Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Privacy-Preserving Classification of SARS-CoV-2 Spike Sequences via Graph-Based Federated Learning

Mahdi Hosseinpour
University of Shahrood
Esmaeel Tahanian
e.tahanian@shahroodut.ac.ir
University of Shahrood
Mohsen Rezvani
University of Shahrood

Research Article

Keywords:

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10115969/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Privacy-Preserving Classification of SARS-CoV-2 Spike Sequences via Graph-Based Federated Learning

Mahdi Hosseinpour[1], Esmaeel Tahanian[2] and Mohsen Rezvani[3]

Abstract

The continuous mutations of the SARS-CoV-2 virus and the sensitivity of genomic data require the development of collaborative learning methods that ensure data privacy across different institutions. Traditional centralized approaches that require aggregating genomic data from various sources are impractical due to privacy, regulatory, and ethical concerns. Fortunately, Federated Learning (FL) provides a promising solution, enabling distributed model training across different research and medical centers without the need for raw data sharing. By leveraging distributed data analysis, our proposed method ensures that local datasets remain confidential while still contributing to the global model's learning process. In our setup, the input data of clients (participants) partially overlap, reflecting real-world scenarios where research institutions may share subsets of genomic sequences. In this regard, Graph Neural Networks (GNNs) have attracted extensive research attention due to their unique capabilities in modeling complex relationships and graph-based data.

GNNs have been increasingly used in various practical applications, especially in the analysis of genomic data, such as SARS-CoV-2 sequences. Given the rising concerns about data privacy, the use of GNNs in FL environments provides a powerful tool for exploiting relationships between data points and improving model performance in scenarios with limited data from each institution. This has led to the rapid development of the emerging research field of Federated Graph NeuralNetworks. Using the proposed approach, we achieve an overall accuracy of 93.67% in the task of identifying SARS-CoV-2 variants.

Keywords Federated Learning, Graph Neural Networks, SARS-CoV-2 Variant Classification.

1. Introduction

The COVID-19 pandemic, caused by the SARS-CoV-2 virus, has had a profound impact on global public health, leading to widespread infections and fatalities[1]. According to the World Health Organization (WHO)[2], as of July 2022, the virus has resulted in over 6 million deaths and 561 million confirmed infections worldwide[3]. The pandemic has prompted significant research efforts in areas such as containment strategies, medical imaging for diagnostics[4, 5], optimal vaccine distribution[6, 7], and genomic analysis. One of the most critical aspects of understanding and monitoring SARS-CoV-2 is the study of its genetic mutations[8]. Viruses like SARS-CoV-2 continuously evolve under various selection pressures from host immune responses, environmental factors, and pharmaceutical interventions[9].

This adaptability has led to the emergence of multiple variants across different geographical regions, such as the Alpha variant (UK)[10], Beta (South Africa)[11], Gamma (Brazil)[12], Delta (India)[13], and Epsilon (California)[14]. Many of these variants exhibit mutations in the spike protein, increasing transmissibility and binding affinity to the ACE2 receptors in human host cells[15].

The SARS-CoV-2 genome, approximately 30kb in length, consists of several key regions, among which the spike protein (S-protein) plays a fundamental role in viral attachment and host infection[16]. The receptor-binding domain (RBD) within the spike protein facilitates viral entry, making it a hotspot for critical mutations that contribute to variant formation[17]. Studies have demonstrated that analyzing spike sequences alone, rather than the entire genome, is sufficient for variant classification[18-20]. Furthermore, the detailed SARS-CoV-2 genome structure is illustrated in Figure 1[16].

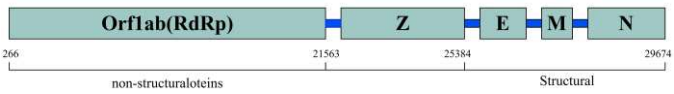

Fig. 1. The SARS-CoV-2 genome, approximately 30kb in length, encodes both structural and non-structural proteins. Among the structural proteins (S, E, M, and N), the spike (S) protein plays a crucial role in viral attachment to the host cell membrane. The spike region consists of 3821 base pairs, which translates into a protein sequence of approximately 1273–1274 amino acids, considering that each amino acid is encoded by a triplet of nucleotides. Mutations in this region often enhance viral transmissibility and immune evasion, making it a key target for genomic analysis[16].

To classify these variants, researchers have employed phylogenetic methods[21], but these approaches often suffer from limited scalability when dealing with large genomic datasets[21]. Given the massive availability of biological sequence data from repositories like GISAID[22], there is an increasing need for scalable and efficient computational models for analyzing SARS-CoV-2 mutations. Recent advancements in machine learning (ML) and deep learning (DL) have facilitated

M. Hosseinpour, E. Tahanian, M. Rezvani are with Department of Artificial Intelligence Engineering, Shahrood University of Technology, Shahrood, Iran. Emails:[1]mahdihoseinpour.se@gmail.com, [2] e.tahanian@shahroodut.ac.ir, [3] mrezvani@shahroodut.ac.ir

more efficient classification and clustering of SARS-CoV-2 variants[23]. These methods typically transform genomic sequences into numerical embeddings before applying vector-space models or kernel-based classifiers for variant identification. As SARS-CoV-2 continues to spread, prolonged exposure to diverse immune responses across different populations and environments accelerates its evolution, leading to the emergence of highly adaptive and immune-evasive variants[24]. Over time, these mutations have largely replaced the original Wuhan strain, reducing vaccine effectiveness[25]. This highlights the urgency of variant characterization across different regions to track transmission patterns and develop more effective containment strategies. Traditional methods for analyzing relationships and detecting mutations in genomic data have limited scalability when dealing with large datasets and struggle to effectively uncover the complex relationships between mutations. To address this issue, we utilized Graph Neural Networks, which perform exceptionally well when handling large-scale data.

In this paper, we present the integration of advanced computational techniques, specifically Graph Neural Networks (GNNs), into variant classification to effectively capture complex mutation relationships and enhance both scalability and accuracy. We further combine GNNs with federated learning to enable decentralized model training across diverse datasets, preserving data privacy while utilizing multi-institutional genomic knowledge for more robust and collaborative variant tracking[26].

The continuous emergence of SARS-CoV-2 variants poses a significant challenge in tracking viral mutations and predicting their impact on transmissibility, immune evasion, and vaccine efficacy. Since mutations in the spike (S) protein play a crucial role in determining viral behavior, an accurate and scalable computational model is needed to classify variants efficiently while maintaining data security and privacy.

In traditional centralized learning approaches, data is aggregated from various sources and transferred to a single location for model training. This approach faces significant challenges, such as privacy, regulatory, and logistical issues, especially when dealing with sensitive healthcare data. Moreover, traditional methods often suffer from biases, as data collection methods may vary across institutions, which reduces the models' ability to generalize to diverse populations. However, in our proposed approach, the input data from clients (different research institutions) partially overlap, reflecting real-world scenarios where institutions may share only subsets of similar genomic sequences. This data overlap allows us to leverage scattered and limited datasets without the need to aggregate all data into a single location, thus maintaining data privacy while enabling the model to learn effectively from decentralized and non-independent data [27].

Unlike traditional FL-based deep learning models[28], which primarily use convolutional or recurrent architectures, GNNs are specifically designed to capture structured relationships within genomic data[29]. By representing spike protein sequences as graph structures, our approach enhances the ability to detect interdependencies between mutations, leading to improved variant classification accuracy.

The main contributions of this work are summarized as follows:

1) We present a novel framework that integrates GNNs with federated learning to classify SARS-CoV-2 variants based on spike protein sequences, enabling decentralized genomic analysis while preserving data privacy.
2) We construct a variant-specific graph structure derived from sequence similarity metrics, allowing the GNN to model complex mutational relationships across different variants of SARS-CoV-2.
3) We analyze the effects of decentralization and graph topology on classification performance, and provide insights into how federated GNNs can be effectively leveraged for large-scale, privacy-aware genomic surveillance.

The rest of this paper is organized as follows. Section 2 reviews related work on SARS-CoV-2 variant classification, covering both traditional machine learning methods and recent advances based on deep learning. Section 3 introduces the proposed approach that integrates GNNs with federated learning to enable decentralized and privacy-preserving classification of viral variants. Section 4 provides implementation details, including data preprocessing, model architecture, and training configuration. Section 5 presents and analyzes the experimental results, demonstrating the performance of the proposed method compared to baseline models across various data settings. Section 6 concludes the paper and discusses directions for future research.

2. RELATED WORK

Machine learning (ML) and deep learning (DL) techniques have been extensively used to enhance the classification and clustering of biological sequences, particularly in the context of SARS-CoV-2 variant analysis[30, 31]. Various methods have been proposed to convert genomic sequences into machine-learning-compatible numerical representations. One of the most common approaches is k-mer-based encoding, which has been widely applied for sequence classification and clustering[18, 32-34]. However, k-mer methods often suffer from storage inefficiencies and computational overhead due to the large number of possible k-mer combinations. position weight matrix (PWM)-based techniques [35] have been introduced, enabling the generation of fixed-length sequence representations by assigning weights to k-mers, thereby preserving the positional significance of sequence fragments.

Beyond traditional feature-engineering-based techniques, federated learning (FL) has emerged as a promising paradigm for privacy-preserving biological sequence classification[36]. FL allows models to be trained collaboratively across decentralized data sources without exposing sensitive information. Several studies have leveraged FL for medical applications, such as COVID-19 symptom detection using wearable devices[37], early breast cancer diagnosis via curriculum learning[38], and neuroimage analysis[39]. In the

context of SARS-CoV-2, FL has been utilized for X-ray-based COVID-19 detection[40]. and clinical prediction of oxygen requirements in symptomatic patients[41]. However, most FL-based studies focus on medical imaging rather than genomic sequence classification[42]. Our work differs by introducing an end-to-end FL-based pipeline tailored for spike protein sequence classification while maintaining data privacy.

Another key challenge in biological sequence analysis is the high dimensionality of raw genomic data. One-hot encoding (OHE) is a straightforward numerical representation[18], but it suffers from sparsity and the curse of dimensionality. Alternative embedding methods, such as k-mer frequency-based representations[34] and PWM-based encodings[35], have been explored to enhance the efficiency of sequence classification. Additionally, kernel-based sequence similarity measures have been proposed, allowing the use of classifiers like Support Vector Machines (SVM) and Random Forest through kernel transformations[43].

Deep learning approaches have also been increasingly adopted to improve biological sequence classification. Some studies have suggested transforming protein sequences into image representations, enabling the use of CNN-based models for feature extraction[44]. In time-series classification, architectures such as ResNet have demonstrated high accuracy[45]. Moreover, techniques like chaos game representation (CGR) and frequency matrix-based CGR (FCGR) have been used to visualize genomic data, offering insights into structural motifs in protein sequences[46].

Despite these advancements, existing methods face limitations in scalability, generalization, and privacy preservation. Our approach integrates federated learning with advanced sequence embeddings to address these challenges, offering a novel and efficient framework for SARS-CoV-2 variant classification.

3. Proposed Approach

This section outlines our FL-based approach for classifying coronavirus variants based on spike protein sequences. We delve into the detailed structure and framework of the proposed model. In recent years, federated learning has gained significant attention as a privacy-preserving learning paradigm, especially in sensitive domains such as healthcare and biomedical research. As the field has evolved, several taxonomies have been established to categorize federated learning based on data structure and feature distribution across participating clients. To better contextualize our proposed framework, we first provide a brief overview of the commonly used classifications in federated learning. Figure 2 summarizes the motivation and conceptual flow guiding the development of the proposed federated GNN-based framework.

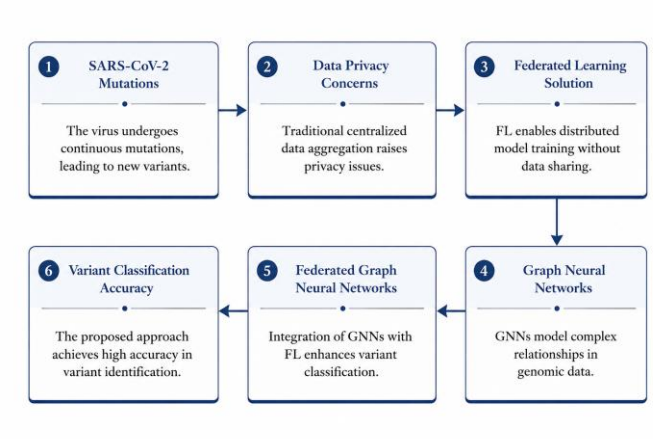

Fig. 2. Overview of the motivation and conceptual workflow leading to the proposed federated learning framework.

Federated learning can generally be categorized into three primary types based on the alignment of feature space and sample distribution across clients:

- Horizontal Federated Learning (HFL)
- Vertical Federated Learning (VFL)
- Federated Transfer Learning (FTL)

In Horizontal Federated Learning, also referred to as sample-based *FL*, participating clients share the same feature space but differ in the samples they possess. For example, multiple hospitals may collect similar types of patient genetic data but from different populations. In this scenario, the structure of features remains consistent, while the available records vary from client to client.

In contrast, Vertical Federated Learning or feature-based FL emerges when organizations share a common set of user identities but store different types of information for each user. For example, a genomic research institution and a clinical center may hold complementary attributes of the same patient cohort. The third category, Federated Transfer Learning (FTL), applies when both the sample space and feature space differ across clients, and only limited overlap exists. This scenario leverages transfer learning techniques to bridge domain gaps, such as when integrating genomic datasets from different regions with different sequencing methodologies.

Given the nature of the dataset and learning strategy employed in this study, our method falls under the Horizontal Federated Learning category. All participating clients process the same type of SARS-CoV-2 spike protein representations (i.e., one-hot encoded amino acid sequences), while only the number and distribution of samples differ across institutions. Additionally, partial overlap between client datasets reflects real-world genomic collaboration scenarios where certain samples may be shared across research facilities while maintaining privacy constraints.

3.1 Graph-Based Model Architecture

The architecture consists of two main components: (1) client models (local) and (2) a federated server model (global). Our approach follows a decentralized data processing strategy,

where the dataset is partitioned into multiple subsets for training, each processed independently by separate clients.

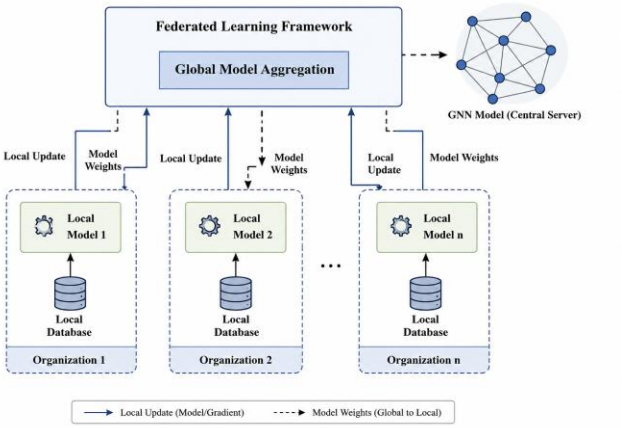

Fig. 3. The federated learning framework utilized for training a model using private datasets from multiple independent organizations, ensuring data privacy and decentralized learning across different institutions.

As illustrated in Figure 3, the federated architecture integrates multiple client models with a centralized global Graph GNN, enabling collaborative learning across organizations while maintaining strict privacy constraints. This design ensures that although data remain isolated at the client level, the system can still benefit from shared learning through model parameter exchanges rather than raw data transfer. After local training is complete, each client outputs a 9-dimensional probability vector representing the likelihood of belonging to different variants. These probability vectors from each client are used as feature vectors for the nodes in the constructed graph. The number of nodes corresponds to the number of clients, with each node receiving its features from the probability vector produced by its corresponding client. The graph, based on the features generated by each client, is then sent to the central server. The central server model, built on GNNs, processes this graph to learn complex dependencies and structural relationships between mutations. This approach, instead of directly sharing model parameters, ensures complete genomic data privacy and enables the detection of hidden dependencies among mutations that traditional linear models might overlook. Integrating graph-based learning within the federated framework significantly enhances variant classification accuracy, as the model leverages not only sequential patterns but also the relational features of mutations. This structure provides a comprehensive view of genomic variations and is particularly crucial for identifying emerging variants. The GNN's graph-based architecture plays a vital role in learning shared features by better understanding the complex interconnections, which allows the system to achieve high generalizability even with data collected from multiple sources with different distributions. Overall, this powerful fusion creates a scalable, secure, and efficient framework for analyzing SARS-CoV-2 variants that not only improves accuracy but also enhances learning efficiency in decentralized environments.

The following sections provide a detailed breakdown of each stage of this approach. Figure 3 illustrates the overall architecture of the proposed federated learning framework. In this design, multiple organizations (clients) independently train local models on their private genomic datasets. These local models periodically send their model outputs as probability vectors (e.g., 9-dimensional vectors representing variant probabilities) to the central federated server. The server uses these probability vectors, which are the outputs of the clients, to construct a graph; specifically, each probability vector (from a single client) is used as the feature vector for its corresponding node in the graph. The information of this constructed graph is then used as the input to the central server model. The central server model, built on GNNs, processes this graph to learn complex dependencies and structural relationships between mutations. This decentralized structure ensures that raw data remains within each organization.

3.1.1 One-Hot Feature Vector Construction

To process biological sequences in machine learning models, it is essential to convert them into numerical feature representations. One of the simplest yet widely used encoding techniques is One-Hot Encoding (OHE)[18], which provides a fixed-length binary representation for biological sequences such as protein sequences (amino acids) and nucleotide sequences (DNA/RNA).

In OHE, each unique character (amino acid or nucleotide) from the biological sequence is mapped to a binary vector, where only one position is marked as 1 while the rest remain 0. The entire sequence is then represented as a concatenation of these individual binary vectors, forming a structured numerical input for classification tasks.

Mathematical Representation**.** Let Σ be the alphabet of possible characters in the sequence. For protein sequences, the alphabet consists of 20 standard amino acids plus an additional placeholder for unknown residues:

$$\Sigma = [A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y]$$

For a sequence i of length l, the OHE feature vector representation Φ_i can be formulated as: $\Phi_i = |\Sigma| \times l$. where $|\Sigma|$ is the total number of unique characters in the sequence alphabet, and l is the sequence length.

Example of OHE for Protein Sequences**.** Consider a short protein sequence: "ACD"

Given the 20-character amino acid alphabet, each amino acid is converted into a 20-dimensional binary vector:

TABLE 1: One-hot encoded amino acid matrix.

Amino acid	A	C	D	...	V	W	Y
A	1	0	0	...	0	0	0
C	0	1	0	...	0	0	0
D	0	0	1	...	0	0	0

3.1.2 Federated Model Implementation

After generating the numerical feature vectors (Φ) for SARS-

CoV-2 spike protein sequences, these vectors are used as input for the Federated Learning (FL)-based model. The dataset is initially divided into two parts: training (Φ_{tr}) and testing (Φ_{ts}). The training dataset (Φ_{tr}) is subsequently split into five distinct subsets, all of which are used for training the client models. Unlike traditional methods where data is evenly distributed among local models, the input data for each client is unequal due to partial overlap between clients. This overlap arises from the graph-based connectivity among clients, where certain spike sequences appear in multiple local datasets. Consequently, while all clients receive data with similar distribution patterns, the volume of data assigned to each client varies, with some clients having larger datasets than others. Furthermore, the federated server model, which is based on GNNs, aggregates the knowledge extracted from local models. By integrating federated learning with graph-based representations, this approach utilizes a graph input that reflects the overlap-based relationships among client data. This, along with leveraging decentralized data distribution and data overlap, is used to extract deeper mutation patterns. As a result, the model achieves higher accuracy in SARS-CoV-2 variant classification by effectively capturing both local and global mutation interactions. This ensures a more adaptive and scalable learning framework for genomic analysis.

3.1.3 Local models

In this approach, five independent client models are initialized, all utilizing the XGB classifier for SARS-CoV-2 variant classification. These models are trained on five distinct subsets of the training dataset ($\Phi_{tr}1$, $\Phi_{tr}2$, $\Phi_{tr}3$, $\Phi_{tr}4$, $\Phi_{tr}5$), with each client processing a different portion of the data. Due to the graph-based connectivity among clients, the amount of input data assigned to each client is not equal. The presence of overlapping data (overlap) across clients results in some clients receiving more training samples than others. However, the overall distribution proportions remain similar, allowing the models to effectively capture shared mutation patterns and structural relationships within the spike protein sequences. During training, each local model learns independently on its assigned dataset, refining its ability to extract relevant mutation features and improving the overall classification performance. During training, each local model learns independently on its assigned dataset, refining its ability to extract relevant mutation features and improving the overall classification performance. After local training, a common evaluation dataset that will be identical for all clients is distributed for updating and refining the client models. The predictions or extracted representations from this shared evaluation data are subsequently forwarded to the central federated server model, enabling global-level evaluation and final decision-making.

3.1.4 Global GraphConv-Based Model Architecture

In this study, the global model deployed on the central server is designed using a graph GNN architecture based on GraphConv [47] Unlike conventional deep learning models that require access to raw data, our federated learning framework strictly adheres to privacy-preserving principles: raw genomic data never leaves the client devices. Instead, each of the five client models, trained independently using XGBoost classifiers on separate portions of the training data, generates a 9-dimensional probability vector for every test sample. These vectors represent the predicted likelihoods of the sample belonging to each of the nine SARS-CoV-2 variant classes. To fully capture inter-node relationships, we construct a fully connected undirected graph, where every node is connected to every other node. This allows the model to exploit relational information across the test samples in addition to individual predictions. The number of nodes in the graph will correspond to the number of evaluation samples. The global model architecture consists of multiple layers designed to effectively process graph-structured data. It begins with an input layer that receives feature vectors representing each node. This is followed by several Graph Convolutional (GraphConv) layers, which perform neighborhood aggregation to update each node's features based on information from its neighbors. These layers are equipped with activation functions, normalization, and dropout techniques to improve training stability and reduce overfitting. Finally, fully connected layers transform the aggregated features into classification outputs representing different variant classes using a suitable activation function. The aggregation operation used in GraphConv is defined as:

$$h_i^{(l+1)} = \sigma\left(W_1.h_i^{(l)} + W_2.AGG\left(\left\{e_{ij}.h_j^{(l)}, j \in \mathcal{N}(i)\right\}\right)\right) \quad (1)$$

where $h_i^{(l)}$ is the feature vector of node i at layer l, $N(i)$ represents the neighboring nodes of node i, e_{ij} is the edge weight between nodes i and j (based on cosine similarity), W_1 and W_2 are learnable weight matrices, AGG is an aggregation function (such as mean or max), and σ is the activation function. The model uses learnable weight matrices and an aggregation function to combine information from neighboring nodes. Each GraphConv layer is supported by normalization and dropout techniques to improve stability and prevent overfitting. Following the graph convolutional layers, fully connected layers transform the features into the final classification output. The model is trained using a suitable loss function and optimized with an efficient optimizer. Dropout is applied after the initial GraphConv layer to further reduce overfitting. By leveraging the graph structure and the relational context among samples, the global model can capture subtle mutation patterns that traditional models might miss. This design ensures that each node benefits from its own local information while also integrating insights from its neighbors through aggregation. Importantly, no raw input data is shared with the global model, maintaining strict adherence to federated learning protocols. This approach provides a robust and privacy-preserving solution for classifying SARS-CoV-2 variants, combining localized learning at clients with powerful global generalization through graph-based methods.

To better illustrate the structure of the centralized global model designed in this study, Figure 4 provides a detailed schematic of the architecture deployed on the federated server. As shown in the figure, the model's input is the constructed graph corresponding to the concatenated probability outputs

from the five client classifiers. Each of these client models produces a 9-element probability vector indicating the likelihood of a test sample belonging to each of the nine SARS-CoV-2 variant classes. In the input layer, this constructed graph is used as the primary input for the Graph Convolutional (GraphConv) layers. Each node in the first GraphConv layer (Conv 1) receives information from the probability outputs and forms an initial graph-based representation for aggregation.

These nodes are then connected through graph edges that facilitate neighborhood aggregation based on the GraphConv mechanism, with edge weights determined based on cosine similarity. Once the graph is constructed and fed into the central model, the training process proceeds iteratively across multiple communication rounds. During each round, the global GNN performs forward propagation using the graph-structured input to learn relational representations among the predicted probability vectors. The model parameters are optimized using supervised feedback based on the true class labels of the evaluation samples available only at the server. This supervised signal enables the GNN to refine inter-node dependencies and detect latent mutation patterns that may not be visible to standalone client models.

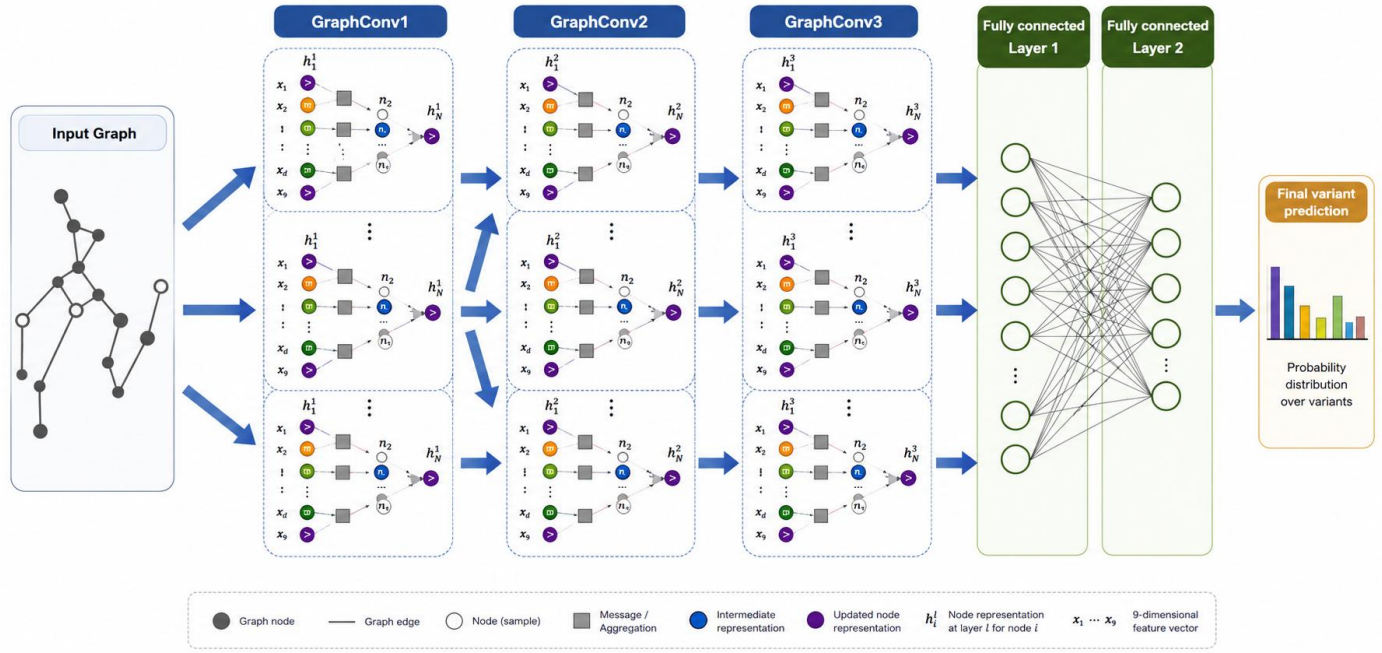

Fig. 4. Architecture of the global model deployed on the central server using GraphConv. The model receives a feature vectrs input vector aggregated from client outputs and processes it through three GraphConv layers followed by fully connected layers to predict the final variant class.

Over successive rounds, the system reaches a stable equilibrium in which prediction consistency across clients improves and the central model converges toward a high-quality representation of genomic mutation dependencies. The integration of the GraphConv architecture within the federated learning workflow therefore enables the framework to move beyond simple parameter aggregation and instead exploit higher-order structural information embedded within client outputs. This mechanism allows the model to capture not only local sequence-level patterns but also inter-sample mutation relationships, ultimately improving the robustness and biological interpretability of SARS-CoV-2 variant classification.

The number of nodes in this layer and the subsequent one is determined based on the desired graph connectivity and the complexity of relational modeling. This architecture allows the model to move beyond isolated predictions by leveraging inter-sample dependencies captured through the graph structure. Consequently, the model can identify complex patterns in mutation behavior while maintaining strict privacy by avoiding access to raw genomic sequences. To clearly demonstrate the flow of operations in our proposed federated learning architecture, Algorithm 1 presents the detailed pseudocode of the method. Starting with the preprocessing step, SARS-CoV-2 spike protein sequences are transformed into numerical vectors using one-hot encoding. The data is then divided into training and testing subsets. The training data is further split across five clients, each of which independently trains an XGBoost classifier on its local subset without sharing any raw data. For each test sample, all clients generate 9-dimensional class probability vectors. These client outputs, which correspond to a single test sample, are treated as node features in a fully connected graph, where each node represents a unique sample. Furthermore, to enhance stability and efficiency, we construct several random graphs with random feature vectors for their nodes. These random feature vectors are generated using the standard deviation of the clients' training or evaluation data. These random graphs are also used to train the central server model. This graph structure is passed to a GraphConv-based global model deployed on the central server, which learns to capture inter-sample relationships and refine classification. The global model is trained using the probability-based node features and their corresponding labels, without direct access to the raw genomic sequences. This entire pipeline ensures both data privacy and enhanced prediction accuracy through relational learning, as formalized in Algorithm 1. This design not only reinforces privacy-preserving principles but also demonstrates the potential of combining ensemble probability

outputs with graph-based learning to improve classification in distributed genomic environments.

Finally, the final output of the central server's graph-based model is compared with the client outputs on the shared evaluation data, and based on this comparison, the client models are updated to improve their local predictions while maintaining privacy constraints. Subsequently, the final accuracy of the entire system is measured using the Test Data.

ALGORITHM 1
PSEUDOCODE FOR FEDERATED VARIANT CLASSIFICATION USING GRAPHCONV

Input:
  Sequence data S
  Ground truth labels y
Output:
  Final predicted variant classes $\hat{V}$
$\Phi \leftarrow OHE(S)$
$(\Phi_{tr}, \Phi_{val}, \Phi_{ts}, y_{tr}, y_{val}, y_{ts}) \leftarrow Split(\Phi, y)$

$>$ *Partition training set into* 5 *clients with partial overlap*
$(\Phi_{tr1}, \Phi_{tr2}, \Phi_{tr3}, \Phi_{tr4}, \Phi_{tr5}) \leftarrow Partition(\Phi_{tr})$
For $i = 1$ *to* 5 *do*
  $model_i \leftarrow Train_{XGBoost}(\Phi_{tri}, y_{tri})$
End For
For each validation sample $x_j \in \Phi_{val}$ *do*
  $\Lambda^j \leftarrow [\,]$
$>$ *empty vector to store outputs of clients*

  For $i = 1$ *to* 5 *do*
$\lambda_i{}^j \leftarrow model_i.predict_{proba}(x_j)$ $>$ 9 − *dimensional vector*
    Append $\lambda_i{}^j$ *to* Λ^j
  End For

  $>$ *Store* Λ^j *as feature for node* j
  $NodeFeature[j] \leftarrow \Lambda^j$
End For
For $r = 1$ *to* R *do*
  $\Phi_r \leftarrow SampleNormal(mean = 0, std = \delta)$
  $RandomGraph[r] \leftarrow BuildGraph(\Phi_r)$
End For
$G \leftarrow BuildGraph(NodeFeature)$
$>$ *Edge weights computed based on overlap or cosine similarity*
$model_g \leftarrow GraphConv()$
$model_g.train(G, labels = y_{val})$
$\hat{V} \leftarrow model_g.predict(G)$
$ComputeAccuracy(\hat{V}, y_{val})$
$>$ *Optionally*: *Compare* $\hat{V}$ *with* Λ^j *to detect mismatches and adjust models*

Return $\hat{V}$

4. EXPERIMENT

In this section, we present a comprehensive evaluation of the proposed federated learning framework for SARS-CoV-2 variant classification based on spike protein sequences. The experiments are designed to assess the performance of both the local client-side XGBoost classifiers and the global GraphConv-based aggregation model operating on the central server. To ensure a fair and systematic analysis, we describe the dataset characteristics, preprocessing procedures, federated data partitioning strategy, and model configurations. We further compare our method against baseline approaches, examine the contribution of graph-structured relational learning, and analyze the effectiveness of random graph augmentation. The experimental results highlight the robustness, accuracy, and privacy-preserving benefits of the proposed architecture in distributed genomic environments.

4.1 Experimental setup

In this section, we describe the spike protein sequence dataset used in our experiments, followed by the configuration of the baseline models and the evaluation metrics employed to assess model performance. Utilizing a federated learning (FL) framework, we develop a solution for SARS-CoV-2 variant classification in which each client independently trains an XGBoost classifier on its local subset without sharing any raw data. All experiments are conducted on a machine equipped with an Intel Core™ i5 processor, 8 GB RAM, and Windows 10 (64-bit) operating system. For each test or validation sample, the five client models produce 9-dimensional probability vectors corresponding to the nine SARS-CoV-2 variant classes. These vectors are concatenated to form the node features used by the global model. The global model, implemented as a GraphConv-based Graph Neural Network (GNN), operates on a fully connected undirected graph whose nodes represent samples and whose edge weights are determined based on the overlap percentage between the client subsets that generated the corresponding node features.

4.2 Dataset Description

Spike protein sequences of the SARS-CoV-2 virus were obtained from the GISAID[1] database, a leading platform that offers open access to genomic data for pathogens such as SARS-CoV-2 and Influenza. Widely used in the bioinformatics community, GISAID supports global efforts in tracking viral evolution and detecting emerging variants in real time. The dataset includes 9,000 spike sequences across nine major variants of the virus, with 1,000 sequences randomly selected for each. These variants are identified based on characteristic mutations in the spike protein region, which serve as the basis for class labeling in the classification task. Table 2 provides detailed statistics, including sequence length distribution, number of mutations per variant, and the region where each variant was first detected, offering a clear overview of the dataset's composition and diversity.

[1] https://www.gisaid.org/

TABLE 2: Summary of the SARS-CoV-2 dataset used in this study.

Lineage	Region of first time	Variant name	No. of sequences
B.1.351	South Africa[11]	Beta	1000
B.1.427	California[14]	Epsilon	1000
B.1.429	California[48]	Epsilon	1000
B.1.525	UK and Nigeria[24]	Eta	1000
B.1.526	New York[49]	Iota	1000
B.1.617.2	India[13]	Delta	1000
B.1.621	Colombia[50]	Mu	1000
C.37	Peru[50]	Lambda	1000
P.1	Brazil[12]	Gamma	1000

4.3 Federated Data Partitioning

In federated learning architectures, the distribution of data across clients plays a critical role in determining the overall performance, convergence behavior, and generalization capability of the global model. This factor becomes particularly important in biological and genomic applications, where datasets are often heterogeneous, unevenly distributed, and collected across different institutions. Introducing controlled overlap between client datasets can facilitate partial alignment among local models, enhance implicit information transfer, and ultimately improve the quality of the global model. However, excessive overlap may reduce diversity and diminish the benefits of decentralized learning.

To systematically analyze the impact of overlap, we design a partitioning strategy in which only the percentage of shared samples between certain client pairs is varied, while all other conditions—including dataset size, model structure, training procedure, and number of repetitions—remain identical. In this study, the training data is divided into five distinct subsets with predefined overlap ratios representing partial data sharing between selected client pairs. The overlap values are defined as follows: (0,1): 0.8, (0,2): 0.6, (1,3): 0.4, (3,4): 0.3, and (2,4): 0.35. This structured partitioning simulates a realistic non-IID federated environment, where institutions may possess partially overlapping genomic datasets while still preserving privacy through decentralized data ownership.

Below are the overlap ratios defined for the two designed scenarios, where only the shared-data percentages differ while all other experimental conditions remain identical. Table 3 summarizes the overlap percentages for each client pair in both the high-overlap and low-overlap scenarios.

TABLE 3: Client Pair Overlap Percentages for the Two Scenarios.

Client Pair	High-Overlap Scenario (%)	Low-Overlap Scenario (%)
Client 1 & Client 2	80%	70%
Client 1 & Client 3	60%	50%
Client 2 & Client 4	40%	40%
Client 4 & Client 5	30%	30%
Client 3 & Client 5	35%	35%

The 9-dimensional probability vectors produced by the local XGBoost models serve as the node features in a fully connected undirected graph. The edge weights in this graph are determined based on the overlap percentages between the client subsets responsible for generating each pair of node features. This graph structure encodes the relational similarity between samples based on shared data sources. Figure 5 illustrates the overlap configuration among clients and highlights the percentage of shared samples across each client pair.

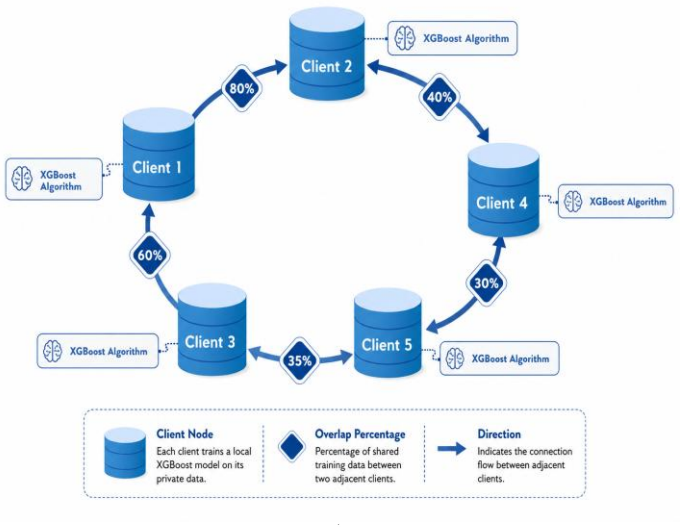

Fig. 5. Overlap Between Clients' Input Data.

These controlled overlap ratios simulate a realistic non-IID data distribution while maintaining privacy by limiting shared information. This configuration ensures that each client holds a partially unique dataset while still retaining meaningful relationships with neighboring clients in the federated network.

4.4 Comparison Methods

To provide a comprehensive comparison for the proposed federated GraphConv framework, several baseline methods from prior literature and classical federated learning architectures are considered. These baselines represent a spectrum of centralized, feature-engineering-based, and federated aggregation approaches commonly used in genomic sequence classification.

PWM2Vec[35] is a handcrafted feature-engineering approach that constructs position-weight matrices (PWMs) to encode spike protein sequences into fixed-length numerical vectors. The method captures positional amino-acid preferences and serves as input for traditional machine-learning classifiers such as Random Forest. Although effective in centralized settings, PWM2Vec relies heavily on manual feature design and is limited in its ability to learn deeper or hierarchical sequence patterns. While PWM2Vec provides interpretable feature vectors and performs reasonably well in centralized settings, it relies heavily on manual feature construction and fixed-length matrices. Consequently, the method cannot adapt to nonlinear or long-range dependencies within protein sequences and lacks the flexibility to learn hierarchical patterns directly from data. Moreover, because it depends on centrally collected sequences, PWM2Vec is not suitable for privacy-preserving or decentralized environments such as federated learning, where raw genomic data cannot be shared across institutions.

Spike2Vec[20] is a scalable sequence-embedding technique that generates k-mers from spike protein sequences and applies Random Fourier Features (RFF) to produce compact numerical representations. These embeddings are designed for downstream classifiers such as Logistic Regression or Naive

Bayes. While computationally lightweight, Spike2Vec primarily models k-mer statistics and does not incorporate relational information or structural dependencies between samples. Although Spike2Vec offers excellent computational efficiency and eliminates the need for sequence alignment, its modeling capacity is inherently limited. The method primarily captures local k-mer statistics and does not account for global context, sequential order beyond the k-mer window, or structural relationships between samples. Additionally, because RFF-based embeddings are not optimized end-to-end with the classifier, they may fail to encode subtle mutation patterns that are critical for distinguishing closely related SARS-CoV-2 variants.

In MLP Server Model, a simple multi-layer perceptron (MLP) serves as the central server model. Each client trains a local classifier independently, and the server processes aggregated client outputs through the MLP to generate final predictions.
FedAvg Aggregation utilizes the classical Federated Averaging strategy, in which the central server aggregates client-side model updates by computing the element-wise average of their parameters. This method does not incorporate any dedicated model architecture and instead depends solely on statistical parameter aggregation across clients. As a minimalistic federated baseline, FedAvg serves as a valuable reference point for assessing the benefits of incorporating structural modeling within the proposed GraphConv-based framework.

4.5 Performance Metrics

To evaluate the performance of the proposed federated GraphConv-based classification framework, several widely used metrics for multi-class classification are employed. Accuracy is used as a global indicator of performance, measuring the proportion of correctly classified samples across all variant classes. However, because accuracy alone may be misleading in the presence of class imbalance, additional class-level metrics are also considered. Precision quantifies the proportion of correctly predicted samples within each predicted class, indicating the model's ability to avoid false positives. Recall (Sensitivity) measures the fraction of actual samples of each class that are correctly identified, reflecting the model's capability to minimize false negatives. To provide a balanced evaluation, the F1-score—the harmonic mean of precision and recall—is reported for each class, offering a single metric that captures both types of classification errors. In addition to these metrics, the Receiver Operating Characteristic–Area Under the Curve (ROC–AUC) is used to assess the model's discriminative ability across all variants. For multi-class classification, a one-vs-rest strategy is applied to compute macro-averaged AUC values, providing insight into how well the model separates each variant from the others. High ROC–AUC values indicate strong separability in the learned feature space, which is particularly important in genomic classification tasks where certain variants may exhibit subtle sequence-level differences. Furthermore, reporting multiple evaluation metrics allows for a more comprehensive understanding of model behavior, especially when dealing with heterogeneous and partially overlapping datasets in federated environments. These metrics collectively enable a robust comparison between client-side models, baseline methods, and the proposed global GraphConv architecture. Such a multi-perspective assessment ensures that improvements in the federated framework are not limited to a single metric but are consistently reflected across all major performance dimensions.

4.6 Results and Discussion

This section presents the experimental findings of the proposed federated learning framework and discusses their implications. We begin by evaluating the performance of the local client models, followed by an analysis of how data overlap, global graph-based aggregation, and baseline comparisons influence overall system accuracy. The results highlight the effectiveness of integrating probabilistic client outputs with a GraphConv-based global model and demonstrate the robustness of the proposed approach under non-IID data conditions.

4.7 Evaluation of Local Client Models

In federated learning systems, the performance of the global model is strongly influenced by the quality of the local client models. Therefore, before integrating client outputs into the global GraphConv-based architecture, it is essential to assess how effectively each local model performs on its own subset of data. In this study, each of the five clients trains an independent XGBoost classifier using only its local data, with no exchange of raw sequences among clients or with the central server. To ensure a fair and homogeneous evaluation, a shared validation set consisting of 350 samples is distributed to all clients. Each client generates a 9-dimensional probability vector for every validation sample, representing its predicted class distribution across the nine SARS-CoV-2 variants. These probability vectors are later integrated into the global graph model; however, they also allow us to measure and compare the standalone performance of each local classifier prior to federated aggregation. To better understand the suitability of XGBoost as the local model, we compare it against two commonly used machine learning algorithms under identical conditions: Logistic Regression, representing a simple linear classifier, and Random Forest, representing an ensemble of decision trees.

All models were trained locally on the same partitioning strategy and evaluated using the identical validation set. This comparison allows us to determine whether XGBoost provides meaningful advantages in terms of accuracy, stability, and generalization when used in a federated setting. Table 4 summarizes the performance of the three baseline models averaged across all five clients using four common classification metrics: accuracy, precision, recall, and F1-score.

TABLE 4: Performance Comparison of Client-Side Models.

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE
XGB	**92.18**	**91.82**	**93.09**	**93.17**
LR	91.91	91.24	92.81	92.94
RF	91.56	91.06	92.61	92.76

These values represent the average performance of the five client models across all federated rounds, and the standard deviation is reported to assess the stability of each method. As shown in the table, the XGBoost-based client model achieves the highest average accuracy with minimal variance, indicating strong stability and consistent behavior across different clients.

In addition to accuracy, XGBoost also demonstrates superior performance in precision, recall, and F1-score compared to the alternative models, its ability to capture complex decision boundaries and maintain a balanced trade-off between sensitivity and specificity. Logistic Regression shows a competitive accuracy level, although its higher standard deviation suggests less stability across clients. This variability can be attributed to the linear nature of the model, which limits its ability to represent the nonlinear patterns commonly observed in biological sequence data. Random Forest performs reasonably well in accuracy and precision; however, it yields the lowest F1-score among the three models. This observation may reflect the model's sensitivity to data imbalance and the heterogeneous distribution of samples across clients in a federated environment. To further investigate the impact of input data overlap on federated learning performance, two distinct overlap scenarios were evaluated. In each scenario, the accuracy of all five client models was monitored across federated rounds to assess convergence behavior, stability, and sensitivity to shared data proportions. The following figures illustrate the accuracy trajectories of the client models under both high-overlap and low-overlap conditions, providing a comparative view of how varying degrees of shared information influence local model performance. These comparative plots allow us to observe how strongly shared data subsets contribute to synchronization among client models over time. Additionally, they help reveal whether increased overlap consistently leads to faster convergence or merely reduces variance without significantly improving average accuracy.

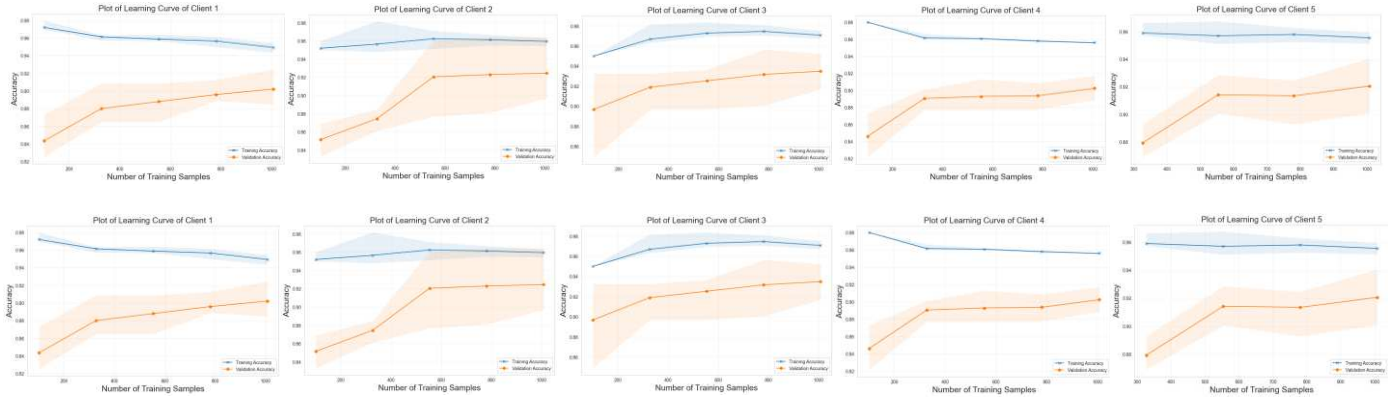

Fig. 6. Training and validation accuracy of the XGBoost model for the 5 different clients with increasing training set size (x-axis) in first Scenario(up) and second Scenario(down).

Figure 6 illustrates the learning curves of all five clients under two different data-overlap scenarios and highlights how shared information among clients influences convergence behavior and classification accuracy. In the high-overlap scenario, where adjacent clients share a larger portion of their training data (e.g., 80% overlap between Clients 1 and 2), the client models exhibit faster convergence and achieve higher validation accuracy across most training sample sizes. The curves show relatively smooth and stable trends, indicating that increased overlap helps align the local data distributions and facilitates more consistent model updates. This improved homogeneity reduces fluctuations during training and allows clients to learn more robust decision boundaries, resulting in enhanced performance stability. In contrast, the low-overlap scenario demonstrates slower convergence and noticeably increased variability across clients. Due to reduced shared information, the local datasets become more heterogeneous and non-IID, making it more challenging for the models to generalize effectively. As reflected in the curves, training accuracy remains relatively high, but validation accuracy rises more gradually and with larger confidence intervals. This behavior reflects the difficulty of learning consistent patterns when clients have limited overlap and operate on more diverse subsets of the spike sequence dataset. Overall, the comparison confirms that higher overlap leads to more stable training dynamics and faster convergence, whereas lower overlap increases data diversity but introduces greater variance in model performance. These observations highlight the importance of controlled overlap in federated genomic applications, where partial data sharing can enhance learning efficiency while maintaining privacy constraints.

4.8 Performance of the Central GraphConv Model

To evaluate the performance of the central server and its ability to integrate predictions from multiple clients, a graph-based representation is constructed from the probabilistic outputs of all local models. Each node in this graph corresponds to a single validation sample and is represented by a 9-dimensional feature vector derived from the softmax predictions of the client classifiers. These vectors collectively capture the multi-view decision patterns generated across clients. The relationships between nodes are defined using the predefined overlap ratios among client datasets, enabling the model to exploit sample-level dependencies that cannot be captured through independent client predictions alone. For supervised training of the global model, the true class label of each node is included, allowing the system to refine its internal representations while maintaining full data privacy. Once the weighted graph is constructed, it is passed to the central GraphConvModel for processing. The model follows a hierarchical architecture composed of multiple graph convolutional layers that aggregate information from neighboring nodes and progressively transform the node representations into more discriminative feature spaces.

Normalization and nonlinear activation functions are applied after each convolutional transformation to stabilize training and enhance representational richness. A single dropout layer is incorporated in the early stage of the network to reduce overfitting, while the final prediction is produced through two fully connected layers culminating in a softmax output that generates a probability distribution over the nine virus variant classes. A concise summary of the model architecture is provided in Table 5, which outlines the sequence of layers, dimensional transformations, activation functions, and regularization components used in the design.

The model is trained using CrossEntropyLoss and optimized with AdamW (learning rate 0.001, weight decay 1e−4). A step scheduler reduces the learning rate by half every 10 epochs over the 50-epoch training procedure. Training is repeated five times with random data splits, and the average performance is reported. After training the global graph model, its predictions are compared with individual client outputs using Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (2)$$

If the MSE for any client exceeds 0.01, that client's XGBoost model undergoes warm-start retraining, allowing the model to refine its decision boundaries using local data without compromising data privacy.

Before evaluating the model using the clients' real outputs, an initial training phase is performed in which the GraphConv-based global model is trained using randomly generated graphs. A total of 50 synthetic graphs are created, each consisting of five nodes. For every node, a 9-dimensional feature vector is sampled from a normal distribution, and one element of the vector is intentionally amplified to create a dominant feature. These feature vectors are then normalized so that their values form a valid probability distribution, ensuring consistency with the structure of the client-generated probability vectors used later in the federated pipeline.

TABLE 5: Layer-wise Configuration of the Central Model.

Layer	Type	Input to Output	Activation	Normalization	Dropout
1	GraphConv	9 → 512	ReLU	LayerNorm	✓ (0.2)
2	GraphConv	512 → 256	ReLU	LayerNorm	✗
3	GraphConv	256 → 128	ReLU	LayerNorm	✗
4	Fully Connected	128 → 64	ReLU	✗	✗
5	Fully Connected	64 → 9	Softmax	✗	✗

All nodes within each synthetic graph are fully connected, and every edge is assigned a weight of 1, resulting in a simple but dense graph structure that enables the model to learn basic relational patterns. Each graph is also assigned a label corresponding to one of the nine variant classes, allowing the model to be trained in a supervised manner. The GraphConv model is then trained using CrossEntropyLoss and the AdamW optimizer, serving as an initial warm-up stage to verify that the architecture can correctly process graph-structured probability features. This preliminary training step provides a controlled environment for testing the capacity of the GraphConv architecture before it is exposed to the real, client-generated probability vectors. The final output of the model for each graph is a 9-dimensional probability vector, which will later be compared with the client outputs during federated updates. After completing the training of the local client models and constructing the final graph representation using their probability outputs, the performance of the GraphConv-based central server model is evaluated on the complete set of 550 test samples. The goal of this stage is to assess how effectively the global model can integrate cross-client information and exploit inter-sample relational structure to achieve accurate SARS-CoV-2 variant classification. For this purpose, the constructed test-time graph—consisting of five nodes with 9-dimensional feature vectors derived from client softmax outputs—is fed into the GraphConvModel, which performs the final classification using its learned graph-based representations. To contextualize the effectiveness of the proposed method, its performance is compared with both centralized baselines (Spike2Vec, PWM2Vec) and federated baselines (Central MLP and Simple Averaging). Table 6 summarizes the accuracy and F1-score of all methods, providing a unified view of how the propose global graph model performs relative to existing centraliz federated approaches. To compare the effectiveness of the proposed federated GraphConv-based approach with both centralized and federated baselines, the performance of all models was evaluated on the same test dataset.

Table 6: Accuracy and F1-score of models and settings.

Method	Setting	Accuracy (%)	F1 (Macro) (%)
Spike2Vec	Centralized	68.00	49.18
PWM2Vec	Centralized	85.00	71.63
Central MLP	Federated	91.29	91.08
Simple Averaging	Federated	91.59	92.69
GraphConvModel	**Federated**	**93.67**	**93.04**

As shown in Table 6, the proposed GraphConvModel achieves the highest performance among all evaluated methods, outperforming both centralized baselines (Spike2Vec and PWM2Vec) and federated alternatives (Central MLP and Simple Averaging). Centralized methods such as Spike2Vec and PWM2Vec demonstrate reasonable accuracy when trained on complete data, yet their feature representations rely heavily on manual engineering and fixed embedding strategies. This limits their ability to capture higher-order dependencies and subtle relational signals present in spike protein sequences, resulting in noticeably lower F1-scores. Among the federated baselines, the Central MLP and Simple Averaging approaches offer improved performance compared to the centralized models, largely due to the benefit of aggregating distributed knowledge across multiple clients. However, these methods treat data samples independently and do not model structural relationships or client-specific uncertainty. As a result, their ability to generalize in non-IID settings remains constrained. In contrast, the GraphConvModel leverages a graph-based representation constructed from the probability outputs of all five clients. By integrating edge weights derived from client data overlap and performing multi-layer neighborhood aggregation, the model captures relational patterns that traditional baselines inherently overlook. This structural modeling allows the GraphConv network to refine predictions based on inter-sample similarity, effectively reducing local inconsistencies and amplifying informative patterns distributed across clients. Consequently, both accuracy and F1-score show notable improvement, demonstrating the capacity of graph neural networks to extract richer and more discriminative representations in federated environments. Overall, the enhanced performance of the proposed model highlights three key advantages:

(1) its ability to incorporate multi-client probabilistic signals into a unified relational structure;

(2) its robustness to data heterogeneity via weighted graph aggregation; and

(3) its superior capability to capture subtle genomic variations through hierarchical message passing.

Figure 7 shows the training loss and accuracy curves of the central GraphConv model over epochs, illustrating the model's convergence behavior during supervised training.

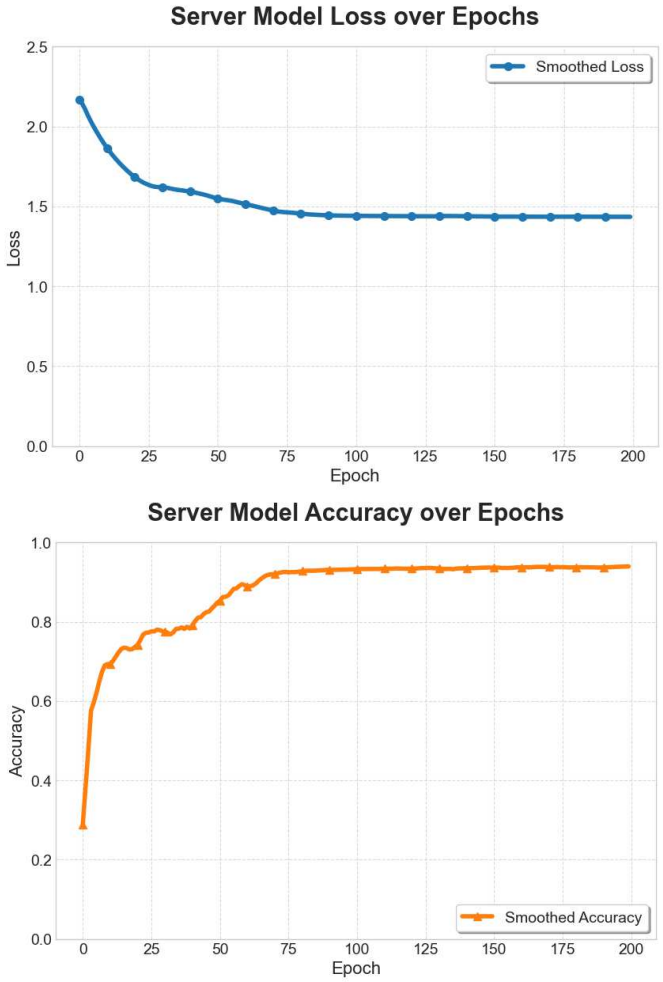

Fig. 7. Convergence of Accuracy and Loss over Epochs for GraphConv.

As illustrated in Figure 7, the accuracy of the central GraphConv model fluctuates between approximately 91% and 93.67% during training. Overall, this behavior reflects an effective learning process in which the model initially undergoes moderate oscillations while adapting to the probabilistic inputs, and then gradually stabilizes as the loss decreases. Achieving an accuracy of 93.67% after 350 evaluation epochs—using only the softmax outputs of the local client models and without any access to raw data—demonstrates the strong capability of the graph-based architecture to exploit inter-sample relationships and consolidate distributed knowledge into reliable central predictions. These results further highlight the ability of the model to generalize effectively despite receiving highly compressed information from heterogeneous local learners. Moreover, the smooth convergence of the training curves indicates well-chosen optimization settings and confirms that the GraphConv architecture is well-suited for weighted graph structures derived from federated outputs. Finally, to assess the practical value of the proposed federated graph-based approach, its performance is compared against multiple baseline methods across different computational settings. Figure 8 presents the multi-class ROC curves of the central GraphConvModel, illustrating its ability to discriminate between SARS-CoV-2 variant classes. The curves show the relationship between the

True Positive Rate (TPR) and False Positive Rate (FPR) for each class under a one-vs-rest evaluation scheme.

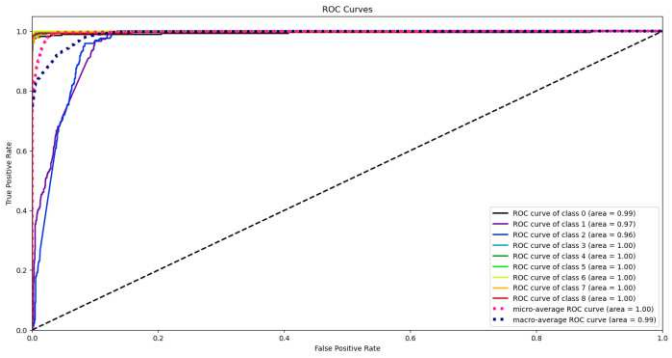

Fig. 8. ROC–AUC Analysis of the Central Graph Model.

As shown in Figure 8, the ROC curves for the central GraphConvModel demonstrate exceptionally strong discriminative capability across all nine variant classes. Most classes achieve an AUC score extremely close to 1.00, indicating near-perfect separability in the learned embedding space. Even the weakest-performing classes, such as Class 0 and Class 1, still reach AUC values around 0.99 and 0.97, which remain significantly high and signal reliable classification boundaries. The macro-averaged ROC–AUC, computed by averaging per-class AUC scores, is 0.99, confirming robust and balanced performance across all variants regardless of class frequency or distribution. Meanwhile, the micro-averaged ROC–AUC, which aggregates decisions across all samples before computing the curve, reaches 1.00, suggesting that the model is particularly effective at making correct predictions in a global, sample-weighted sense. A key observation from Figure 8 is that almost all ROC curves rapidly approach the upper-left region of the plot, which corresponds to high sensitivity (recall) with minimal false-positive rates. This behavior indicates that the central graph model successfully exploits the structural relationships encoded in the federated graph—specifically, the weighted connections derived from client overlap percentages and the probability-based node embeddings. As a result, the model is able to differentiate between variants even when their softmax distributions from client models contain subtle variations. The strong AUC performance also highlights the advantage of combining client-level probabilistic representations with graph-based aggregation. Unlike classical centralized baselines, the GraphConvModel captures inter-sample relational patterns that enhance class separability while maintaining data privacy. This demonstrates the model's ability to generalize beyond individual client outputs and synthesize collective knowledge in a federated environment.

5.Conclusion

A novel federated learning-based approach was proposed for SARS-CoV-2 variant classification, utilizing only spike protein sequences as input. The results demonstrated that this method effectively enhances predictive performance and achieves high accuracy in variant classification. Unlike traditional feature-engineering models, which rely on manually crafted features, the proposed model benefits from a self-learning framework that directly leverages sequence data for classification. A key advantage of this approach is privacy preservation, as only the outputs from local models are shared with the global model, ensuring sensitive user data remains protected. This feature is particularly crucial in healthcare, where data privacy is a significant concern, and our model addresses these challenges without compromising performance. In terms of model training, overlapping data among clients was used to simulate realistic non-IID data conditions. This approach not only helped improve model performance but also allowed for effective processing of data without the need to share raw data between clients. For the central server, we employed GraphConv graph networks, which are highly effective at modeling complex relationships in biological sequence data. This enabled the model to capture interactions between amino acids in the spike region, significantly improving performance. Future work could involve the use of deep learning-based local models for sequence classification and exploring approximate methods to reduce computational overhead and optimize training time. Combining various machine learning classifiers in a unified federated framework could further enhance performance in SARS-CoV-2 variant classification. Additionally, incorporating other features, such as geographic and temporal data, to build a vertical federated learning model is an exciting avenue for further research. Finally, investigating the generalization of the proposed model to other protein regions holds potential for extending its applicability.

References

[1] J. Majumder and T. Minko, "Recent developments on therapeutic and diagnostic approaches for COVID-19," *The AAPS journal,* vol. 23, pp. 1-22, 2021.

[2] B. Kotlar, E. Gerson, S. Petrillo, A. Langer, and H. Tiemeier, "The impact of the COVID-19 pandemic on maternal and perinatal health: a scoping review," *Reproductive health,* vol. 18, pp. 1-39, 2021.

[3] S. Patil and R. V. Kamble, "OPEN ACCESS E-RESOURCES ON COVID-19: A STUDY."

[4] R. W. Peeling and S. K. Sia, "Lessons from COVID-19 for improving diagnostic access in future pandemics," *Lab on a Chip,* vol. 23, no. 5, pp. 1376-1388, 2023.

[5] H. Ali, "AI for pandemic preparedness and infectious disease surveillance: predicting outbreaks, modeling transmission, and optimizing public health interventions," *Int J Res Publ Rev,* vol. 5, no. 8, pp. 4605-19, 2024.

[6] M. Ahmad, S. Ali, J. Tariq, I. Khan, M. Shabbir, and A. Zaman, "Combinatorial trace method for network immunization," *Information Sciences,* vol. 519, pp. 215-228, 2020.

[7] R. A. Lee, J. C. Herigon, A. Benedetti, N. R. Pollock, and C. M. Denkinger, "Performance of saliva, oropharyngeal swabs, and nasal swabs for SARS-CoV-2 molecular detection: a systematic review and meta-analysis," *Journal of clinical microbiology,* vol. 59, no. 5, pp. 10.1128/jcm. 02881-20, 2021.

[8] M. Hasan, "Evolution of Leading Mutations in Spike Glycoproteins of SARS-CoV-2," Hong Kong University of Science and Technology (Hong Kong), 2024.

[9] P. V. Markov *et al.*, "The evolution of SARS-CoV-2," *Nature Reviews Microbiology,* vol. 21, no. 6, pp. 361-379, 2023.

[10] A. Fontanet, B. Autran, B. Lina, M. P. Kieny, S. S. A. Karim, and D. Sridhar, "SARS-CoV-2 variants and ending the COVID-19 pandemic," *The Lancet,* vol. 397, no. 10278, pp. 952-954, 2021.

[11] S. M. Al Hasan, J. Saulam, F. Mikami, K. Kanda, H. Yokoi, and T. Hirao, "COVID-19 outbreak trends in South Africa: A comparison of Omicron (B. 1.1. 529), Delta (B. 1.617. 2), and Beta (B. 1.351)

variants outbreak periods," *Journal of infection and public health,* vol. 15, no. 7, pp. 726-733, 2022.

[12] C. A. Banho *et al.*, "Impact of SARS-CoV-2 Gamma lineage introduction and COVID-19 vaccination on the epidemiological landscape of a Brazilian city," *Communications medicine,* vol. 2, no. 1, p. 41, 2022.

[13] W. Yang and J. Shaman, "COVID-19 pandemic dynamics in India, the SARS-CoV-2 Delta variant and implications for vaccination," *Journal of the Royal Society Interface,* vol. 19, no. 191, p. 20210900, 2022.

[14] S. Yang, P. Hemarajata, E. E. Hilt, T. K. Price, O. B. Garner, and N. M. Green, "Investigation of SARS-CoV-2 Epsilon variant and hospitalization status by genomic surveillance in a single large health system during the 2020-2021 winter surge in Southern California," *American journal of clinical pathology,* vol. 157, no. 5, pp. 649-652, 2022.

[15] V. Tragni *et al.*, "Modeling SARS-CoV-2 spike/ACE2 protein–protein interactions for predicting the binding affinity of new spike variants for ACE2, and novel ACE2 structurally related human protein targets, for COVID-19 handling in the 3PM context," *EPMA Journal,* vol. 13, no. 1, pp. 149-175, 2022.

[16] P. Chourasia, T. Murad, Z. Tayebi, S. Ali, I. U. Khan, and M. Patterson, "Efficient classification of sars-cov-2 spike sequences using federated learning," in *Annual International Conference on Information Management and Big Data*, 2023: Springer, pp. 80-96.

[17] S. Xue, Y. Han, F. Wu, and Q. Wang, "Mutations in the SARS-CoV-2 spike receptor binding domain and their delicate balance between ACE2 affinity and antibody evasion," *Protein & Cell,* vol. 15, no. 6, pp. 403-418, 2024.

[18] S. Ali, B. Sahoo, N. Ullah, A. Zelikovskiy, M. Patterson, and I. Khan, "A k-mer based approach for sars-cov-2 variant identification," in *Bioinformatics Research and Applications: 17th International Symposium, ISBRA 2021, Shenzhen, China, November 26–28, 2021, Proceedings 17*, 2021: Springer, pp. 153-164.

[19] Z. Tayebi, S. Ali, and M. Patterson, "Robust representation and efficient feature selection allows for effective clustering of sars-cov-2 variants," *Algorithms,* vol. 14, no. 12, p. 348, 2021.

[20] S. Ali and M. Patterson, "Spike2vec: An efficient and scalable embedding approach for covid-19 spike sequences," in *2021 IEEE international conference on big data (Big Data)*, 2021: IEEE, pp. 1533-1540.

[21] B. Q. Minh *et al.*, "IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era," *Molecular biology and evolution,* vol. 37, no. 5, pp. 1530-1534, 2020.

[22] GISAID Website: https://www.gisaid.org/ (2021), [Online; accessed 29-December-2021]

[23] K. Kuzmin *et al.,* "Machine learning methods accurately predict host specificity of coronaviruses based on spike sequences alone," B*iochemical and Biophysical Research Communications, v*ol. 533, no. 3, pp. 553-558, 2020.

[24] N. Z. Zabidi *et al.,* "Evolution of SARS-CoV-2 variants: implications on immune escape, vaccination, therapeutic and diagnostic strategies," V*iruses, v*ol. 15, no. 4, p. 944, 2023.

[25] J. Wu *et al.,* "The antigenicity of SARS-CoV-2 Delta variants aggregated 10 high-frequency mutations in RBD has not changed sufficiently to replace the current vaccine strain," S*ignal Transduction and Targeted Therapy, v*ol. 7, no. 1, p. 18, 2022.

[26] C. He *et al.,* "Fedgraphnn: A federated learning system and benchmark for graph neural networks," a*rXiv preprint arXiv:2104.07145, 2*021.

[27] X. Li, Y. Gu, N. Dvornek, L. H. Staib, P. Ventola, and J. S. Duncan, "Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results," M*edical image analysis, v*ol. 65, p. 101765, 2020.

[28] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," C*omputers & Industrial Engineering, v*ol. 149, p. 106854, 2020.

[29] X.-M. Zhang, L. Liang, L. Liu, and M.-J. Tang, "Graph neural networks and their current applications in bioinformatics," F*rontiers in genetics, v*ol. 12, p. 690049, 2021.

[30] S. Kar and M. Ganguly, "Application of genomic signal processing as a tool for high-performance classification of SARS-CoV-2 variants: a machine learning-based approach," S*oft Computing, v*ol. 28, no. 4, pp. 2891-2918, 2024.

[31] K. S. Azevedo, L. C. de Souza, M. G. Coutinho, R. de M. Barbosa, and M. A. Fernandes, "Deepvirusclassifier: a deep learning tool for classifying SARS-CoV-2 based on viral subtypes within the coronaviridae family," B*MC bioinformatics, v*ol. 25, no. 1, p. 231, 2024.

[32] D. E. Wood and S. L. Salzberg, "Kraken: ultrafast metagenomic sequence classification using exact alignments," G*enome biology, v*ol. 15, pp. 1-12, 2014.

[33] S. Ali, T. E. Ali, M. A. Khan, I. Khan, and M. Patterson, "Effective and scalable clustering of SARS-CoV-2 sequences," in P*roceedings of the 5th international conference on big data research,* 2021, pp. 42-49.

[34] S. Solis-Reyes, M. Avino, A. Poon, and L. Kari, "An open-source k-mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes," P*loS one, v*ol. 13, no. 11, p. e0206409, 2018.

[35] S. Ali, B. Bello, P. Chourasia, R. T. Punathil, Y. Zhou, and M. Patterson, "PWM2Vec: An efficient embedding approach for viral host specification from coronavirus spike sequences," B*iology, v*ol. 11, no. 3, p. 418, 2022.

[36] N. Latif, W. Ma, and H. B. Ahmad, "Advancements in securing federated learning with IDS: a comprehensive review of neural networks and feature engineering techniques for malicious client detection," A*rtificial Intelligence Review, v*ol. 58, no. 3, p. 91, 2025.

[37] A. Natarajan, H.-W. Su, and C. Heneghan, "Assessment of physiological signs associated with COVID-19 measured using wearable devices," N*PJ digital medicine, v*ol. 3, no. 1, p. 156, 2020.

[38] J. D. Figueroa *et al.,* "The impact of the Covid-19 pandemic on breast cancer early detection and screening," P*reventive medicine, v*ol. 151, p. 106585, 2021.

[39] Y. Choi and M. K. Lee, "Neuroimaging findings of brain MRI and CT in patients with COVID-19: A systematic review and meta-analysis," E*uropean journal of radiology, v*ol. 133, p. 109393, 2020.

[40] W. Zhang *et al.,* "Dynamic-fusion-based federated learning for COVID-19 detection," I*EEE Internet of Things Journal, v*ol. 8, no. 21, pp. 15884-15891, 2021.

[41] Y. Igarashi *et al.,* "Machine learning prediction for supplemental oxygen requirement in patients with COVID-19," J*ournal of Nippon Medical School, v*ol. 89, no. 2, pp. 161-168, 2022.

[42] N. Boscarino, R. A. Cartwright, K. Fox, and K. S. Tsosie, "Federated learning and Indigenous genomic data sovereignty," N*ature machine intelligence, v*ol. 4, no. 11, pp. 909-911, 2022.

[43] K. Wang, L. Cheng, and B. Yong, "Spectral-similarity-based kernel of SVM for hyperspectral image classification," R*emote Sensing, v*ol. 12, no. 13, p. 2154, 2020.

[44] Y. Liu, H. Pu, and D.-W. Sun, "Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices," T*rends in Food Science & Technology, v*ol. 113, pp. 193-204, 2021.

[45] Á. G. Dieste *et al.,* "ResNeTS: a ResNet for Time Series Analysis of Sentinel-2 Data Applied to Grassland Plant-Biodiversity Prediction," I*EEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2*024.

[46] T. Murad, S. Ali, I. Khan, and M. Patterson, "Spike2CGR: an efficient method for spike sequence classification using chaos game representation," M*achine Learning, v*ol. 112, no. 10, pp. 3633-3658, 2023.

[47] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *in International Conference on Learning Representations (ICLR)*, 2017.

[48] WHO Website: https://www.who.int/en/activities/tracking-SARS-CoV-2-variants/

[49] A. P. West Jr *et al.,* "Detection and characterization of the SARS-CoV-2 lineage B. 1.526 in New York," N*ature communications, v*ol. 12, no. 1, p. 4886, 2021.

[50] C. Franco-Muñoz *et al.,* "Substitutions in Spike and Nucleocapsid proteins of SARS-CoV-2 circulating in South America," I*nfection, Genetics and Evolution, v*ol. 85, p. 104557, 2020.