# Spike2Vec: An Efficient and Scalable Embedding Approach for COVID-19 Spike Sequences

**Authors**

- SARWAN ALI
- MURRAY PATTERSON

# Table of Content

- Motivation
- Problem
- Real-World Applications
- Challenges
- Previous Work
- Our Contribution
- Proposed Approach
- Dataset Statistics
- Data Visualization
- Results
- Conclusion

# Motivation

- With the rapid global spread of COVID-19, more and more data related to this virus is becoming available
    - Genomic sequence data
    - Clinical Data
- The total number of genomic sequences that are publicly available on platforms such as GISAID is currently several million, and is increasing with every day
- The availability of such “Big Data” creates a new opportunity for researchers to study this virus in detail
- This is particularly important with all of the dynamics of the COVID-19 variants which emerge and circulate

# Research Problem

- How can we design a fixed length representation of protein sequences that can be scaled to multi-million sequences?

# Real World Applications

- Genomic surveillance: Tracking the spread of pathogens in terms of genomic content
- Real time identification of new and rapidly emerging variants
- Track the spread of known variants in new municipalities, regions, countries and continents

# Challenges

- The number of sequences is so huge that any way of extracting useful features becomes even more critical
- Mutations happen disproportionately in the spike region of the genome

# Previous Work

- Some efforts have been done to perform classification and clustering of SARS-CoV-2 spike sequences
- However, those methods are not scalable to the amount of data we use in this study
- Although they were successful in getting high predictive accuracy, it is not clear if the proposed methods are robust and will give the same predictive performance on larger datasets

# Our Contribution

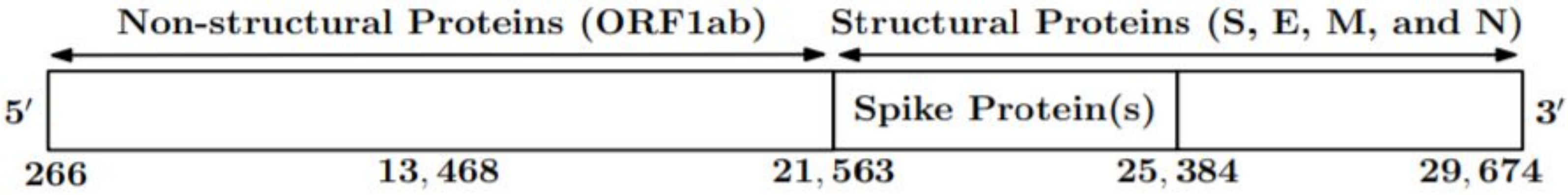
- We propose an embedding approach, called Spike2Vec that outperforms the baseline classification method in terms of predictive accuracy
- We show that our method is scalable on larger datasets by using ≈2.5 million spike sequences
- We show from the results that the machine learning models are not efficiently scalable on these larger datasets while using traditional embedding approach.
- This robust checking helps us to analyze the machine learning models in detail in terms of their appropriateness for SARS-CoV-2 spike sequences
- We also show that in terms of clustering, our embedding approach is better than the baseline model

# Proposed Approach

- Use of Spike Sequence
- One-Hot Encoding
- k-mers Generation from Spike Sequences (Spike2Vec)
- Frequency Vectors Generation
- Keras Classifier
- Low Dimensional Representation of Data
- Classification and Clustering

# Spike Sequence

- Since the spike protein is the entry point of the virus to the host cell, it is an important characterizing feature of a coronavirus
- the mRNA vaccines (e.g., Pfizer and Moderna) for COVID-19 are designed to target only the SARS-CoV-2 spike protein (unlike traditional vaccines which comprise an entire virome)
- Since the spike region is sufficient to characterize most of the important features of a viral sample, yet is much smaller in length, we focus on an embedding approach tailored to the spike region of the sequences


Non-structural Proteins (ORF1ab)
Structural Proteins (S, E, M, and N)
5′
Spike Protein(s)
3′
266
13,468
21,563
25,384
29,674

# K-mers Generation

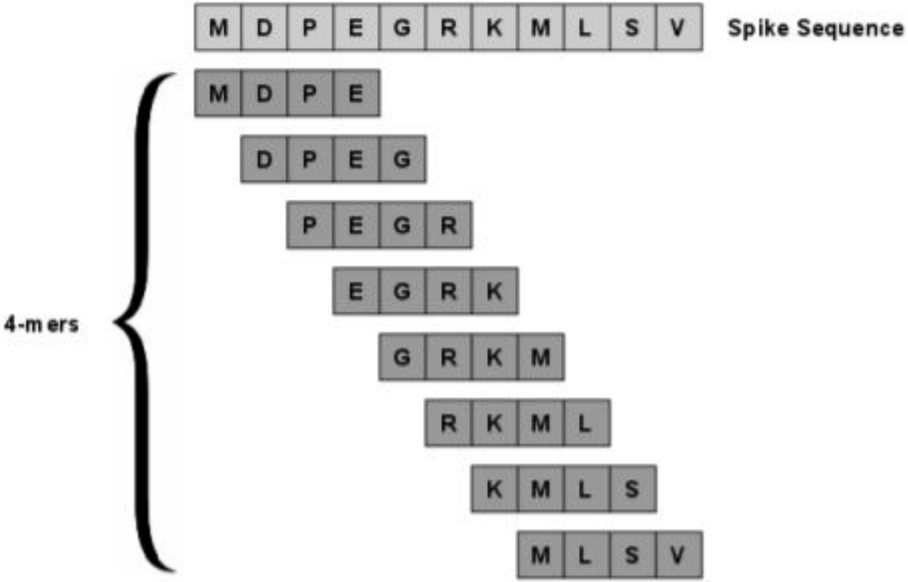


Fig. 2: Example of 4-mers of the amino acid sequence "MD-PEGRKMLSV".

# Frequency Vectors Generation

- We design a feature vector that contains the count of each k-mer in its respective spike sequence
- Each sequence A is over an alphabet Σ (amino acids of the spike sequence)
- These fixed length frequency vectors have length |Σ|^k (the number of possible k-mers of a spike sequence)
- Since the total number of alphabets in our data are 21 (the number of amino acids), the length of each frequency vector becomes 21^3 = 9261

# Low Dimensional Representation

- For typical supervised and unsupervised  classification/clustering tasks, dimensionality reduction methods such as principal component analysis, ridge regression, and lasso regression are used
  - Problem: Not scalable on bigger data
- Solution: User Kernel method with Kernel Trick
- <u>Kernel Trick:</u> It is used to generate features for an algorithm which depends on the inner product between only the pairs of input data points. The main idea is to avoid the need to map the input data (explicitly) to a high-dimensional feature space

# Kernel Trick

- Kernel Trick relies on the following observation:
  - Any positive definite function f(a,b), where a, b ∈ R^d, defines an inner product and a lifting φ so that we can quickly compute the inner product between the lifted data points

$$\langle \phi(a), \phi(b) \rangle = f(a, b)$$

- Drawback: In case of large training data, the kernel method suffers from large initial computational and storage costs.

# Random Fourier Features (RFF)

- To overcome these computational problems, we use an approximate kernel method called random Fourier features (RFF)
- RFF maps the input data to a randomized low dimensional feature space (euclidean inner product space)

$$z : \mathcal{R}^d \rightarrow \mathcal{R}^D$$

- In this way, we can approximate the inner product between a pair of transformed points

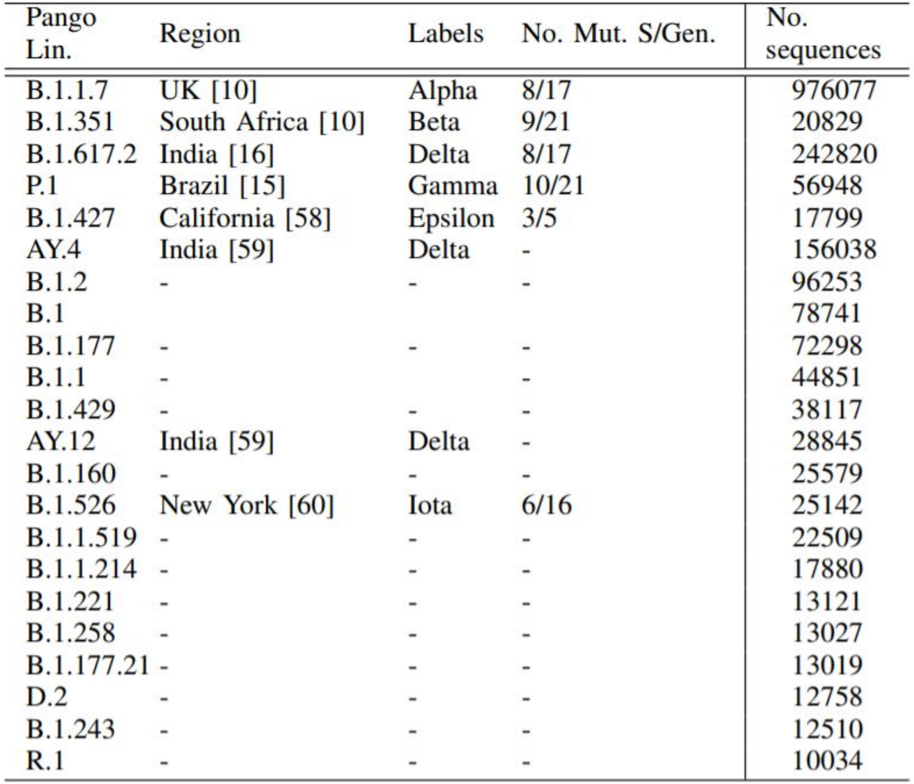
$$f(a,b) = \langle \phi(a), \phi(b) \rangle \approx z(a)' z(b)$$

- z is low dimensional (unlike the lifting φ)
- In this way, we can transform the original input data with z, which acts as the approximate low dimensional embedding for the original data

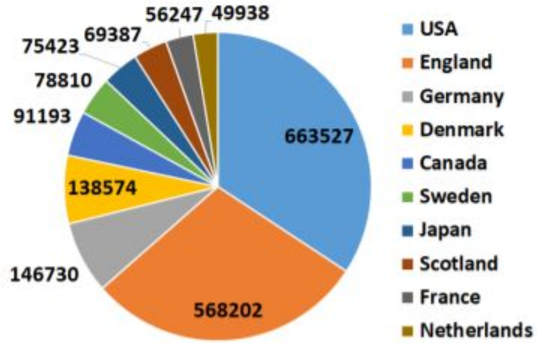
# Classification and Clustering

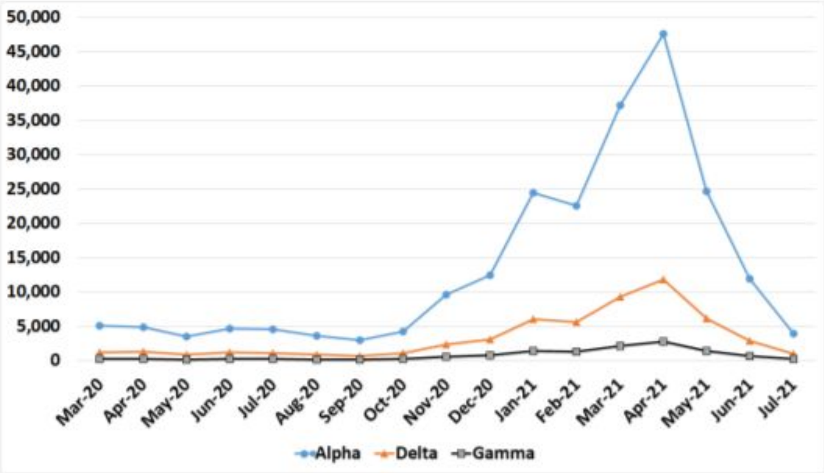
- The low dimensional representation from RFF is used as an input for different ML tasks
- Baseline Approach: One-Hot Embedding (OHE)
- Classification:
  - Naive Bayes
  - Ridge Classifier
  - Logistic Regression
- Clustering:
  - k-means

# Dataset



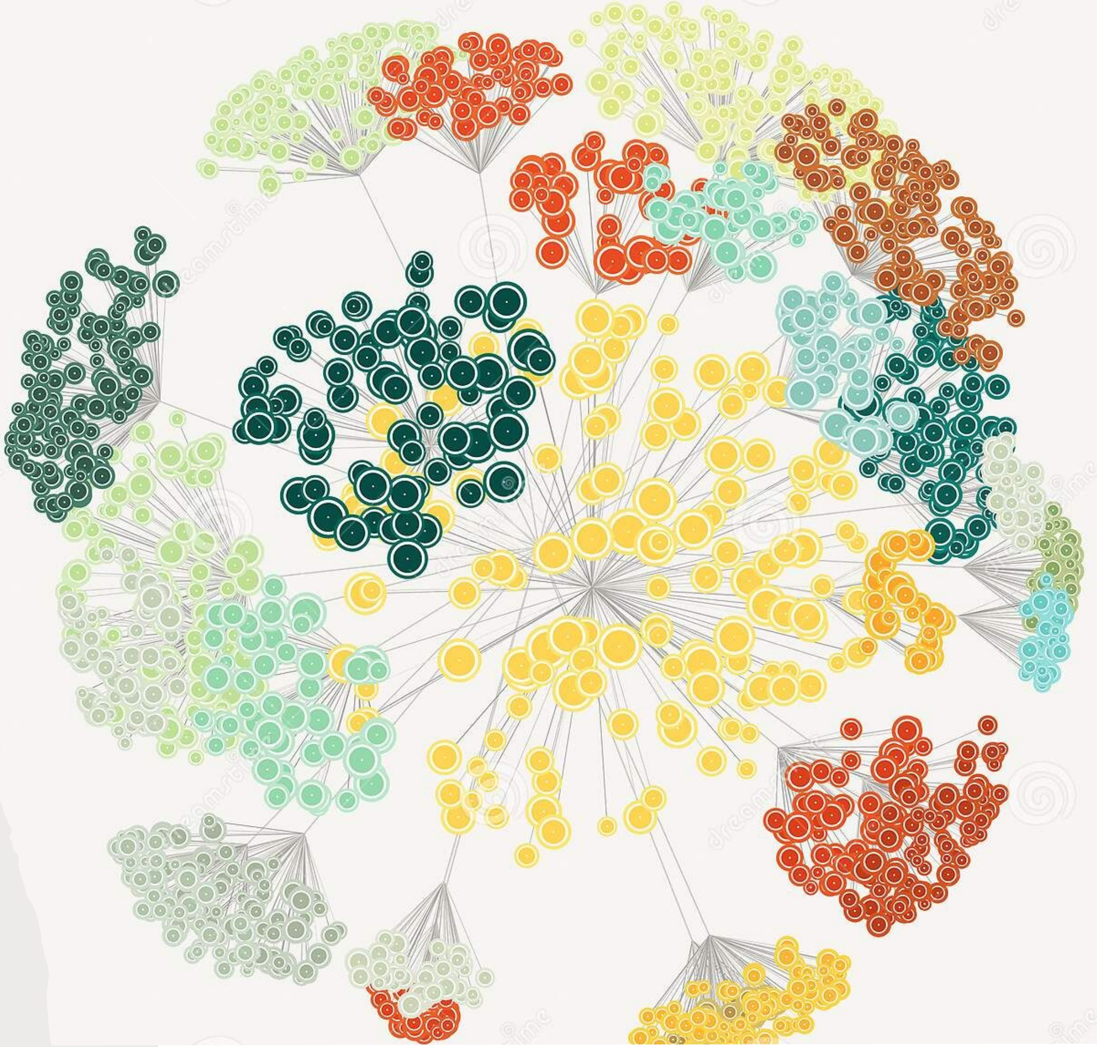
| Pango Lin. | Region | Labels | No. Mut. S/Gen. | No. sequences |
|---|---|---|---|---|
| B.1.1.7 | UK [10] | Alpha | 8/17 | 976077 |
| B.1.351 | South Africa [10] | Beta | 9/21 | 20829 |
| B.1.617.2 | India [16] | Delta | 8/17 | 242820 |
| P.1 | Brazil [15] | Gamma | 10/21 | 56948 |
| B.1.427 | California [58] | Epsilon | 3/5 | 17799 |
| AY.4 | India [59] | Delta | - | 156038 |
| B.1.2 | - | - | - | 96253 |
| B.1 | | | | 78741 |
| B.1.177 | - | - | - | 72298 |
| B.1.1 | - | | - | 44851 |
| B.1.429 | - | - | - | 38117 |
| AY.12 | India [59] | Delta | - | 28845 |
| B.1.160 | - | - | - | 25579 |
| B.1.526 | New York [60] | Iota | 6/16 | 25142 |
| B.1.1.519 | - | - | - | 22509 |
| B.1.1.214 | - | - | - | 17880 |
| B.1.221 | - | - | - | 13121 |
| B.1.258 | - | - | - | 13027 |
| B.1.177.21 | - | - | - | 13019 |
| D.2 | - | - | - | 12758 |
| B.1.243 | - | - | - | 12510 |
| R.1 | - | - | - | 10034 |



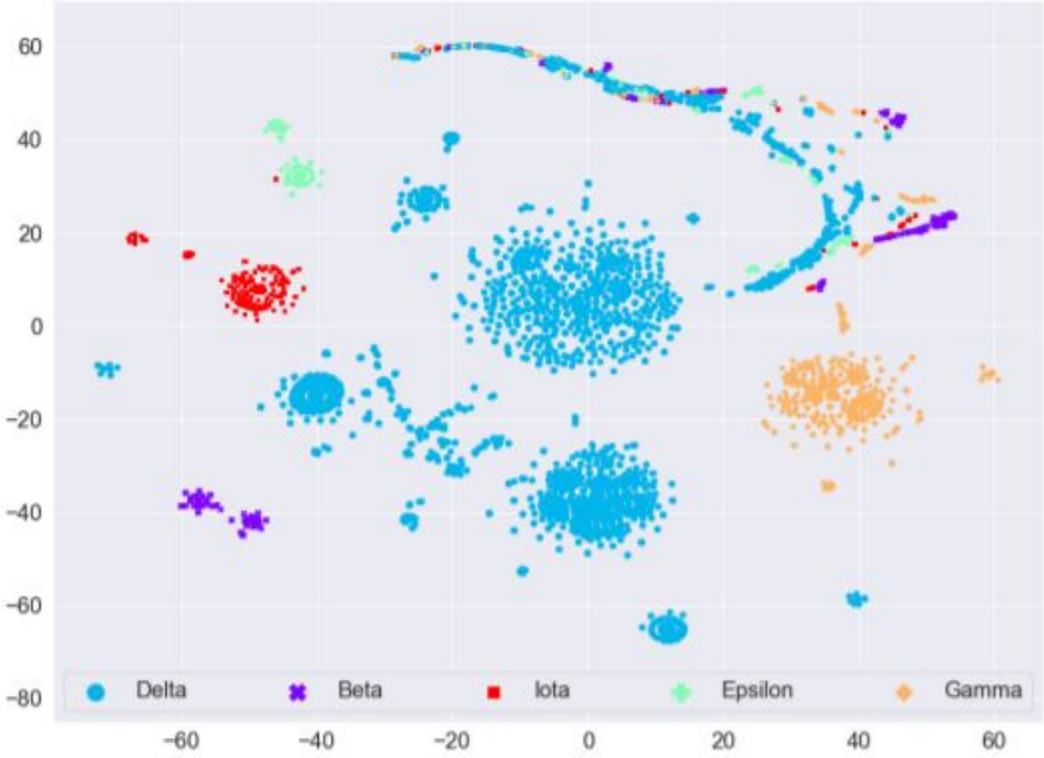


Spread pattern of Alpha (blue line), Delta (orange line), and Gamma (black line) variants in the USA from March 2020 to July 2021. The y-axis shows the total number of COVID-19 infected patients.

# Data Visualization

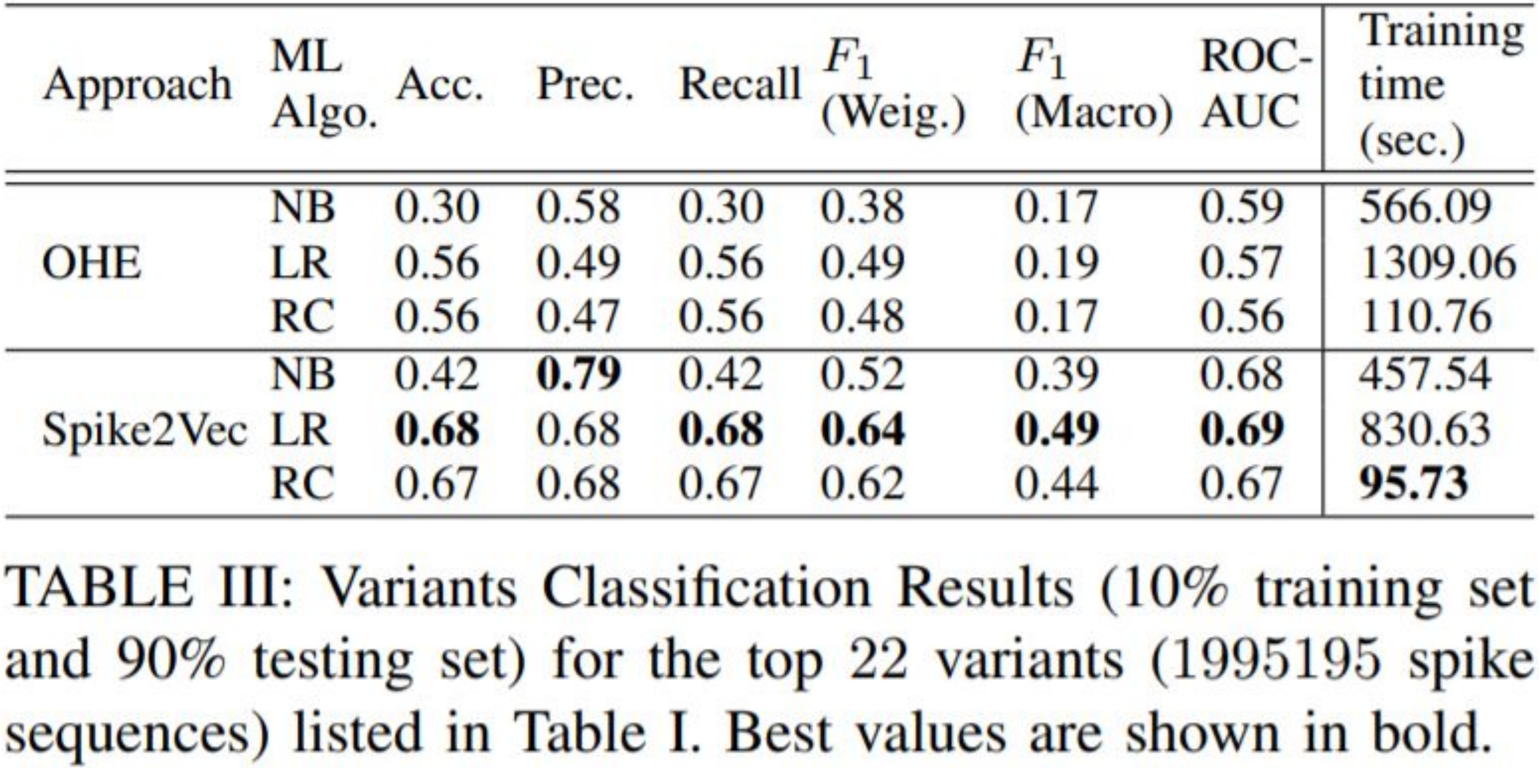


# t-distributed stochastic neighbor embedding (t-SNE)



# Results

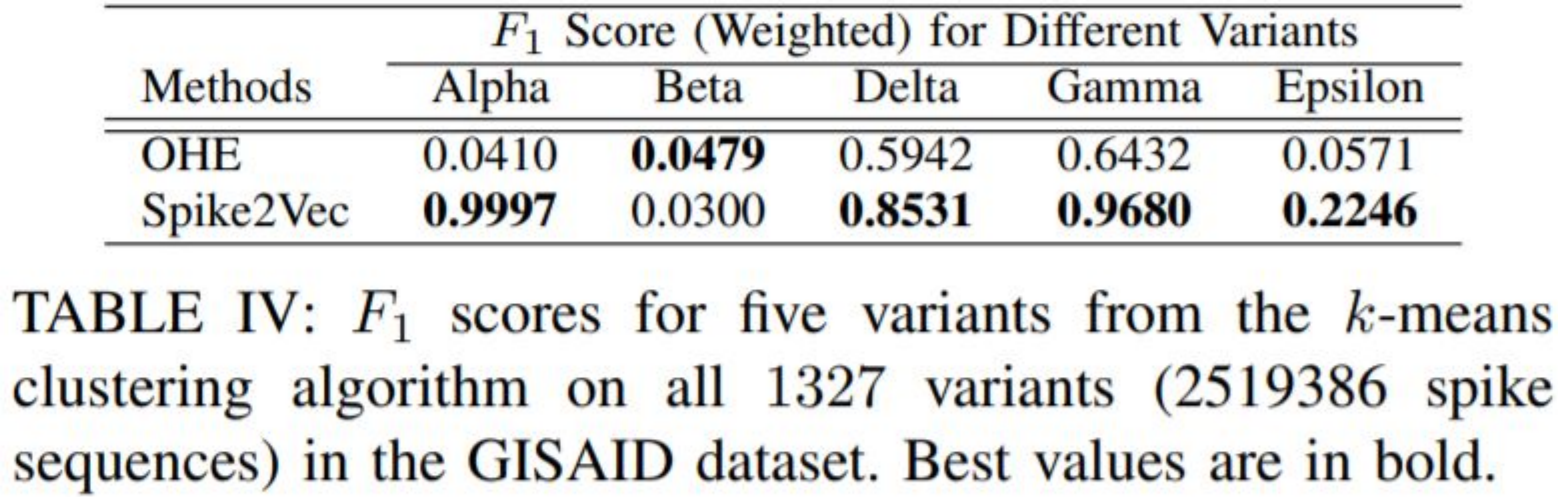
# Classification Results



| Approach | ML Algo. | Acc. | Prec. | Recall | $F_1$ (Weig.) | $F_1$ (Macro) | ROC-AUC | Training time (sec.) |
|---|---|---|---|---|---|---|---|---|
| OHE | NB | 0.30 | 0.58 | 0.30 | 0.38 | 0.17 | 0.59 | 566.09 |
| | LR | 0.56 | 0.49 | 0.56 | 0.49 | 0.19 | 0.57 | 1309.06 |
| | RC | 0.56 | 0.47 | 0.56 | 0.48 | 0.17 | 0.56 | 110.76 |
| Spike2Vec | NB | 0.42 | **0.79** | 0.42 | 0.52 | 0.39 | 0.68 | 457.54 |
| | LR | **0.68** | 0.68 | **0.68** | **0.64** | **0.49** | **0.69** | 830.63 |
| | RC | 0.67 | 0.68 | 0.67 | 0.62 | 0.44 | 0.67 | **95.73** |

TABLE III: Variants Classification Results (10% training set and 90% testing set) for the top 22 variants (1995195 spike sequences) listed in Table I. Best values are shown in bold.

# Clustering Results



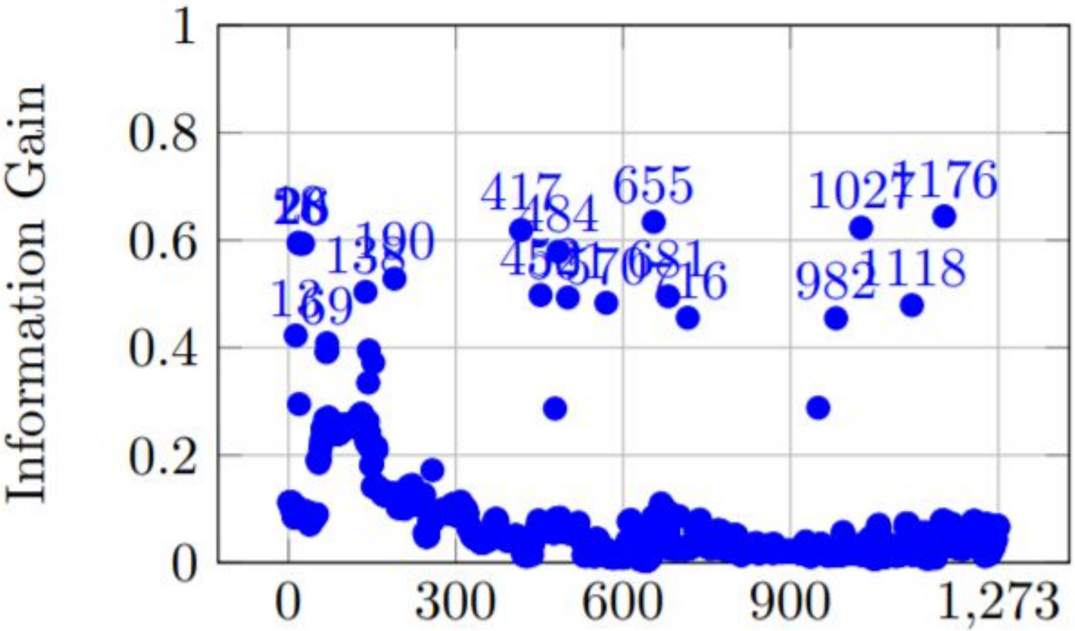
| Methods | $F_1$ Score (Weighted) for Different Variants | | | | |
|---|---|---|---|---|---|
| | Alpha | Beta | Delta | Gamma | Epsilon |
| OHE | 0.0410 | **0.0479** | 0.5942 | 0.6432 | 0.0571 |
| Spike2Vec | **0.9997** | 0.0300 | **0.8531** | **0.9680** | **0.2246** |

TABLE IV: $F_1$ scores for five variants from the $k$-means clustering algorithm on all 1327 variants (2519386 spike sequences) in the GISAID dataset. Best values are in bold.

# Importance of each amino acid

**Technique:**

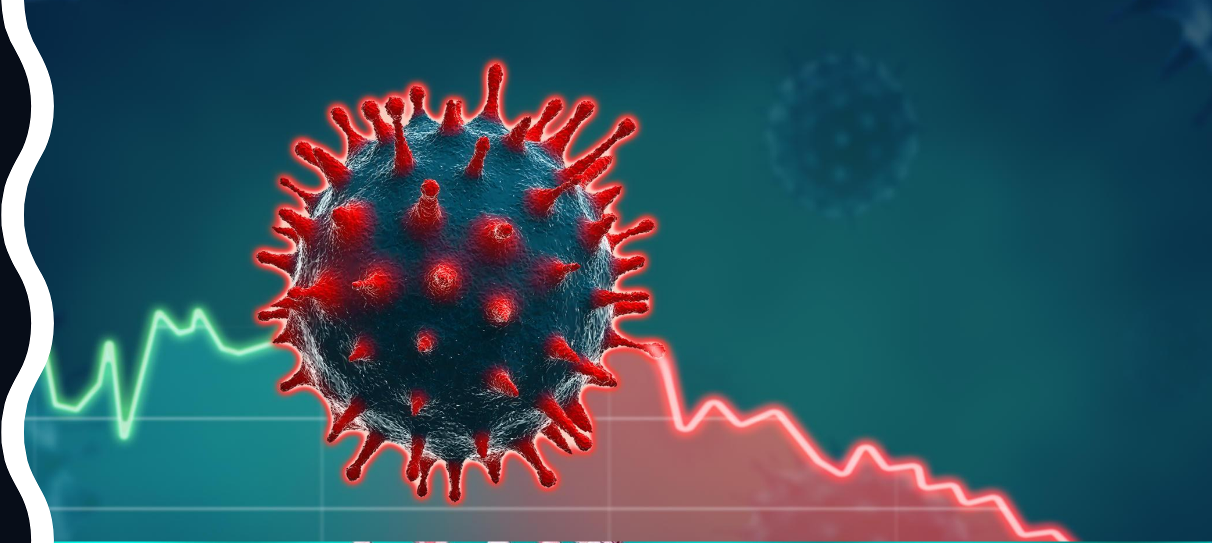
- We compute Information Gain (IG) between each attribute (amino acid position) and the class (variant).
- The USA's Centers for Disease Control and Prevention (CDC) identified mutations at certain positions from one variant to the other. We use their mutation information to compare them with the attributes having high IG values in the figure above.

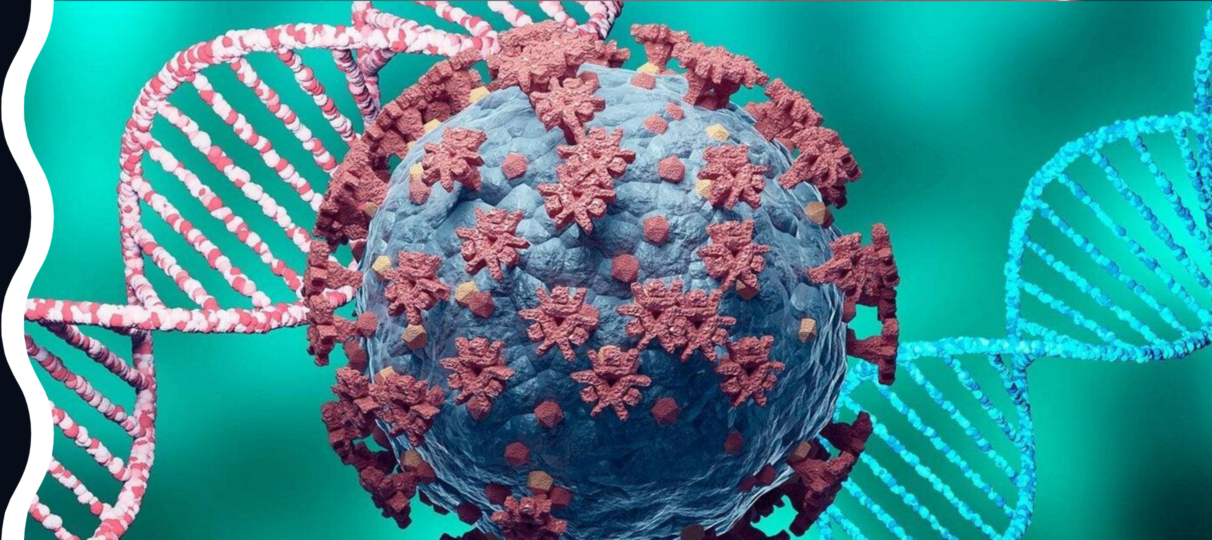


**Observation:** Our high IG value attributes are the same as those given by CDC.

# Conclusion

- We propose an embedding approach that can be used to perform different machine learning tasks on the SARS-CoV-2 spike sequences
- We show that our model can scale to several million sequences, and it also outperforms the baseline models significantly
- Since the COVID-19 disease is relatively new, we do not have enough information available for different coronavirus variants so far





# Future Work

- We will explore the new (and existing) variants in more detail in the future.
- We will use deep learning models to enhance the prediction performance of Spike2Vec
- We will use full genome data for variant classification
- We will use short read data to further understand the behavior of the coronavirus

# Questions!!

# External Resources

- Dataset: https://drive.google.com/drive/folders/1-YmlM8ipFpj-glr9hSF3t6VuofrpgWUa
- Code: https://github.com/sarwanpasha/Spike2Vec