www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(4), Sep-Dec 2025, 131-145. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i4.1142

Dimensionality reduction for SARS-CoV-2 antibodies prediction

Yasmín Hernández[1], *Samuel Narciso-Galván*[1], *P. Alejandra Cuevas-Chavez*[1], *Javier Ortiz-Hernández*[1], *Juan Miguel-Ruiz*[1]

[1] TecNM Centro Nacional de Investigación y Desarrollo Tecnológico, México
{yasmin.hp, m22ce051, d18ce074, javier.oh, d20ce065}@cenidet.tecnm.mx

Abstract. Antibodies are proteins that bind to specific antigens to inactivate pathogens. Antibody classification requires datasets with structural and functional information about antibodies. This paper examines the impact of dimensionality reduction on the classification of SARS-CoV-2 antibodies from genetic sequence data of the Observed Antibody Space database. We focus on transforming amino acid sequences into word embeddings. The transformed data presents the challenge of the curse of dimensionality, which can affect the performance of the models. To address this problem, two dimensionality reduction techniques are evaluated: Principal Component Analysis and Uniform Manifold Approximation and Projection. We developed 36 classification models using Support Vector Machines, Random Forest, and k-Nearest Neighbors algorithms. The objective is to determine whether dimensionality reduction improves model performance. The study provides insights into how these techniques can facilitate predictive analysis in bioinformatics and contribute to the development of efficient models for identifying relevant antibodies in immunology.

Keywords: Dimensionality reduction, PCA, predictive model, SARS-CoV2 antibodies, UMAP.

Article Info
Received May 26, 2025
Accepted June 24, 2025

1 Introduction

Data mining is a multidisciplinary field of study and research encompassing research and applications such as predictive analytics, statistical modeling, pattern extraction, and bioinformatics. Data mining and machine learning have been applied to address complex life science problems such as protein structure prediction, disease gene identification, and expression analysis of genes (Zaki et al., 2003). Data mining in bioinformatics enables detailed analysis of genomic data to comprehend biological processes at the molecular level. One of the most challenging emerging applications is the classification of antibodies according to the antigens they bind. Antibodies, the heart of the immune system, are proteins that bind to specific antigens to inactivate pathogens. Antibody classification requires datasets with structural and functional information about antibodies. The Observed Antibody Space (OAS) is a dataset collecting genomic sequences of antibodies from several species, within the context of diseases such as HIV, Ebola, and SARS-CoV-2. OAS contains information on the structure of antibodies, such as heavy and light chains, and it is an excellent resource for biological research and immunology studies (OAS, 2024).

Although OAS data are highly informative and detailed, the genomic sequences representation is not appropriate for machine learning algorithms, for example, the OAS datasets do not have labels to categorize the antibodies as belonging to or being matched to antigens, therefore a preprocessing of data is needed. Additionally, this data must be focused at the most significant parts of the antibody, the complementarity determining regions, CDR, which are functionally significant regions in antibodies that bind antigens. CDR3 provides specific information related to antigen recognition (Xu & Davis, 2000). Because of its critical significance, the region CDR3 is of special interest to computational processing and antibodies classification, therefore these amino acid sequences must be converted to appropriate representations in order to be used by machine learning models.

Natural language processing approaches were originally developed for text analysis, but they have been adapted to biological sequence analysis. Amino acid sequences in the CDR3 can be translated into word embeddings, a vector representation preserving semantic relations among sequences, alike words preserve meaning in text. However, transforming biological

sequences into word embeddings has one more challenge: the high dimensionality of the resulting embeddings, typically up to 100 dimensions. Such a dimensionality may increase computational complexity and decrease the generalization ability of machine learning models. To focus on this problem, dimensionality reduction techniques must apply to allow retaining only significant dimensions, removing noise, and enhancing model performance without losing important information.

In this paper, we proposed a pipeline to process antibodies data from OAS, specifically those which targeting SARS-CoV-2, by encoding the sequences of the CDR3 region into word embeddings. We focus on the impact of dimensionality reduction on classification model performance by comparing model performance on original and reduced data. This evaluation will facilitate the establishment of the effectiveness of dimensionality reduction techniques in increasing the performance of predictive models for antibody analysis. The paper is organized as follows. The remainder of this section presents the related work. Section 2 explained the experimental procedures to build the predictive models. In Section 3, we discussed the results of using PCA and UMAP as dimensionality reduction techniques. Finally, conclusions are presented in Section 4.

1.2 Related Work

This section reviews relevant and recent research in dimensionality reduction and biological sequence processing, highlighting the advances and limitations of current techniques.

Wang (2019) proposed a method for selecting the optimal number of dimensions in word embeddings using Principal Component Analysis (PCA). This method removes dimensions one by one until the best model accuracy is achieved. The process begins by transforming the embedding using PCA. Dimensions are then progressively removed, starting with those that contribute the least to the variance explanation. Although each dimension contributes differently to the explained variance, they all contribute equally to the inner product calculation. Dimensions with lower variance but equal weighting in the inner product can decrease the discriminative power of the model. Therefore, removing them allows focusing on the most discriminative dimensions. This approach offers an alternative to the traditional use of cumulative variance. Although the method has not been tested with amino acid sequences, it could be useful in this context.

Zebari et al. (2020) conducted a literature review covering articles published between 2017 and 2020, focusing on dimensionality reduction techniques. The review distinguishes between two main approaches: feature selection techniques and feature extraction techniques. In most of the reviewed articles on feature selection, new techniques or combinations of techniques are compared and evaluated using classification algorithms. Meanwhile, in the reviewed articles on feature extraction, PCA stands out as the most common technique. As with feature selection, the performance of extraction techniques is evaluated using classification algorithms. Furthermore, the review reveals that the most used classification algorithms in the reviewed studies are Support Vector Machines and K-Nearest Neighbors. It also shows that the most used metric to evaluate the effectiveness of dimensionality reduction techniques is accuracy.

Arowolo et al. (2021) conducted a review of studies applying dimensionality reduction techniques focusing on those that optimize data clustering efficiently, reduce computational processing time, and improve classification in RNA sequencing (RNA-Seq). The reviewed studies developed variants of existing techniques such as Principal Component Analysis, t-distributed Stochastic Neighbor Embedding, Locally Linear Embedding, Isomap, Diffusion Maps, and Laplacian Eigenmaps. Regarding classification techniques, the reviewed studies preferred Support Vector Machines and Multilayer Perceptron neural networks. These techniques stand out for their ability to handle the complexity and high dimensionality of RNA-Seq data, contributing to improving classification accuracy in malaria vector analysis.

Enríquez et al. (2021) compare the dimensionality reduction techniques PCA, and the CUR algorithm applied to COVID-19 testing data obtained from a clinical laboratory in Ibarra, Ecuador. The dataset used contains a few instances and dimensions. Although PCA is traditionally oriented towards attribute extraction, the experiment was focused on identifying the most relevant attributes (performing attribute selection). PCA reduced the original 7 dimensions to only 3, maintaining 80% of the data variance. However, the results obtained with PCA were more difficult to interpret, complicating the identification of the most relevant attributes. In contrast, the CUR algorithm showed a greater relevance of each attribute, reducing to 3 dimensions, but with a better ability to identify the most significant attribute in the dataset. For this reason, the authors highlight the CUR algorithm for its superior interpretability compared to PCA.

Adjuik and Ananey-Obiri (2022) developed a COVID-19 virus sequences classification model using numerical protein vectors generated and the word embedding technique. They used two datasets. The first contained COVID-19 virus protein sequences, and the second was obtained from the NCBI platform. The amino acid sequences were transformed into numerical vectors using

the Continuous Bag of Words (CBOW) model. Since the initial vectors had 200 dimensions, the PCA dimensionality reduction technique was applied, reducing the vectors to only 10 dimensions. These new vectors were used to train several classification models, including Logistic Regression, Random Forest, Support Vector Machines, K-Nearest Neighbors, and Linear Discriminant Analysis. They evaluated the models based on their accuracy, highlighting the Random Forest-based model as the best performer.

Brandes et al. (2022) proposed ProteinBERT, a deep language model designed to capture the unique features of proteins. ProteinBERT is pre-trained by combining language modeling with a Gene Ontology annotation prediction task, allowing the model to learn the structure of protein sequences and their biological function. This approach enables the model to process local, sequence-specific details and global, broader context representations, offering a comprehensive analysis of protein sequences. Experiments show that ProteinBERT is highly effective in predicting protein properties, even when using limited datasets, suggesting its potential as a valuable tool in bioinformatics for protein function and feature prediction. However, due to the deep learning nature of the model, its high computational cost is relevant to consider.

Khalilian et al. (2022) proposed a deep learning dimensionality reduction model called VAEResDR, which combines a Variational Autoencoder (VAE) with a Residual Neural Network (ResNet). This model was designed to analyze heavy chain CDR3 sequences to identify hidden patterns and improve the clustering of antibody and nanobody CDR3 sequences. Experiments were performed using several datasets with higher records. On the other hand, string sequences were converted to numerical vectors using the One-Hot Encoding technique. Two experiments were conducted. In the first one, the ability of different dimensionality reduction techniques to maintain the relatedness of data within a single cluster after reducing the dimensions to only two was compared. In the second one, an additional group was added, and the analysis was repeated. The techniques compared included PCA, t-SNE, UMAP, VAE, scCCESS(AE), VAEDR, and the proposed VAEResDR technique. In both experiments, VAEResDR obtained a higher performance, as the clusters maintained their structure and the relationship between the data better than the other dimensionality reduction techniques.

Rustam et al. (2022) proposed a dimensionality reduction algorithm called Modified Weiszfeld (MWA), which was compared with other dimensionality reduction algorithms such as PCA, Classical Multidimensional Scaling (CMDS), Laplacian Eigenmaps, and Locally Linear Embedding (LLE). Experiments were conducted using a metabolite dataset from Indonesian clove shoots. To evaluate the data quality after dimensionality reduction, the quality of the generated clusters was analyzed using the fuzzy c-means algorithm. The results showed that MWA and PCA were the most effective techniques as they reduced the dimensions to a smaller number while maintaining the correct relationship between the data. The clusters formed by these techniques contained more coherent and interrelated data than the clusters generated by the other evaluated techniques.

Weber et al. (2024) conducted a comprehensive review, exploring recent advances in T cell receptor binding prediction using machine learning techniques. The development of immuno-sequencing techniques and experimental methods has generated a vast amount of data on the TCR repertoire. This is driving the creation of increasingly sophisticated predictive models. The evolution of these techniques has followed a path ranging from unsupervised clustering approaches to supervised classification models, culminating in the most recent applications of Protein Language Models, PLMs. The review highlights the significant impact of transformer-based models in bioinformatics. These models, pre-trained on bast collections of unlabeled protein sequences, allow the conversion of amino acid sequences into vectorized embeddings that capture relevant biological properties. Recent attempts to leverage PLMs have achieved competitive results in tasks related to TCR binding prediction.

2 Predictive Models construction

This section details how the data related to SARS-CoV-2 antibodies was obtained to build the datasets based on the most variable region of antibodies, CDR3. Furthermore, we described the processing of these amino acid sequences using natural language processing techniques and the subsequent generation of classification models. These models were developed with original datasets, without dimensionality reduction, and with data whose dimensionality was reduced using techniques such as Principal Component Analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP).

2.1 Antibodies datasets

From OAS we obtained 468 datasets from healthy individuals and 266 datasets from sick individuals with SARS-CoV-2 disease. Each dataset contains the antibodies of a single individual, and each sample is an antibody present in that individual. We conducted an analysis over these 734 datasets to identify and remove those datasets containing fewer than 100 samples

(antibodies) in order to have enough data for the subsequent stages of the study. This process reduced the datasets from 734 to 395, from 172 healthy individuals and 223 sick individuals with SARS-CoV-2.

Our approach to build the datasets is based on the premise that when an individual is infected with a virus, their immune system responds by significantly increasing the production of specific antibodies to attack and neutralize that virus (Stafford et al., 2016). Therefore, although the original datasets contain antibodies that respond to other diseases, the most repeated antibodies are expected to be those specific to SARS-CoV-2 since the samples were collected from infected individuals with this virus at the time of sampling. This same approach was also applied to the 172 datasets of healthy individuals, although the antibody distribution is anticipated to be different, reflecting a lower presence of specific antibodies.

We grouped and ordered CDR3 sequences according to their frequency of occurrence in each dataset. Then, we built three datasets: CD1, CD10 and CD100. In CD1, we grouped the most frequent CDR3 in each of the 395 datasets, resulting in a dataset with 395 samples representing the predominant antibody in each individual. In CD10, we grouped the ten most frequent CDR3 from each dataset, generating 3,950 records. In CD100, we grouped the 100 most frequent antibodies in each dataset, resulting in a dataset with 39,500 records.

In addition to the amino acid sequences in the CDR3, we added three attributes: CDR length, CDR frequency and class. These attributes enrich the CD1, CD10, and CD100 datasets and allow a more detailed and precise analysis. CDR length is the number of amino acids in the CDR3 sequence; since CDR3 length can vary significantly between different antibodies, this attribute is relevant to understanding how variations in length may relate to the specificity and binding efficiency of the antibody to the antigen. The CDR frequency indicates the number of times a specific CDR3 appears in the original dataset. The class attribute labels each record according to the origin of the sample; there are two classes: SARS and NoSARS. The SARS class was assigned to samples of sick individuals, and the NoSARS class to those from healthy individuals. Table 1 shows some representative examples of the datasets.

Table 1. Extract of the datasets

CDR3	CDR Length	Frequency	Class
ARVFPRWLQFDPYFDY	16	4949	SARS
AKGATKVDY	9	612	SARS
ARGEDSAKLGKGNY	14	470	No SARS
AREGYNYFDT	10	684	No SARS

The most relevant attribute in the three datasets is the CDR3, an amino acid sequence represented as a text string. This sequence is critical because it contains key information about the specificity of the antibody toward the antigen, in this case the SARS-CoV-2 virus.

2.2 Transformation of sequences into numerical representation

Machine learning algorithms cannot directly process these amino acid form in their text strings form. Therefore, it is necessary to convert the CDR3 sequences into a numerical representation. We applied advanced natural language processing techniques that allow encoding these strings as numerical vectors and they can be used by machine learning algorithms (Patil et al., 2023). The most widely known technique in this area is Word2Vec, which has emerged as a common technique to represent words as vectors in high-dimensional space. Word2Vec assumes words that appear in comparable contexts share comparable vector representations, thus deriving semantic relationships from large text corpora (Mikolov et al., 2013). Word2Vec was designed to work with text, in which words have semantic meaning within the context of a sentence, and it follows a rigidly defined grammatical structure. However, it does not directly apply to CDR3 sequences. CDR3 are not formed according to the same principles as words in natural language. CDR3 sequences do not bear semantic meaning and do not occur in sentences in the linguistic sense. Applying Word2Vec to these sequences may not be capable of capturing the complexity and variability that characterize CDR3s. To overcome this limitation, Asgari and Mofrad (2015) developed a customized version for biological sequences called ProtVec. ProtVec is a neural network capable of representing amino acid sequences, such as CDR3 sequences, as high-dimensional numerical vectors. ProtVec was trained using the Swiss-Prot database which contains data on 7,027 unique protein families, which allowed the neural network to learn helpful numerical representations for amino acid sequences. Unlike words in natural language, antibodies amino acid sequences convey no grammatical structure and do not form sentences linguistically. ProtVec fills this gap by operating on amino acid sequences similar to how Word2Vec operates on sentences but adapted to the biological context. Instead of words, ProtVec uses trigrams, which are amino acid triplets to capture the

information in protein sequences. Fig. 1 illustrates this analogy, wherein sentences are amino acid sequences, and words are trigrams. ProtVec can discover patterns and relationships in CDR3 sequences, circumventing the constraints of Word2Vec.

Fig. 1. Natural language and the amino acid sequences similarity of antibodies.

The application of ProtVec to CDR3 sequences results in a numerical representation in the form of word embeddings which are 100-dimensional vectors and each vector represents a CDR3 sequence. The vectors are generated from random initial values and then altered when training the network. Despite this lack of direct interpretability, the generated word embeddings capture amino acid sequence richness and variability, a valuable tool for immunological data analysis like CDR3.

2.3 Data preprocessing

Before to build the learning models some preprocessing was conducted. The data was normalized with the Min-Max scaler technique. It scales the feature values to a range of 0 to 1, which is significant for those algorithms such as SVM and kNN sensitive to feature magnitudes. Normalization ensures that the features contribute equally to the classification without allowing those features with greater ranges to control the analysis. Additionally, we undersampled the data with the RandomUnderSampler technique. This technique equals the number of instances by reducing the majority class (SARS) over the minority class instances (NoSARS). Undersampling the data prevents the classification model from becoming biased towards the majority class, affecting the accuracy and model generalization.

Also, SeqLogo was used to identify the most frequent repeated n-grams at the beginning and end of the CDR3 sequences. SeqLogo is used as a bioinformatics tool to visualize the frequency of residues, amino acids, or nucleotides at specified positions in aligned sequences, enabling the identification of conserved patterns (Crooks et al., 2004). SeqLogo allows us to identify repeated n-grams in CDR3 sequences. Consistently occurring n-grams in all sequences were considered redundant as they lacked sufficient discriminative information for the classification problem. The bigrams at the beginning of the sequences and the 3-grams at the end were removed to avoid noise in the classification model. Additionally, it improves the model's ability to discriminate between SARS and NoSARS classes.

2.4 Dimensionality reduction

The word embeddings allow machine learning algorithms to process the CDR3s as numerical inputs. However, it is not feasible to interpret each of the 100 dimensions of the word embeddings generated by ProtVec straightforwardly. We applied the Principal Component Analysis technique, PCA (Holland, 2008) to reduce dimensionality and performed it in two datasets of tests using a cumulative variance of 90% and 95%. The intention is to evaluate whether a higher percentage of cumulative variance (95%) significantly improves the performance of classification models compared to 90% variance or whether the lower variance level is sufficient to maintain good performance with lower computational complexity.

The original CD1, CD10, and CD100 datasets, which initially contained 100 dimensions, were reduced to only 2 dimensions by PCA. This drastic reduction in the number of dimensions simplifies the data, facilitating the model training and reducing the risk of overfitting. The three reduced datasets were named CD1_PCA90, CD10_PCA90, and CD100_PCA90 to reflect that PCA was applied with 90% cumulative variance.

PCA with 95% of the cumulative variance applied to the CD1, CD10, and CD100 datasets reduced their dimensions significantly but with more components than the 90% cumulative variance approach. The reduced datasets are CD1_PCA95, CD10_PCA95, and CD100_PCA95, consisting of 10, 9, and 10 dimensions respectively.

We also applied the dimensionality reduction technique Uniform Manifold Approximation and Projection, UMAP (McInnes & Healy, 2018) Unlike PCA, UMAP does not have a direct, quantitative method, such as the cumulative variance value, to determine the optimal number of dimensions that capture the most relevant information from the dataset. Due to this particularity of UMAP, it was necessary to adopt an empirical approach to identify the appropriate number of dimensions,

allowing the classification models to achieve their highest performance. Multiple datasets with different dimensionalities were generated from the initial datasets CD1, CD10, and CD100. For this technique, the original 100 dimensions were reduced from 1 to 90 dimensions.

We proceed to build a classification model for each one of the 90 datasets generated. Then, we continue evaluating the performance of each model to identify the dimensions configuration offering the highest results in accuracy, sensitivity, and other relevant metrics. The datasets with best performance were selected for each initial dataset: CD1_UMAP consists of 5 dimensions, CD10_UMAP consists of 6 dimensions, and CD100_UMAP consists of 7 dimensions. The Table 2 shows the 12 resultant datasets.

Table 2. Datasets

Dataset	Dimensionality reduction technique	Dimensions
CD1	None	100
CD10	None	100
CD100	None	100
CD1_PCA90	PCA 90% cumulative variance	2
CD10_PCA90	PCA 90% cumulative variance	2
CD100_PCA90	PCA 90% cumulative variance	2
CD1_PCA95	PCA 95% cumulative variance	10
CD10_PCA95	PCA 95% cumulative variance	9
CD100_PCA95	PCA 95% cumulative variance	10
CD1_UMAP	UMAP	5
CD10_UMAP	UMAP	6
CD100_UMAP	UMAP	7

2.5 Hyperparameter configuration

Three classification techniques were selected, as they are known to work well in classifying word representations or word embeddings numerically: Support Vector Machines (Mammone et al., 2009), Random Forest (Rigatti, 2017) and k Nearest Neighbours (Kramer, 2013). We used Bayesian optimization for hyperparameter tuning. We used the Tree-structured version of the Parzen estimator to find good hyperparameter combinations in complex, high-dimensional search spaces (Yang & Shami, 2020). The optimized hyperparameters for each algorithm are shown in Tables 3, 4, and 5 for every dataset, divided into complete datasets and dimensionality-reduced datasets.

We split the dataset into 80% for training and 20% for testing to boost the strength and reliability of the results. This initial split guarantees that the model is trained on a representative sample of the data and reserves an independent dataset for determining its performance. In addition to this split, a 10-fold cross-validation was performed on the SVM, RF, and kNN classifiers.

Table 3. Hyperparameter configuration for models with datasets CD1, CD1_PCA90 and CD1_PCA95

Algorithm	Hyperparameter	CD1	CD1_PCA90	CD1_PCA95	UMAP
RF	*Criterion*	entropy	entropy	gini	gini
	Max_depth	9	26	23	29
	Max_features	59	3	24	17
	Min_samples_leaf	3	3	1	8
	Min_samples_split	2	8	2	11
	N_estimators	47	52	16	45
SVM	*C*	5	5	5	5
	kernel	poli	poli	poli	poli
	degree	3	3	3	3
kNN	*N_neighbors*	16	4	15	5

Table 4. Hyperparameter configuration for models with datasets CD10, CD10_PCA90 and CD10_PCA95

Algorithm	Hyperparameter	CD10	CD10_PCA90	CD10_PCA95	UMAP
RF	*Criterion*	entropy	entropy	gini	Gini
	Max_depth	40	9	15	37
	Max_features	59	44	60	29
	Min_samples_leaf	8	5	3	4
	Min_samples_split	6	9	9	9
	N_estimators	98	11	96	62
SVM	*C*	5	5	5	5
	kernel	poli	poli	poli	poli
	degree	3	3	3	3
kNN	*N_neighbors*	15	2	8	9

Table 5. Hyperparameter configuration for models with datasets CD100, CD100_PCA90 and CD100_PCA95

Algorithm	Hyperparameter	CD100	CD100_PCA90	CD100_PCA95	UMAP
RF	*Criterion*	entropy	entropy	gini	Gini
	Max_depth	40	7	41	37
	Max_features	59	50	45	29
	Min_samples_leaf	8	6	3	4
	Min_samples_split	6	5	7	9
	N_estimators	98	80	72	62
SVM	*C*	5	5	5	5
	kernel	poli	poli	poli	poli
	degree	3	3	3	3
kNN	*N_neighbors*	15	8	6	9

2.5 Datasets without dimensionality reduction

For the models without dimensionality reduction, we created nine classification models to establish a framework for comparing the performance of models incorporating dimensionality reduction techniques. We used three classification algorithms: Support Vector Machines, K-nearest neighbors, and Random Forest. We trained these classifiers using the datasets CD1, CD10, and CD100. Since we focused on the SARS label, we determined which models offer the best performance for classifying this class. However, the NoSARS class will also evaluate its performance between models to find a balance in both classes. We focused on the F1-score metric because it combines precision (the proportion of true positives among identified positives) and recall (the proportion of true positives among all true positives).

Table 6 shows the performance of the model performance trained using dataset CD1, which contains 395 records. The RF model stood out by obtaining a precision of 93% and a sensitivity of 76% in the SARS class, indicating a solid performance in identifying positive cases. In addition, this model achieved an F1-Score of 0.83, showing a balance between precision and sensitivity. This behavior is consistent in the NoSARS class, where RF also showed a sensitivity of 94% and a precision of 81%, achieving an F1-Score of 0.87. kNN showed a balanced performance, with a precision of 63%, a sensitivity of 0.73, and an F1-Score of 0.68 in the SARS class. In the NoSARS class, kNN had a precision of 0.71 and a sensitivity of 0.61, with an F1-Score of 0.66. Finally, the SVM model presented a precision of 0.72, a sensitivity of 0.55, and an F1-score of 0.62 in the SARS class. This result suggests that SVM may not be as effective in identifying SARS cases in comparison to RF and kNN.

Table 6. Performance of models with CD1 dataset

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.67	0.71	0.61	0.66
	SARS		0.63	0.73	0.68
RF	NoSARS	**0.86**	0.81	0.94	0.87
	SARS		**0.93**	**0.76**	**0.83**
SVM	NoSARS	0.68	0.66	0.81	0.73
	SARS		0.72	0.55	0.62

Table 7 shows the performance of the three classification models trained using the CD10 dataset containing 3,950 records. As with the CD1 dataset, the RF model excelled, achieving a precision of 0.92 and a sensitivity of 0.71 in the SARS class. These results indicate that the RF algorithm effectively identifies positive cases in higher data volume scenarios. In addition, RF achieved an F1-Score of 0.80, underscoring its balance between precision and sensitivity, ensuring good performance in SARS-CoV-2 specific antibody classification. The RF performance in the NoSARS class obtained a sensitivity of 0.95, a precision of 0.79, and an F1-Score of 0.86. This consistent behavior suggests that RF is effective in detecting both positive cases (SARS) and negative cases (NoSARS). The kNN model obtained a precision of 67% and a sensitivity of 71%, along with an F1-Score of 0.68 in the SARS class. In the NoSARS class, kNN had an F1-Score of 0.72. SVM showed a lower performance than RF. In the SARS class, SVM achieved a precision of 0.80, a sensitivity of 0.52, and an F1-Score of 0.63. In the NoSARS class, SVM achieved an F1-Score of 0.77, suggesting that SVM is more reliable in detecting negative cases than kNN, albeit with lower performance in the class of interest, SARS.

Table 7. Performance of models with CD10 dataset

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.70	0.72	0.71	0.72
	SARS		0.67	0.71	0.68
RF	NoSARS	**0.84**	0.79	0.95	0.86
	SARS		**0.92**	**0.71**	**0.80**
SVM	NoSARS	0.68	0.68	0.89	0.77
	SARS		0.80	0.52	0.63

Table 8 shows the three-classification model's performance using the CD100 dataset containing 39,500 records. As in the previous cases, the RF model again excelled, achieving the highest precision (0.93), a sensitivity of 0.67, and an F1-Score of 0.78 in the SARS class. In the NoSARS class, RF obtained an F1-Score of 0.83. These results indicate that RF continues to be the most effective model, even for this dataset with the most records, demonstrating consistently balanced performance in both classes. In the SARS class, kNN obtained a precision of 0.87, a sensitivity of 0.63, and an F1-Score of 0.73. In the NoSARS class, kNN achieved an F1-Score of 0.80. SVM obtained a precision of 0.75, a sensitivity of 0.52, and an F1-Score of 0.68. In the NoSARS class, the F1-Score was 0.81, higher than the kNN classifier.

Table 8. Performance of models with CD100 dataset

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.77	0.71	0.90	0.80
	SARS		0.87	0.63	0.73
RF	NoSARS	**0.81**	0.75	0.95	0.83
	SARS		**0.93**	**0.67**	**0.78**
SVM	NoSARS	0.76	0.88	0.75	0.81
	SARS		0.75	0.52	0.68

2.6 Datasets with 90% PCA

We applied the Principal Component Analysis technique for the models with dimensionality reduction and performed it in two datasets of tests using a cumulative variance of 90% and 95%. The intention is to evaluate whether a higher percentage of cumulative variance (95%) significantly improves the performance of classification models compared to 90% variance or whether the lower variance level is sufficient to maintain good performance with lower computational complexity.

The original CD1, CD10, and CD100 datasets, which initially contained 100 dimensions, were reduced to only 2 dimensions by PCA. This drastic reduction in the number of dimensions simplifies the data, facilitating the model training and reducing the risk of overfitting. The three reduced datasets were named CD1_PCA90, CD10_PCA90, and CD100_PCA90 to reflect that PCA was applied with 90% cumulative variance.

Table 9 shows the results of the models trained with the CD1_PCA90 dataset, which was reduced to 2 dimensions and contains 395 records. The kNN and RF models particularly excelled in the SARS class, both classifiers achieving a precision of 0.76, a sensitivity of 0.79, and an F1-Score of 0.78. These results indicate that kNN and RF are highly effective and similar in identifying SARS-positive cases, maintaining an appropriate balance between precision and sensitivity.

The SVM model achieved a precision of 0.75, a sensitivity of 0.71, and an F1-Score of 0.73. Although these results are competitive, SVM fails to match the performance of kNN and RF in identifying SARS-positive cases. In the NoSARS class, however, SVM achieved an F1-Score of 0.84. These results suggest that SVM enhances identifying negative cases, showing outstanding performance in this class. However, given that the SARS class is the most relevant in this research, SVM's lower performance in detecting positive cases positions it as the model with the least favorable performance compared to the kNN and RF models.

Table 9. Performance of models with dataset CD1_PCA90

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.78	0.80	0.78	0.79
	SARS		0.76	0.79	0.78
RF	NoSARS	0.78	0.80	0.78	0.79
	SARS		0.76	0.79	0.78
SVM	**NoSARS**	**0.80**	**0.95**	**0.75**	**0.84**
	SARS		0.75	0.71	0.73

Table 10 shows the performance of the three classification models trained using the CD10_PCA90 dataset, containing 3,950 records. The kNN model shows the best performance in the SARS class, achieving a precision of 0.86, a sensitivity of 0.61, and an F1-Score of 0.72. In contrast, the RF model excelled in the SARS class, achieving a high precision of 0.97, a sensitivity of 0.55, and an F1-Score of 0.71. The SVM model had the lowest performance in the SARS class, with a precision of 75%, a sensitivity of 57%, and an F1-Score of 0.65. These results position SVM as the least effective model in SARS case detection. In the NoSARS class, both RF and SVM had the highest F1-Score values. In this dataset, the outstanding model is kNN, as it has a good balance in its SARS class metrics and a higher performance in the NoSARS class.

Table 10. Performance of models with dataset CD10_PCA90

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.77	0.73	0.92	0.81
	SARS		0.86	0.61	0.72
RF	NoSARS	**0.78**	0.72	0.98	0.83
	SARS		**0.97**	**0.55**	**0.71**
SVM	NoSARS	0.76	0.90	0.75	0.82
	SARS		0.75	0.57	0.65

Table 11 shows the performance of the classification models trained on the CD100_PCA90 dataset, which contains 39,500 records. The kNN model shows the best performance, standing out for its balance in the SARS class, as it achieved a precision of 0.86 and a sensitivity of 0.62, with an F1-Score of 0.72. Although its accuracy is lower than the RF model, kNN achieves a better balance between precision and sensitivity. The RF model achieved the highest precision among the three models, with 0.96 in the SARS class. However, its sensitivity was only 0.57, reflecting a significant imbalance between precision and sensitivity. The F1-Score obtained was 0.71. The SVM model obtained a precision of 0.75 and a sensitivity of 0.66. For this dataset with the highest records, the model kNN offers a good balance between precision and sensitivity in the SARS class.

Table 11. Performance of models with dataset CD100_PCA90

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.76	0.71	0.90	0.79
	SARS		0.86	0.62	0.72
RF	NoSARS	**0.77**	0.70	0.97	0.81
	SARS		**0.96**	**0.57**	**0.71**
SVM	NoSARS	0.77	0.95	0.75	0.84
	SARS		0.75	0.66	0.70

2.7 Datasets with 95% PCA

We evaluate the classification algorithm's performance when trained with datasets with 95% of the cumulative variance. With this configuration, the CD1, CD10, and CD100 datasets reduced their dimensions significantly but with more components than the 90% cumulative variance approach. The constructed datasets are CD1_PCA95, CD10_PCA95, and CD100_PCA95, consisting of 10, 9, and 10 dimensions respectively.

Table 12 shows the classification model's performance trained with the CD1_PCA95 ensemble, which has 10 dimensions out of the initial 100. In the SARS class, RF achieved a precision of 0.87, a sensitivity of 0.82, and an F1-Score of 0.84. The kNN achieved a high precision of 0.96, a sensitivity of 0.70, and an F1-Score of 0.81. Its overall performance in the SARS class is slightly lower than the RF model. In the SARS class, SVM achieved a precision of 0.72, a sensitivity of 0.74, and an F1-Score of 0.73. In the NoSARS class, SVM achieved an F1-Score of 0.84.

Table 12. Model performance on the CD1_PCA95 dataset

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.84	0.78	0.97	0.86
	SARS		**0.96**	**0.70**	**0.81**
RF	NoSARS	0.77	0.84	0.89	0.86
	SARS		0.87	0.82	0.84
SVM	NoSARS	0.80	0.92	0.77	0.84
	SARS		0.72	0.74	0.73

Table 13 shows the model's performance on the CD10_PCA95 dataset. In the SARS class, RF achieved a precision of 94%, a sensitivity of 65%, and an F1-Score of 0.77. The kNN model also showed a competitive performance, with a precision of 0.86, a sensitivity of 0.60, and an F1-Score of 0.71. However, in the NoSARS class, the F1-Score reached 0.81. Finally, the SVM model showed the lowest performance among the three. In the SARS class, SVM achieved a precision of 0.77, a sensitivity of 0.58, and an F1-Score of 0.66. In the NoSARS class, although SVM obtained a precision of 0.92, its F1-Score was 0.82.

Table 13. Performance of models with dataset CD10_PCA95

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.77	0.73	0.91	0.81
	SARS		0.86	0.60	0.71
RF	NoSARS	**0.82**	0.76	0.96	0.85
	SARS		**0.94**	**0.65**	**0.77**
SVM	NoSARS	0.77	0.92	0.74	0.82
	SARS		0.77	0.58	0.66

Table 14 shows the model's performance in the CD100_PCA95 dataset. RF achieved a precision of 0.94, a sensitivity of 0.68, and an F1-Score of 0.79. In the NoSARS class, it obtained an F1-Score of 0.84. The second highest-performing model is kNN. In the SARS class, it obtained a precision of 0.88, a sensitivity of 0.65, and an F1-Score of 0.74. The NoSARS class achieved an F1-Score of 0.81. In the SARS class, the SVM model achieved a precision of 0.71 and a sensitivity of 0.73, achieving an F1-Score of 0.72. In the NoSARS class, SVM obtained a precision of 91% and an F1-Score of 0.83, which is higher than the obtained by the kNN model but lower than SVM.

Table 14. Performance of models with dataset CD100_PCA95

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.78	0.72	0.91	0.81
	SARS		0.88	0.65	0.74
RF	NoSARS	**0.82**	0.75	0.96	0.84
	SARS		**0.94**	**0.68**	**0.79**
SVM	NoSARS	0.77	0.91	0.76	0.83
	SARS		0.71	0.73	0.72

2.8 Datasets with UMAP

This section describes the application of the UMAP dimensionality reduction technique. Unlike PCA, UMAP does not have a direct, quantitative method, such as the cumulative variance value, to determine the optimal number of dimensions that capture the most relevant information from the dataset. Due to this particularity of UMAP, it was necessary to adopt an empirical approach to identify the appropriate number of dimensions, allowing the classification models to achieve their highest performance. Multiple datasets with different dimensionalities were generated from the initial datasets CD1, CD10, and CD100. For this technique, the original 100 dimensions were reduced from 1 to 90 dimensions.

We proceed to build a classification model for each one of the 90 datasets generated. Then, we continue evaluating the performance of each model to identify the dimensions configuration offering the highest results in precision, sensitivity, and other relevant metrics. The highest-performing datasets were selected for each initial dataset. CD1_UMAP consists of 5 dimensions, CD10_UMAP consists of 6 dimensions, and CD100_UMAP consists of 7 dimensions. The detailed results of this selection are presented in Tables 15, 16, and 17, which summarize the performance of the three classification models on the three most optimized datasets. This approach allowed for maximizing the effectiveness of UMAP in dimensionality reduction, achieving a balance between model simplicity and retention of critical information, resulting in better overall performance of the classification models.

Table 15 shows the performance of the classification models trained with the CD1_UMAP dataset. The SVM model in the SARS class reached a precision of 0.52, a sensitivity of 0.85, and an F1-Score of 0.64. The F1-Score of the NoSARS class reached 0.53. The RF model achieved a sensitivity of 0.76 and an F1-Score of 0.63 despite its low precision of 0.54. These results suggest that RF is effective in identifying most SARS-positive cases, although it has a significant false positive rate, reflected in its moderate precision. On the other hand, the kNN model performed the lowest in the SARS class, with a precision of 52% and a sensitivity of 65%, resulting in an F1-Score of 0.58. As for the NoSARS class, the kNN model obtained a precision of 68%, a sensitivity of 0.56, and an F1-Score of 0.61.

Table 15. Performance of models with dataset CD1_UMAP

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.59	0.68	0.56	0.61
	SARS		0.52	0.65	0.58
RF	NoSARS	0.62	0.74	0.51	0.61
	SARS		0.54	0.76	0.63
SVM	**NoSARS**	**0.59**	**0.78**	**0.40**	**0.53**
	SARS		0.52	0.85	0.64

Table 16 shows the performance of the classification models using the CD10_UMAP dataset. The RF model slightly outperforms the others in the SARS class, achieving a precision of 0.70, a sensitivity of 0.78, and an F1-Score of 0.73. This suggests that RF offers a balance between the ability to identify positive SARS cases and minimize false positives. The kNN model improved over its performance in the previous model, with a precision of 69%, a sensitivity of 71%, and an F1-Score of 0.70. Although its performance improved compared to the model trained on the CD1 dataset, its ability to correctly identify positive cases is limited, placing it behind the RF model in terms of overall performance for the SARS class. In the SARS class, the SVM model achieved a precision of 0.57 and a sensitivity of 0.83, with an F1 score of 0.68. While its sensitivity was the highest of the three models, its precision was also the lowest, making it the worst-performing model on average.

Table 16. Performance of models with dataset CD10_UMAP

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.66	0.63	0.61	0.62
	SARS		0.69	0.71	0.70
RF	NoSARS	**0.69**	0.68	0.58	0.63
	SARS		**0.70**	**0.78**	**0.73**
SVM	NoSARS	0.56	0.51	0.22	0.31
	SARS		0.57	0.83	0.68

Table 17 shows the performance of the classification models on the CD100_UMAP dataset. The RF model achieved a precision of 0.72, a sensitivity of 0.75, and an F1-score of 0.73. In the SARS class, the SVM obtained a precision of 0.71, a sensitivity of 0.73, and an F1-score of 0.72. Finally, the SVM model showed the lowest performance in the SARS class, achieving a precision of 0.59, but with a significantly higher sensitivity than the other models, 85%, and an F1-score of 0.69. However, in the NoSARS class, the model only reached a sensitivity of 0.24 and an F1-Score of 0.34.

Table 17. Performance of models with dataset CD100_UMAP

Model	Class	Accuracy	Precision	Sensitivity	F1-score
kNN	NoSARS	0.68	0.64	0.61	0.63
	SARS		0.71	0.73	0.72
RF	NoSARS	**0.69**	0.66	0.62	0.64
	SARS		**0.72**	**0.75**	**0.73**
SVM	NoSARS	0.58	0.55	0.24	0.34
	SARS		0.59	0.85	0.69

2.9 Evaluation of the predictive models

The models built on the UMAP-reduced data failed to exceed the overall precision of the PCA-trained models and the initial models in their full dimensions. However, UMAP outperformed all the rest of the models in the sensitivity metric for the SARS class. Despite the achievement, the precision of the NoSARS class was lower than the rest of the models, revealing that UMAP does not perform well to possess a balance among the measures of both classes. Though UMAP can be a better choice in some instances where accurate SARS detection is the class with more relevance, its general use in this context could not recommended due to the class performance imbalance. Tables 18 and 19 shows the performance of all models in the SARS class alone, where it is apparent that for most metrics, the best performers were the PCA-reduced models with 90% and 95% cumulative variance. PCA, specifically the 95% cumulative variance setup, is the best dimensionality reduction technique for identifying SARS-CoV-2 antibody sequence classification.

Table 18. performance on accuracy and precision of all models in the SARS class

Dataset	Model	SRD	PCA90	PCA95	UMAP	SRD	PCA90	PCA95	UMAP
		Accuracy				Precision			
CD1	kNN	0.67	0.78	0.84	0.59	0.63	0.76	0.96	0.52
	RF	0.86	0.78	0.86	0.62	0.93	0.76	0.87	0.54
	SVM	0.68	0.80	0.80	0.59	0.72	0.75	0.72	0.52
CD10	kNN	0.70	0.77	0.77	0.66	0.67	0.86	0.86	0.69
	RF	0.84	0.78	0.82	0.69	0.92	0.97	0.94	0.70
	SVM	0.72	0.76	0.77	0.56	0.80	0.75	0.77	0.57
CD100	kNN	0.77	0.76	0.78	0.68	0.87	0.86	0.88	0.71
	RF	0.81	0.77	0.82	0.69	0.93	0.96	0.94	0.72
	SVM	0.76	0.77	0.77	0.58	0.75	0.75	0.71	0.59

Table 19. Performance on sensitivity and F1-score of all models in the SARS class

Dataset	Model	SRD	PCA90	PCA95	UMAP	SRD	PCA90	PCA95	UMAP
		Sensitivity				F1-score			
CD1	kNN	0.73	0.79	0.70	0.65	0.68	0.78	0.81	0.58
	RF	0.76	0.79	0.82	0.76	0.83	0.78	0.84	0.63
	SVM	0.55	0.71	0.74	0.85	0.62	0.73	0.73	0.64
CD10	kNN	0.71	0.61	0.60	0.71	0.68	0.72	0.71	0.70
	RF	0.71	0.55	0.65	0.78	0.80	0.71	0.77	0.73
	SVM	0.52	0.57	580.	0.83	0.63	0.65	0.66	0.68
CD100	kNN	0.63	0.62	0.65	0.73	0.73	0.72	0.74	0.72
	RF	0.67	0.57	0.68	0.75	0.78	0.71	0.79	0.73
	SVM	0.52	0.66	0.73	0.85	0.68	0.70	0.72	0.69

3 Results

The models CD1, CD10, and CD100 (without dimensionality reduction) serve as a basis to compare the performance of models built with dimensionality reduced datasets. The PCA models were 90% and 95% of cumulative variance retention. These models allow us to evaluate the effect of the percentage of variance retained on the performance of the classifiers. The UMAP models produce multiple copies of the original datasets to see how UMAP affects performance.

Table 20 datasets out the highest-performing models accordingly to F1-score. The evaluation metric was chosen for its ability to provide an overall view between precision and sensitivity, offering a generalization of how well each model performs. A further observation regarding the F1-score in the trained model is that it decreases as the number of records in the datasets increases, from 0.83 in CD to 0.78 in CD100. It is significant and could indicate that the greater the number of instances, the greater the probability there were noisy or mislabeled instances, detracting from model performance.

Table 20. Better models without dimensionality reduction accordingly to F1-score

Dataset	Model	Label	Accuracy	Precision	Sensitivity	F1-score
CD1	RF	NoSARS	0.86	0.81	0.94	0.87
		SARS		0.93	0.76	**0.83**
CD10	RF	NoSARS	0.84	0.79	0.95	0.86
		SARS		0.92	0.71	0.80
CD100	RF	NoSARS	0.81	0.75	0.95	0.83
		SARS		0.93	0.67	**0.78**

The models built by the RF classifier are very close to the kNN models. However, increasing the number of records may have introduced noisy or mislabeled data, affecting the performance of the classifier. Although the top models built using PCA and 90% variance were lower in performance, they showed very robust performance with only two dimensions. Whereas the performance of the best-performing models dipped slightly, the overall performance of the rest of the models improved, which suggest that dimensionality reduction by PCA not only minimizes the complexity of the models but also evens out the performance of different algorithms and datasets. Table 21 shows only the best performance of each dataset.

Table 21. Better models with 90% PCA dimensionality reduction accordingly to F1-score

Dataset	Model	Label	Accuracy	Precision	Sensitivity	F1-score
CD1	kNN	NoSARS	0.78	0.80	0.78	0.79
		SARS		0.76	0.79	0.78
CD10	kNN	NoSARS	0.77	0.73	0.92	0.81
		SARS		0.86	0.61	0.72
CD100	kNN	NoSARS	0.76	0.71	0.90	0.79
		SARS		0.86	0.62	0.72

The SVM, kNN, and RF models were trained on the datasets CD1_PCA95, CD10_PCA95, and CD100_PCA95, whose dimensions had been reduced from 100 dimensions to 10, 9, and 10, respectively, with 95% of the cumulative variance retained (The dimensionality reduction process is described in section 2.4). The 95% variance models enhance the results by gaining only 5% in cumulative variance, compared with the 90% variance models. The model Random Forest with CD100_95 dataset was still better than the top model in the category trained on 90% variance (kNN with CD1_PCA90). The use of 95% variance provides an increase in the classification models. On the other hand, compared to the performance of the models without dimensionality reduction, which train their models on the original 100 dimensions, the models with 95% variance outperform the models without dimensionality reduction. These results highlight that more variance preservation is necessary for dimensionality reduction. Table 22 shows only the best performance of each dataset.

The SVM, kNN, and RF models were trained on the datasets CD1_UMAP, CD10_UMAP, and CD100_UMAP, whose dimensions had been reduced to 5, 6, and 7, respectively. As compared to the rest of the cases where the F1-score reduced as the number of datasets increased, in this case, the F1-score increased. However, the UMAP-reduced models have a decreased performance in the NoSARS class. The performance of the two classes is not balanced. A class-imbalanced model may be less reliable in identifying positive and negative cases. The models built with the UMAP-scaled datasets performed lower than the

models built without dimensionality reduction or those reduced by PCA. Table 23 shows only the best performance of each dataset.

Table 22. Better models with 95% PCA dimensionality reduction accordingly to F1-score

Dataset	Model	Label	Accuracy	Precision	Sensitivity	F1-score
CD1	RF	NoSARS	0.86	0.84	0.89	0.86
		SARS		0.87	0.82	0.84
CD10	RF	NoSARS	0.82	0.76	0.96	0.85
		SARS		0.94	0.65	0.77
CD100	RF	NoSARS	0.82	0.75	0.96	0.84
		SARS		0.94	0.68	0.79

Table 23. Better models with UMAP dimensionality reduction accordingly to F1-score

Dataset	Model	Label	Accuracy	Precision	Sensitivity	F1-score
CD1	SVM	NoSARS	0.59	0.78	0.40	0.53
		SARS		0.52	0.85	0.64
CD10	RF	NoSARS	0.69	0.68	0.58	0.63
		SARS		0.70	0.78	0.73
CD100	RF	NoSARS	0.69	0.66	0.62	0.64
		SARS		0.72	0.75	0.73

Table 24 shows the leading models for each dataset based on F1-score metric on SARS class. These figures are extracted from Tables 6 to 17. We labeled the models with three acronyms: the classifier, the dataset, and the dimensionality reduction technique applied to the ensemble. In most cases, the best-performing classification algorithm was Random Forest, while the second best was K-Nearest Neighbors. From the dimensionality reduction methods, PCA was the most successful at preserving data representation while drastically lowering the number of dimensions. PCA not only reduces data but can also improve model performance. PCA was the best method for dimensionality reduction, as it works well with both 90% and 95% cumulative variance. Retaining 90% of the variance brings the dataset down to two dimensions, which is easy to use for exploratory analyses.

Table 24. Best models accordingly to F1-Score

Classifier	Dataset	Reduction technique	F1-Score
RF	CD1_PCA95	PCA95	0.84
RF	CD1	SRD	0.83
RF	CD10	SRD	0.80
RF	CD100_PCA95	PCA95	0.79
kNN	CD1_PCA90	PCA90	0.78
RF	CD100	SRD	0.78
RF	CD10_PCA95	PCA95	0.77
RF	CD10_UMAP	UMAP	0.73
RF	CD100_UMAP	UMAP	0.73
kNN	CD10_PCA90	PCA90	0.72
kNN	CD100_PCA90	PCA90	0.72
SVM	CD1_UMAP	UMAP	0.64

4 Conclusions

Three machine learning models used in this current work (Random Forest, K-Nearest Neighbors, and Support Vector Machines) were optimal for classifying correctly antibody data for SARS-CoV-2. The application of 95% cumulative variance PCA was the best dimensionality reduction technique, as it significantly reduced the complexity of the dataset at a very high level while keeping its classification ability intact.

The best model in this study was Random Forest on the minimized CD1 dataset with PCA to retain 95% of the overall variance (RF_CD1_PCA95). This model achieved an accuracy of 86%, precision of 87%, sensitivity of 82%, and F1-score of 84%, outperforming others. Random Forest addresses the initial problem of efficiently classifying antibody sequences in a high-dimensional environment. Moreover, PCA provided the best results when used for dimensionality reduction, which means that for this type of data, numerical representations of biological sequences, PCA is the technique that reduces their dimensions most efficiently and preserves the underlying semantic relationships of the data.

References

Adjuik, T. A., & Ananey-Obiri, D. (2022). Word2vec neural model-based technique to generate protein vectors for combating COVID-19: A machine learning approach. *International Journal of Information Technology*, 14(7), 3291-3299.

Arowolo, M. O., Adebiyi, M. O., Aremu, C., & Adebiyi, A. A. (2021). A survey of dimensionality reduction and classification methods for RNA-Seq data on malaria vector. *Journal of Big Data*, 8(1), 50.

Asgari, E., & Mofrad, M. R. K. (2015). Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLOS ONE*, 10(11), e0141287.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., & Linial, M. (2022). ProteinBERT: A universal Deep-learning model of protein sequence and function. *Bioinformatics*, 38(8), 2102-2110.

Crooks, G. E., Hon, G., Chandonia, J.-M., & Brenner, S. E. (2004). WebLogo: A Sequence Logo Generator: Figure 1. *Genome Research*, 14(6), 1188-1190.

Enríquez, M., Naranjo, S., Amaro, I., & Camacho, F. (2021). Dimensionality Reduction using PCA and CUR Algorithm for Data on COVID-19 Tests. *Artificial Intelligence, Computer and Software Engineering Advances*, 1326, 121-134.

Holland, S. M. (2008). Principal components analysis (PCA). Department of Geology, University of Georgia, Athens, GA, 30602, 2501.

Khalilian, S., Nasr Isfahani, M., Moti, Z., Baloochestani, A., Chavosh, A., & Hemmatian, Z. (2022). *A Deep Dimensionality Reduction method based on Variational Autoencoder for Antobody Complementarity Determining Region Sequence Analysis*. 116-105.

Kramer, O. (2013). K-Nearest Neighbors. O. Kramer, *Dimensionality Reduction with Unsupervised Nearest Neighbors*, 51, 13-23.

Mammone, A., Turchi, M., & Cristianini, N. (2009). Support vector machines. *WIREs Computational Statistics*, 1(3), 283-289.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ARXIV*.

McInnes, L, Healy, J. (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, ArXiv e-prints 1802.03426.

OAS (october, 2024). Search OAS: Unpaired sequences. OAS Observed Antibody Space. https://opig.stats.ox.ac.uk/webapps/oas/oas_unpaired

Patil, R., Boit, S., Gudivada, V., & Nandigam, J. (2023). A Survey of Text Representation and Embedding Techniques in NLP. *IEEE Access*, 11, 36120-36146.

Rigatti, S. J. (2017). Random Forest. *Journal of Insurance Medicine*, 47(1), 31-39.

Rustam, Gunawan, A. Y., & Kresnowati, M. T. A. P. (2022). Data dimensionality reduction technique for clustering problem of metabolomics data. *Heliyon*, 8(6), e09715.

Stafford, P., Wrapp, D., & Johnston, S. A. (2016). General Assessment of Humoral Activity in Healthy Humans. *Molecular & Cellular Proteomics*, 15(5), 1610-1621.

Wang, Y. (2019). Single Training Dimensions Selection for Word Embedding with PCA. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* 3595-3600.

Weber, A., Pélissier, A., & Rodríguez Martínez, M. (2024). T-cell receptor binding prediction: A machine learning revolution. ImmunoInformatics, 15, 100040.

Xu, J. L., & Davis, M. M. (2000). Diversity in the CDR3 Region of VH Is Sufficient for Most Antibody Specificities. *Immunity*, 13(1), 37-45.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295-316.

Zaki, M. J., Wang, J. T. L., & Toivonen, H. T. T. (2003). Data mining in bioinformatics: Report on BIOKDD'03. *ACM SIGKDD Explorations Newsletter*, 5(2), 198-199.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Journal of Applied Science and Technology Trends*, 1(2), 56-70.