCGR at k=2 and MTF at Order 2 the accuracy is much lower, and variability much higher; this would suggest that these settings are less reliable.

5. Limitation and Future Work

One limitation of our approach as of now is that data currently comes from North America. Despite the fact that the data came from a variety of geographical backgrounds, we think that having data from around the globe will help us test and validate our models on a larger dataset. We have concentrated on building a strong categorization system with the data at hand in this part of our study. The rigorous validation tests to come next will involve applying our model to brand-new, untested datasets gathered from various geographic locations and therapeutic situations. This will assist us in evaluating the model practical effectiveness and ability to adjust to novel viral strains. Moreover, It is plan to implement continual learning mechanisms that allow the model to adapt to new data and variants without needing to be retrained from scratch, enhancing its responsiveness to evolving virus strains. Additionally, another future work is to enhance model to not only handle existing COVID-19 strains but also predict and analyze new virus strains and their variants. Lastly, goal is to develop a comprehensive, all-in-one technique for virus and variant detection, positioning our system at the forefront of DNA sequence classification.

6. Conclusion

A novel and efficient method for SARS-CoV-2 genomic sequence classification was proposed based on the FCGR and MTF approaches. The proposed DWPCA block showed a better ability in feature fusion with outstanding accuracy in classification. In general, performance tends to improve with the increase in complexity for example, higher values of k-mer in FCGR and higher orders in MTF. The performances reached nearly perfect at the optimal configurations for example, k-mer=4 or 5 and Order=3 or 4 concerning richer feature representations. After some threshold was reached, performance increases then flattened out in many cases, indicating diminishing returns but showing the importance of striking a balance between complexity and computational cost. This paper demonstrates the synergy of state-of-the-art GIP techniques with the DWPCA block to provide a scalable framework for genomic classification that will pave new ways in bioinformatics research.

Acknowledgments

We greatly appreciate the valuable contributions of our community advisory committee members. We would also like to thank the University of Windsor and Lawerence Technological University and every team member who took the time to participate in this study.

Author's Contribution

Vatsal Shah: Conceptualization, Methodology, Drafting, Validation, Data curation. **Love Fadia:** Conceptualization, Methodology, Drafting, Validation, Visualization. **Mohammad Hassanzadeh:** Investigation resources, Supervision, Writing-review, Methodology. **Jonathan Wu:** Supervision, Writing-review, Methodology. **Majid Ahmadi:** Supervision, Writing-review, Methodology. **George Pappas:** Supervision, Methodology.

Funding Not Applicable

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Informed consent

Obtained.

Ethics approval The Publication Ethics Committee of the Canadian Center of Science and Education. The journal's policies adhere to the Core Practices established by the Committee on Publication Ethics (COPE).

Provenance and peer review

Not commissioned; externally double-blind peer reviewed.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

Data sharing statement

Data will be made available upon request.

References

Abir, F. Fuad, et al. (2023). PCovNet+: A CNN-VAE anomaly detection framework with LSTM embeddings for smartwatch-based COVID-19 detection. *Engineering Applications of Artificial Intelligence, 122,* 106130. https://doi.org/10.1016/j.engappai.2023.106130

Adjei-Mensah, I., et al. (2024). Cov-Fed: Federated learning-based framework for COVID-19 diagnosis using chest X-ray scans. *Engineering Applications of Artificial Intelligence, 128,* 107448. https://doi.org/10.1016/j.engappai.2023.107448

Ali, S., et al. (2022). Spike2Signal: Classifying Coronavirus Spike Sequences with Deep Learning. In Proc. IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService) (pp. 81-88). https://doi.org/10.1109/BigDataService55688.2022.00020

Amin, H., et al. (2022). *End-to-End Deep Learning Model for Corn Leaf Disease Classification.* IEEE Access. https://doi.org/10.1109/ACCESS.2022.3159678

Andre, M., et al. (2023). From Alpha to Omicron: How Different Variants of Concern of the SARS-Coronavirus-2 Impacted the World. *Biology (Basel), 12*(9), 1267. https://doi.org/10.3390/biology12091267

Anitas, E. M. (2022). Fractal analysis of DNA sequences using frequency chaos game representation and Small-Angle scattering. *International Journal of Molecular Sciences, 23*(3), 1847. https://doi.org/10.3390/ijms23031847

Basu, S., & Campbell, R. H. (2022). *Classifying COVID-19 variants based on genetic sequences using deep learning models.* In Springer Series in Reliability Engineering (pp. 347-360). https://doi.org/10.1007/978-3-031-02063-6_19

Boruah, M., & Das, R. (2024). MLCapsNet+: A multi-capsule network for the identification of the HIV ISs along important sequence positions. *Image and Vision Computing, 145,* 104990. https://doi.org/10.1016/j.imavis.2024.104990

Bray, N., et al. (2024). RT-PCR genotyping assays to identify SARS-CoV-2 variants in England in 2021: a design and retrospective evaluation study. *The Lancet Digital Health, 5*(2), e173-e180. https://doi.org/10.1016/S2666-5247(23)00320-8

Centers for Disease Control and Prevention. (2024). SARS-CoV-2 Variant Classifications and Definitions. Retrieved July 17, 2024, from https://www.cdc.gov/coronavirus/2019-ncov/variants/variant-classifications.html

Choi, S. R., & Lee, M. (2023). Transformer Architecture and Attention Mechanisms in Genome Data Analysis: A Comprehensive Review. *Biology (Basel), 12*(7), 1033. https://doi.org/10.3390/biology12071033

Cinar, N., Ozcan, A., & Kaya, M. (July 2022). A hybrid DenseNet121-UNet model for brain tumor segmentation from MR Images. *Biomedical Signal Processing and Control, 76,* 103647. https://doi.org/10.1016/j.bspc.2022.103647

D., S., & Kalpana, R. (2023). Prognosticating various acute covid lung disorders from COVID-19 patient using chest CT Images. *Engineering Applications of Artificial Intelligence, 119,* 105820. https://doi.org/10.1016/j.engappai.2023.105820

Deng, J., et al. (2009). *ImageNet: A large-scale hierarchical image database.* In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248-255). Miami, FL, USA. https://doi.org/10.1109/CVPR.2009.5206848

Dip, S. D., Sarkar, S. L., Setu, M. A. A., et al. (2023). Evaluation of RT-PCR assays for detection of SARS-CoV-2 variants of concern. *Scientific Reports, 13,* 2342. https://doi.org/10.1038/s41598-023-28275-y

Emrani, J., et al. (2021). SARS-COV-2, infection, transmission, transcription, translation, proteins, and treatment: A review. *International Journal of Biological Macromolecules, 193,* 1249-1273. https://doi.org/10.1016/j.ijbiomac.2021.10.172

Farag, H., et al. (Aug. 2021). Hyperparameters optimization for ResNet and Xception in the purpose of diagnosing COVID-19. *Journal of Intelligent & Fuzzy Systems, 41*(2). https://doi.org/10.3233/JIFS-210925

Gomes, J. C., et al. (2021). Covid-19 diagnosis by combining RT-PCR and pseudo-convolutional machines to characterize virus sequences. *Scientific Reports, 11*(1), 11545. https://doi.org/10.1038/s41598-021-90766-7

Habib, G., & Qureshi, S. (2022). GAPCNN with HyPar: Global Average Pooling convolutional neural network with novel NNLU activation function and HYBRID parallelism. *Frontiers in Computational Neuroscience, 16,* 1004988. https://doi.org/10.3389/fncom.2022.1004988

Hammad, M. S., et al. (2023). A hybrid deep learning approach for COVID-19 detection based on genomic image processing techniques. *Scientific Reports, 13*(1). https://doi.org/10.1038/s41598-023-30941-0

Hammad, M. S., et al. (2023). Genomic image representation of human coronavirus sequences for COVID-19 detection. *Alexandria Engineering Journal, 63,* 583-597. https://doi.org/10.1016/j.aej.2022.08.023

Hammad, M. S., Ghoneim, V., & Mabrouk, M. (2021). *Detection of COVID-19 Using Genomic Image Processing Techniques.* In 2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES). IEEE. https://doi.org/10.1109/NILES53778.2021.9600525

Harikrishnan, N. B., Pranay, S. Y., & Nagaraj, N. (2022). Classification of SARS-CoV-2 viral genome sequences using Neurochaos Learning. *Medical & Biological Engineering & Computing, 60*(8), 2245-2255. https://doi.org/10.1007/s11517-022-02591-3

Harvey, W. T., et al. (2021). SARS-CoV-2 variants, spike mutations and immune escape. *Nature Reviews Microbiology, 19,* 409-424. https://doi.org/10.1038/s41579-021-00573-0

He, K., et al. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770-778). https://doi.org/10.1109/CVPR.2016.90

He, S., et al. (2023). Nucleic Transformer: Classifying DNA Sequences with Self-Attention and Convolutions. *ACS Synthetic Biology, 12*(11), 3205-3214. https://doi.org/10.1021/acssynbio.3c00154

Hodgson, S., Harrison, R. F., & Cross, S. S. (2006). An automated pattern recognition system for the quantification of inflammatory cells in hepatitis-C-infected liver biopsies. *Image and Vision Computing, 24*(9), 1025-1038. https://doi.org/10.1016/j.imavis.2006.02.019

Li, M., et al. (2022). COVID-19 vaccine development: milestones, lessons and prospects. *Signal Transduction and Targeted Therapy, 7*(1). https://doi.org/10.1038/s41392-022-00996-y

Li, X., et al. (2020). Classification of breast cancer histopathological images using interleaved DenseNet with SENet (IDSNet). *PLOS ONE.* https://doi.org/10.1371/journal.pone.0232127

Lin, T., et al. (2022). A survey of transformers. AI Open, 3, 111-132. https://doi.org/10.1016/j.aiopen.2022.10.001

Liyanarachchi, R., et al. (2023). COVID-19 symptom identification using Deep Learning and hardware emulated systems. *Engineering Applications of Artificial Intelligence, 125,* 106709. https://doi.org/10.1016/j.engappai.2023.106709

L?chel, H. F., & Heider, D. (2021). Chaos game representation and its applications in bioinformatics. *Computational and Structural Biotechnology Journal, 19,* 6263-6271. https://doi.org/10.1016/j.csbj.2021.11.008

Lopez-Rincon, A., et al. (2021). Classification and specific primer design for accurate detection of SARS-CoV-2 using deep learning. *Scientific Reports, 11*(1), 947. https://doi.org/10.1038/s41598-020-80363-5

Messaoudi, I., Oueslati, A. E., & Lachiri, Z. (Dec. 2014). Wavelet analysis of frequency chaos game signal: a time-frequency signature of the C. elegans DNA. *EURASIP Journal on Bioinformatics and Systems Biology, 2014*(1), 16. https://doi.org/10.1186/s13637-014-0016-z

Munshi, R. M., et al. (2024). A novel approach for breast cancer detection using optimized ensemble learning framework and XAI. *Image and Vision Computing, 142,* 104910. https://doi.org/10.1016/j.imavis.2024.104910

Muralidar, S., et al. (2020). The emergence of COVID-19 as a global pandemic: Understanding the epidemiology, immune response and potential therapeutic targets of SARS-CoV-2. *Biochimie, 179,* 85-100. https://doi.org/10.1016/j.biochi.2020.09.018

Mwanga, M. J., et al. (2023). *Enhanced Deep Convolutional Neural Network for SARS-COV-2 variants classification.* bioRxiv. https://doi.org/10.1101/2023.08.09.552643

Qin, J., et al. (Jan. 2024). Rapid classification of SARS-CoV-2 variant strains using machine learning-based label-free SERS strategy. *Talanta, 267,* 125080. https://doi.org/10.1016/j.talanta.2023.125080

Rizzo, R., et al. (2016). *Classification experiments of DNA sequences by using a deep neural network and chaos game representation.* In Proceedings of the 17th International Conference on Computer Systems and Technologies 2016 (pp. 222-228). https://doi.org/10.1145/2983468.2983489

Ryabko, B., & Usotskaya, N. (2008). *DNA-sequence analysis using Markov chain models.* In Proceedings of the 2008 IEEE Information Theory Workshop (pp. 119-123). https://doi.org/10.1109/ITW.2008.4578634

Shen, J., et al. (Jan. 2024). ICAFusion: Iterative cross-attention guided feature fusion for multispectral object detection. *Pattern Recognition, 145,* 109913. https://doi.org/10.1016/j.patcog.2023.109913

Togrul, M., & Arslan, H. (2022). *Detection of SARS-CoV-2 Main Variants of Concerns using Deep Learning.* In Proc Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1-5). https://doi.org/10.1109/ASYU56188.2022.9925559

Ullah, W., et al. (2022). Multi-Stage Temporal Convolution Network for COVID-19 variant classification. *Diagnostics, 12*(11), 2736. https://doi.org/10.3390/diagnostics12112736

Usama, M., et al. (2019). Deep Learning Based Weighted Feature Fusion Approach for Sentiment Analysis. *IEEE Access, 7,* 140252-140260. https://doi.org/10.1109/ACCESS.2019.2940051

Virus, NCBI. (n.d.). Available: https://www.ncbi.nlm.nih.gov/labs/virus/vssi/

Wang, H., & Gao, J. (2023). SCV Filter: a hybrid deep learning model for SARS-COV-2 variants classification. *Current Bioinformatics, 18.* https://doi.org/10.2174/1574893618666230809121509

Wei, X., et al. (2020). *Multi-Modality Cross Attention Network for Image and Sentence Matching.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Computer Vision Foundation. University of Science and Technology of China; Kuaishou Technology: IEEE. https://doi.org/10.1109/CVPR42600.2020.01095

Wu, G., & Duan, J. (2022). BLCov: A novel collaborative-competitive broad learning system for COVID-19 detection

from radiology images. *Engineering Applications of Artificial Intelligence, 115,* 105323. https://doi.org/10.1016/j.engappai.2022.105323

Wu, H., et al. (n.d.). *Multi-Level Feature Network with Multi-Loss for Person Re-identification.* Beihang University, Beijing, China. Xu, W., Fu, Y. L., & Zhu, D. (Oct. 2023). ResNet and its application to medical image processing: Research progress and challenges. *Computer Methods and Programs in Biomedicine, 240,* 107660. https://doi.org/10.1016/j.cmpb.2023.107660

Yoon, B. J. (Sept. 2009). Hidden Markov Models and their Applications in Biological Sequence Analysis. *Current Genomics, 10*(6), 402-415. https://doi.org/10.2174/138920209789177575

Zakarczemny, M., & Zajecka, M. (2022). Note on DNA analysis and redesigning using Markov chain. *Genes, 13*(3), 554. https://doi.org/10.3390/genes13030554

Zhou, Q., et al. (2022). Transfer Learning of the ResNet-18 and DenseNet-121 Model Used to Diagnose Intracranial Hemorrhage in CT Scanning. *Current Pharmaceutical Design, 28*(4), 287-295. https://doi.org/10.2174/1381612827666211213143357

Zhu, H., et al. (2023). *Dual Cross-Attention Learning for Fine-Grained Visual Categorization and Object Re- Identification*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Computer Vision Foundation. Beijing, China: IEEE. https://doi.org/10.1109/CVPR52688.2022.00465