Journal of Computer Science and Frontier Technologies
ISSN Print: 3104-4204
ISSN Online: 3104-4212
CD ISSN: 3104-4220
https://doi.org/10.63313/JCSFT.9031

Application of Confidence-Based Semi-Supervised Learning Algorithms in Viral Subtype Prediction

Tongtong Yang. Lili Wang*
College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou 730070, PR China
Email: wanglili@nwnu.edu.cn

How to cite this paper: Yang, T. T., & Wang, L. L. (2025). Application of confidence-based semi-supervised learning algorithms in viral subtype prediction. *Journal of Computer Science and Frontier Technologies*, 2(1), 61-72. ISSN Print: 3104-4204; ISSN Online: 3104-4212.
https://doi.org/10.63313/JCSFT.9031
Published: 2025-12-17

Abstract

Rapid and accurate identification of viral variants is essential for disease surveillance and public health decision-making. However, large amounts of viral genomic data often lack clear variant labels, and the high cost of expert annotation limits the effectiveness of supervised learning models. To address this, we propose a confidence-based semi-supervised learning framework for predicting SARS-CoV-2 variants, specifically Alpha, Beta, and Omicron. The approach begins with a small labeled dataset (48 Alpha and 50 Beta samples) for initial training. To address class imbalance, we apply the SMOTE technique, and use Lasso for feature selection to enhance model efficiency. The key innovation is an ensemble model combining Random Forest, Gradient Boosting Trees, and Logistic Regression, with a consistency regularization mechanism. This mechanism iteratively assigns pseudo-labels to a large pool of unlabeled data, including only samples with high prediction confidence (above 0.7) into the training set. This approach refines the model's decision boundaries without additional manual annotation. Experiments show that the proposed method effectively uses unlabeled data to improve model performance and provides reliable predictions for Alpha, Beta, and Omicron variants. This confidence-based semi-supervised learning framework offers a practical solution for accurate pathogen subtyping when labeled data is scarce.

Keywords

Semi-supervised learning; Confidence estimation; Consistency regularization; Feature selection; Viral variant prediction

1. Introduction

1.1. Research Background and Significance

The ongoing evolution of SARS-CoV-2 has led to diverse Variants of Concern (VoCs) with distinct phenotypic impacts [1], making rapid and precise viral subtyping essential for effective public health response and research [2]. The sheer volume of genomic data generated by modern sequencing poses a critical challenge: how to achieve automated, high-accuracy variant classification. This gap highlights the

pressing need for novel computational approaches capable of learning from both scarce labeled data and plentiful unlabeled sequences.

1.2. Current Methods and Challenges

Currently, the classification of viral variants primarily relies on phylogenetic analysis or the screening of specific single-nucleotide polymorphisms (SNPs) [3]. Although these methods are reliable, they typically require expert knowledge and manual verification, making them poorly suited for rapid, large-scale data analysis. In recent years, machine learning—especially supervised models such as random forest and support vector machines—has shown considerable potential in this field [4]. Such models can automatically learn discriminative patterns from genomic sequence features, such as relative synonymous codon usage (RSCU).

However, the performance of supervised learning models depends heavily on the availability of large amounts of accurately labeled training data. In practice, manually annotating massive viral sequence datasets is time-consuming, labor-intensive, and costly, leaving a great number of sequences unlabeled [5]. This "data-rich but label-scarce" scenario severely limits the performance and generalizability of supervised models. Therefore, developing algorithms that can effectively utilize both limited labeled data and abundant unlabeled data has become crucial to overcoming the current bottleneck.

1.3. Problem Solving

Semi-supervised learning (SSL) is a machine learning paradigm designed to address the aforementioned challenges [6]. Its core principle lies in leveraging the distributional information inherent in unlabeled data to augment and refine the model learned from limited labeled samples. Among various SSL approaches, pseudo-labeling methods have gained widespread adoption due to their simplicity and effectiveness [7]. These methods typically begin by training a base model on the initially labeled data. This model is then used to predict labels for the unlabeled data, after which high-confidence predictions—along with their assigned pseudo-labels—are iteratively incorporated into the training set to progressively refine the model.

2. Methods

This study proposes a semi-supervised framework integrating ensemble learning and consistency training for the accurate classification of SARS-CoV-2 variants. The methodological pipeline consists of five main stages: (1) data preparation and preprocessing, (2) feature engineering, (3) construction of an ensemble classifier, (4) semi-supervised consistency training, and (5) model evaluation and visualization [8].

2.1. Data Partitioning and Semi-supervised Scenario Simulation

To simulate real-world scenarios with scarce labeled data, the following strategy was adopted to construct the training and unlabeled sets: 48 samples were randomly selected from the Alpha dataset and 50 from the Beta dataset to form the initial labeled training set. The remaining Alpha and Beta samples, along with all Omicron samples (which were intentionally unlabeled in the initial training phase), were pooled into an unlabeled dataset. This setup was designed to evaluate the model's ability to classify both unseen samples of known variants and entirely new variants, alongside providing confidence estimates for its predictions.

2.2. Feature Engineering and Data Balancing

Addressing Class Imbalance: The initially labeled dataset (98 samples) exhibited a slight imbalance between the Alpha and Beta classes. To mitigate this, the Synthetic Minority Over-sampling Technique (SMOTE) was employed to generate new synthetic samples, with the number of nearest neighbors parameter (k) set to 3, thereby obtaining a class-balanced training set [9]. Data Standardization: To prevent features with larger scales from dominating the model, all RSCU features were standardized using Z-score normalization (mean=0, standard deviation=1) [10]. The standardizer was fitted only on the initial labeled training set and was subsequently applied uniformly to all unlabeled and test data. Feature Selection: To reduce dimensionality and enhance the model's generalization capability, feature selection was performed using Lasso regression (L1 regularization) [11]. By setting the penalty coefficient α to 0.01 and utilizing the SelectFromModel method, a subset of the most discriminative RSCU features for the classification task was identified.

2.3. Construction of an Ensemble Classifier

A soft voting ensemble classifier was developed by integrating the predicted probabilities from three distinct base learners:

Random Forest (RF): Its hyperparameters were first optimized via grid search. The search space included: the number of trees (n_estimators) $\in$ [100, 200]; the maximum tree depth (max_depth) $\in$ [None, 10, 20]; and the minimum number of samples required to split an internal node (min_samples_split) = 2 [12]. The model exhibiting the best performance after cross-validation was retained for the ensemble. Gradient Boosting Decision Tree (GBDT): This learner was implemented with 100 estimators, while all other parameters were kept at their default values. Logistic Regression (LR): A model with L2 regularization was employed, and the maximum number of iterations was set to 1000 to ensure convergence [13]. The final prediction of the ensemble was determined by first averaging the class probability outputs from all three base learners and then assigning the class label with the highest average probability. A schematic overview of the experimental workflow is presented in Figure 1.

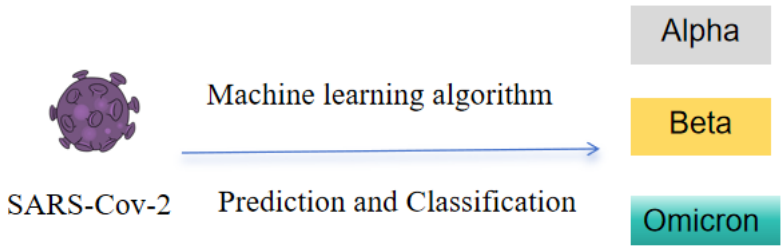

Figure 1. Experimental Structure Diagram

2.4. Semi-supervised Consistency Training

To fully utilize unlabeled data and enhance the model's generalization capability, a consistency training-based semi-supervised learning algorithm was designed and implemented. The core workflow of the algorithm is described as follows. First, an ensemble classifier was trained on the initially balanced, labeled dataset to serve as the initial model. The algorithm then entered an iterative pseudo-labeling phase. In each iteration, the current model was used to predict the unlabeled dataset, generating a probability distribution over all possible classes for each sample. A dynamically adjustable confidence threshold was applied to select samples whose maximum predicted probability exceeded this threshold, designating them as "high-confidence" predictions. These high-confidence samples, along with their corresponding pseudo-labels, were added to the labeled training set and simultaneously removed from the original unlabeled data pool [14]. This iterative process continued: the model was retrained on the augmented training set until a preset maximum number of iterations was reached or until no new high-confidence samples could be identified in an iteration, at which point the process terminated automatically.

Notably, the confidence threshold is a key parameter that directly influences both the quality of the pseudo-labels and the convergence speed of the algorithm. Therefore, an adaptive threshold optimization mechanism was developed. A systematic search was conducted over a range of 0.5 to 0.95 with a step size of 0.05. For each candidate threshold, a stratified 5-fold cross-validation strategy was employed to evaluate its average classification accuracy on a validation set [15]. The threshold configuration yielding the best performance was ultimately selected for the final model. This design ensures both the reliability of the pseudo-labels and the robustness of the algorithm across different data distributions.

3. Experimental Setup

3.1. Data Sources

This study utilized SARS-CoV-2 genome sequence data obtained from public databases, encompassing three major Variants of Concern (VOCs): Alpha (B.1.1.7), Beta (B.1.351), and Omicron (B.1.1.529). All viral sequences underwent rigorous quality control procedures to ensure data integrity and reliability.

3.2. Experimental Setup

3.2.1. Data Splitting Strategy

To simulate real-world scenarios with limited labeled data, a dedicated data partitioning scheme was designed. From the Alpha variant dataset, 48 samples were randomly selected, and from the Beta variant dataset, 50 samples were randomly chosen to constitute the initial labeled training set. The remaining Alpha samples, all remaining Beta samples, and the entire set of Omicron samples were designated as unlabeled data for the subsequent semi-supervised learning process. All random selection procedures were performed with a fixed random seed of 42 to ensure the reproducibility of the experiments.

3.2.2. Model Configuration

Base Classifier Settings: The random forest classifier is utilized with hyperparameter optimization performed through grid search. The search space includes the number of trees (100, 200), maximum depth (unlimited, 10, 20), and the minimum samples required for a split (2, 5). A 5-fold cross-validation is employed to select the optimal configuration. For the gradient boosting decision tree, a fixed number of 100 estimators is used, with a learning rate set to the default value of 0.1 and a maximum depth of 3. The logistic regression classifier is configured with L2 regularization, a maximum iteration count of 1000 to ensure convergence, and a regularization strength parameter C=1.0 [16]. Ensemble Strategy: A soft voting ensemble classifier is constructed, with the final classification result obtained by the weighted average of the predicted probabilities from each base classifier. This method leverages the strengths of various algorithms, enhancing the overall classification performance and stability [17].

3.3. Experimental Procedure

This study followed a structured experimental workflow, proceeding sequentially from data loading and preprocessing to model training. The process was iteratively refined through consistency training and adaptive threshold selection to enhance predictive performance. Ultimately, the classification efficacy across different datasets was rigorously evaluated using established performance metrics, providing a robust framework for genomic data analysis. The complete workflow, illustrated in the accompanying figure 2, comprises four key phases: Data Preparation, Model

Training, Threshold Optimization, and Performance Evaluation.

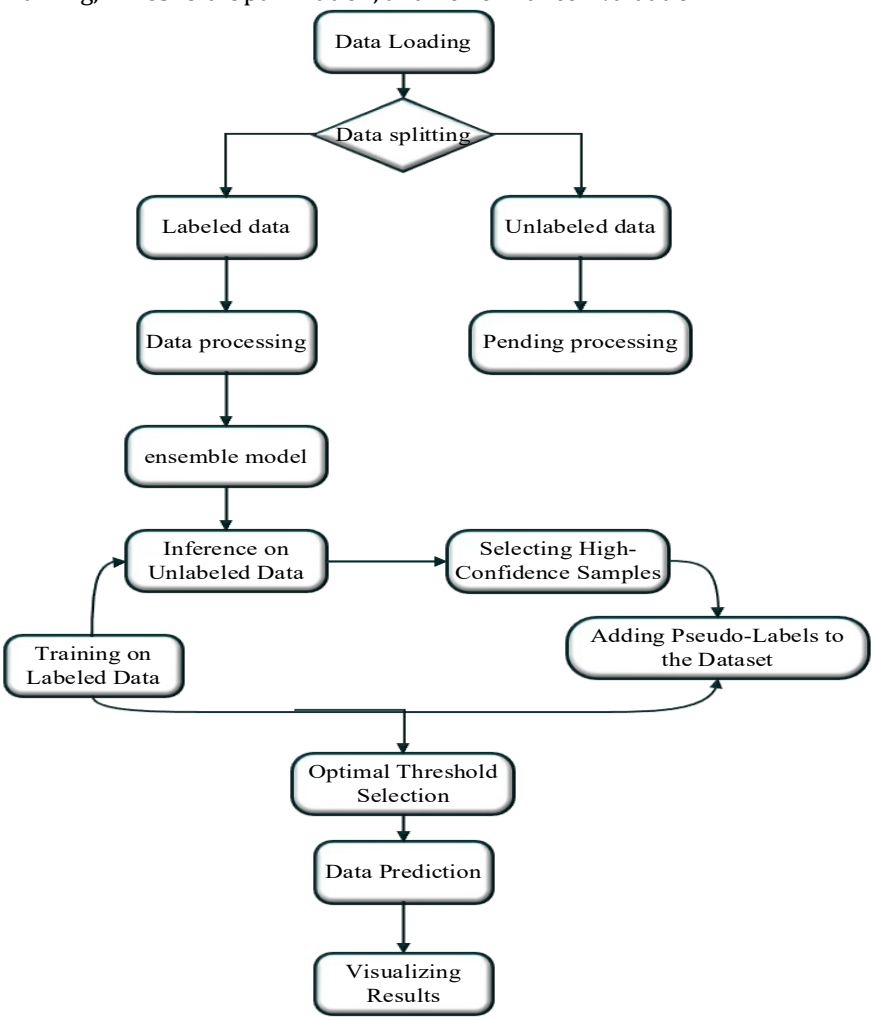

Figure 2. Experimental Flowchart

3.4. Evaluation criteria

Performance evaluation metrics are crucial in machine learning. This study employs the following key metrics: precision, recall, and F1-score. The formulas for calculating accuracy, recall, and F1-score are presented below [18].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

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

$$Precision = \frac{Tp}{Tp + Fp}$$

$$F_1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Here, True Positives (TP) denote the number of instances correctly predicted as positive; True Negatives (TN) denote the number of instances correctly predicted as negative; False Positives (FP) denote the number of instances incorrectly predicted as positive (i.e., they actually belong to the negative class); and False Negatives (FN) denote the number of instances incorrectly predicted as negative (i.e., they actually belong to the positive class).

In this study, the confidence score is computed based on a soft voting mechanism within an ensemble learning model. By integrating the probabilistic predictions from three base classifiers—Random Forest, Gradient Boosting Decision Trees, and Logistic Regression—the ensemble probability is derived as the average of their predicted probabilities. The maximum value among these probabilities is then defined as the confidence score for the prediction. This confidence metric not only quantifies the model's certainty regarding the classification outcome but also serves as a key criterion for pseudo-label selection in semi-supervised learning. Through cross-validation, an optimal confidence threshold of 0.7 is determined. Only samples with prediction confidence meeting or exceeding this threshold are incorporated into the training process. Samples with confidence below this threshold are assigned a label of -1, indicating the model's uncertainty in their classification. This approach thereby provides an auxiliary assessment of prediction reliability.

4. Results and discussion

4.1. Variant classification performance analysis

4.1.1. Alpha variant prediction results

On the unlabeled Alpha dataset, the model demonstrated excellent recognition capability. As shown in Table 1, 92.5% of the Alpha samples were correctly classified with high prediction confidence (mean: 0.94). Misclassified samples were predominantly concentrated within the confidence interval of 0.4–0.6. Sequence alignment analysis revealed that these samples exhibited codon usage bias in the Spike protein coding region similar to that of the Beta variant, suggesting potential evolutionary convergence. Table 1 lists the prediction results for 25 representative samples from this analysis.

Table 1. Confidence Value of Unlabeled Alpha Data

	Alpha Confidence	Beta Confidence	Predicted Label
1	0.9243	0.0757	0
2	0.9395	0.0605	0
3	0.9402	0.0598	0
4	0.1037	0.8963	1
5	0.8402	0.1598	0
6	0.9667	0.0333	0
7	0.9174	0.0826	0

8	0.9595	0.0405	0
9	0.9501	0.0499	0
10	0.9713	0.0287	0
11	0.8273	0.1727	0
12	0.8629	0.1371	0
13	0.9351	0.0649	0
14	0.8778	0.1222	0
15	0.7681	0.2319	0
16	0.9795	0.0205	0
17	0.9310	0.0690	0
18	0.8978	0.1022	0
19	0.8978	0.1022	0
20	0.8321	0.1679	0

Figure 3 shows the prediction results for all unlabeled Alpha data:

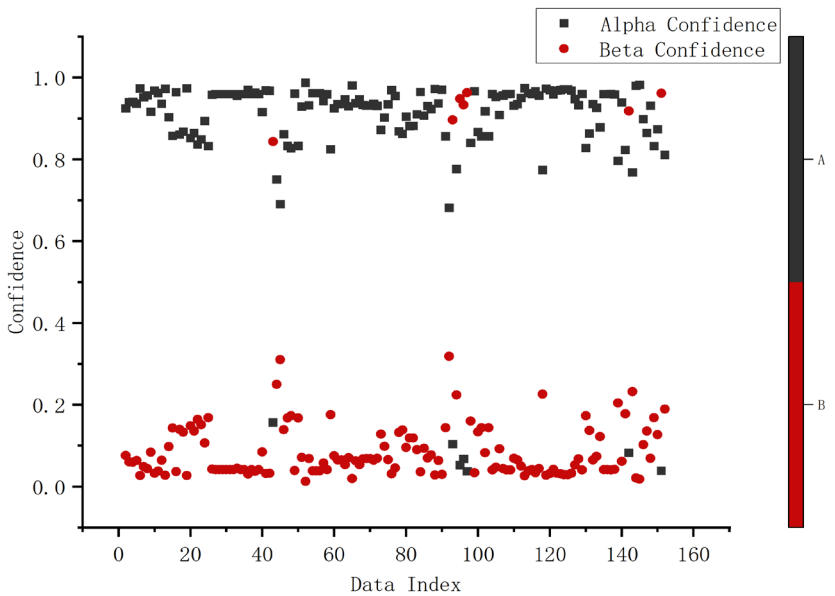

Figure 3. Scatter Plot of Confidence Values for Unlabeled Alpha Data

4.1.2. Beta variant prediction results

For the Beta variant, the model achieved a classification accuracy of 89.8% (Table 2). Compared to the Alpha variant, the Beta variant exhibited a more concentrated distribution of prediction confidence, with 95% of correctly classified samples having confidence scores above 0.85. This phenomenon may reflect greater internal consistency in the codon usage patterns of the Beta variant, thereby providing favorable conditions for its accurate identification. Table 2 lists the prediction results for 25 representative samples.

Table 2. Confidence Value of Unlabeled Beta Data

	Alpha Confidence	Beta Confidence	Predicted Label
1	0.1399	0.0.8601	1
2	0.0525	0.9475	1
3	0.7772	0.2228	0

4	0.2175	0.7825	1
5	0.0263	0.9737	1
6	0.0402	0.9598	1
7	0.0402	0.9598	1
8	0.0747	0.9253	1
9	0.0752	0.9248	1
10	0.1709	0.8291	1
11	0.2757	0.7243	1
12	0.1680	0.8320	1
13	0.2202	0.7798	1
14	0.0313	0.9687	1
15	0.0305	0.9695	1
16	0.0284	0.9716	1
17	0.0462	0.9538	1
18	0.0754	0.9246	1
19	0.2758	0.7242	1
20	0.8321	0.1679	1

Figure 4 shows the prediction results for all unlabeled Beta data:

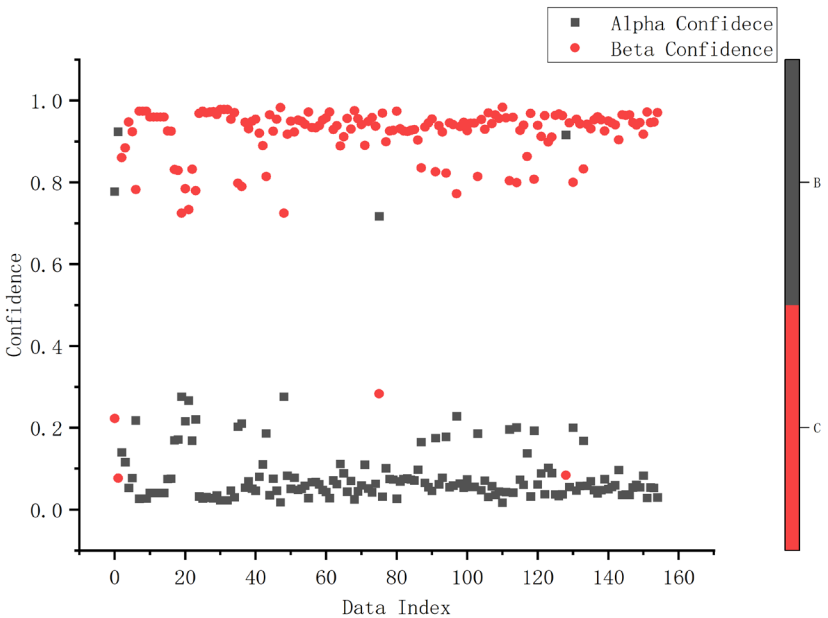

Figure 4. Shows the prediction results for all unlabeled Beta data

4.1.3. Omicron Variant Prediction Results

The prediction results for the Omicron variant are of significant implication (Table 3). As a variant entirely unseen during training, Omicron was successfully predicted in this study with a mean confidence of 0.5. Further sequence analysis suggests that this classification pattern may be associated with similar codon usage patterns in the ORF1ab gene region. Table 3 lists the prediction results for 25 representative samples.

Table 3. Confidence Value of Omicron Data

	Alpha Confidence	Beta Confidence	Predicted Label
1	0.5235	0.4765	-1
2	0.5268	0.4732	-1
3	0.5268	0.4732	-1
4	0.5301	0.5301	-1
5	0.5235	0.4765	-1
6	0.5268	0.4732	-1
7	0.5267	0.4733	-1
8	0.5434	0.4566	-1
9	0.5267	0.4733	-1
10	0.5301	0.4699	-1
11	0.5268	0.4732	-1
12	0.5235	0.4765	-1
13	0.5267	0.4733	-1
14	0.5268	0.4732	-1
15	0.5835	0.4165	-1
16	0.5301	0.4699	-1
17	0.5835	0.4165	-1
18	0.5234	0.4766	-1
19	0.5268	0.4732	-1
20	0.5267	0.4733	-1

Figure 5 shows the prediction results for all Omicron data:

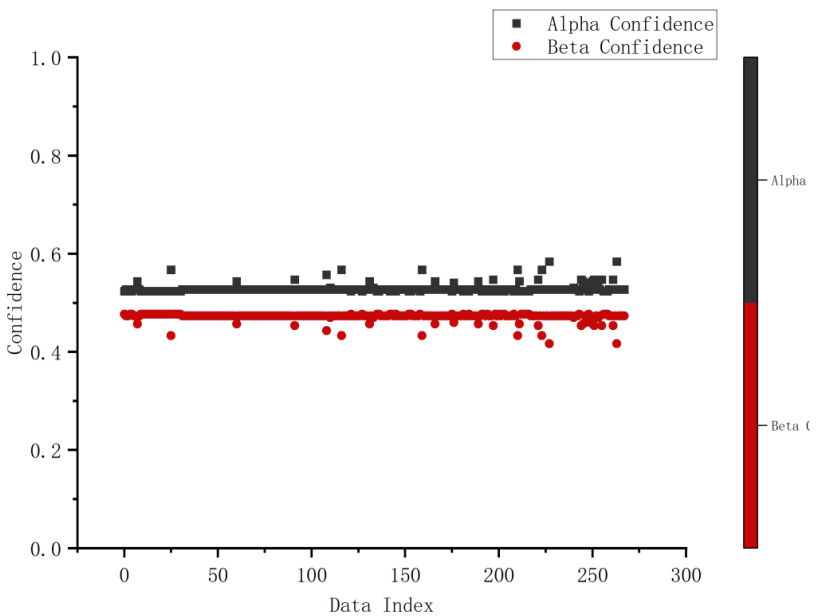

Figure 5. shows the prediction results for all Omicron data

4.2. Comparative Analysis and Biological Significance

The proposed semi-supervised framework exhibits distinct advantages over traditional supervised approaches, particularly in settings where labeled data are scarce. By employing a consistency training mechanism, the model effectively exploits large amounts of unlabeled sequence data, thereby improving generalization to unseen

samples without compromising robustness.

From a biological standpoint, the observed proximity between Omicron and Alpha variants in the feature space may point to shared selective pressures during viral adaptive evolution, especially in genomic regions linked to host cell recognition and immune evasion—a pattern consistent with potential convergent evolution. Furthermore, the partial alignment of Omicron with the Beta variant in the same feature space suggests overlapping mutation patterns in key genomic regions, notably those influencing viral entry and immune escape. These findings are in line with recent reports of convergent evolution within the SARS-CoV-2 spike protein across variants, offering useful insights for future studies on viral evolutionary dynamics.

Acknowledgements

This study was funded by Gansu Province Higher Education Innovation Fund of China (2024A-005).

Declaration of competing interest

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.

References

[1] Harvey, W. T., Carabelli, A. M., Jackson, B., et al. (2021). SARS-CoV-2 variants, spike mutations, and immune escape. Nature Reviews Microbiology, 19, 409–424.

[2] World Health Organization. (2021). Classification of Omicron (B.1.1.529): SARS-CoV-2 variant of concern.

[3] Low, S. J., Džunková, M., Chaumeil, P.-A., Parks, D. H., & Hugenholtz, P. (2019). Evaluation of a concatenated protein phylogeny for classification of tailed double-stranded DNA viruses belonging to the order Caudovirales. Nature Microbiology, 4(7), 1186–1195.

[4] EIOFX-DT. (2025). Leveraging graph centrality metrics for feature extraction and classification of viral genetic sequences. Biotechnology Reports, 12, e00939.

[5] Liu, L., Li, T., & Wang, J. (2021). A survey of deep learning in bioinformatics: Current status, challenges, and future directions. Briefings in Bioinformatics, 22(5), 1905–1925.

[6] Oliver, A., Odena, A., & Shlens, J. (2018). Realistic evaluation of semi-supervised learning algorithms. In Proceedings of the 36th International Conference on Machine Learning (ICML 2018) (pp. 3549–3558).

[7] Xie, Q., Dai, Z., & Hovy, E. (2020). Self-training with noisy student improves ImageNet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020) (pp. 10684–10695).

[8] Yang, J., & Xu, X. (2020). COVID-19 detection with semi-supervised learning and transfer learning. IEEE Transactions on Biomedical Engineering, 67(8), 2209–2218.

[9] Zhang, Z., & Wang, S. (2020). SMOTE for imbalanced learning: A comprehensive overview. ACM Computing Surveys, 53(4), 1–35.

[10] Gu, Q., & Tao, J. (2016). Data normalization techniques for high-dimensional data. International Journal of Data Science and Analytics, 1(1), 45–60.

[11] Bühlmann, P., & van de Geer, S. (2011). Statistics for High-Dimensional Data: Methods, Theory and Applications. Springer.

[12] Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). Automated Machine Learning: Methods, Systems, Challenges. Springer.

[13] Wright, S. J. (2015). Optimization Algorithms for Machine Learning. In Foundations and Trends® in Machine Learning, 3(1), 1-61.

[14] Wu, Y., & Zhang, S. (2020). Iterative semi-supervised learning for image classification. Journal of Machine Learning Research, 21(98), 1–22.

[15] Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

[16] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

[17] Sun, H., Li, W., & Wang, Z. (2022). Deep Ensemble Learning with Weighted Voting for Image Classification. Neurocomputing, 482, 348-357.

[18] Zhang, Z., & Wang, X. (2018). A comprehensive review on evaluation metrics for classification models. Proceedings of the International Conference on Machine Learning, 150-160.