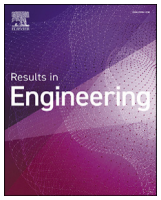
Results in Engineering 24 (2024) 102878

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering



Research paper

DeepQSP: Identification of Quorum Sensing Peptides Through Neural Network Model ☆

Md. Ashikur Rahman [a], Md. Mamun Ali [a,b], Kawsar Ahmed [c,d,e,*], Imran Mahmud [a], Francis M. Bui [c], Li Chen [c], Santosh Kumar [f], Mohammad Ali Moni [g,h]

[a] Department of Software Engineering, Daffodil International University, Daffodil Smart City, Birulia, Dhaka-1216, Bangladesh
[b] Division of Biomedical Engineering, University of Saskatchewan, 57 Campus Drive, Saskatoon, SK S7N 5A9, Canada
[c] Department of Electrical and Computer Engineering, University of Saskatchewan, 57 Campus Drive, Saskatoon, SK S7N 5A9, Canada
[d] Bio-photomatiχ, Information and Communication Technology, Mawlana Bhashani Science and Technology University, Santosh, Tangail 1902, Bangladesh
[e] Health Informatics Research Lab, Department of Computer Science and Engineering, Daffodil International University, Daffodil Smart City, Birulia, Dhaka-1216, Bangladesh
[f] Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh 522302, India
[g] AI & Digital Health Technology, Artificial Intelligence & Cyber Future Institute, Charles Stuart University, Bathurst, NSW 2795, Australia
[h] AI & Digital Health Technology, Rural Health Research Institute, Charles Stuart University, Orange, NSW 2800, Australia

ARTICLE INFO

Keywords:
Quorum sensing peptide
Convolutional neural network
Latent semantic analysis
Microorganisms
Bioengineering

ABSTRACT

Quorum Sensing Peptides (QSP) are small molecules crucial for microbial communication, enabling bacterial populations to coordinate behaviors such as biofilm formation and virulence. The identification of QSP is vital for understanding these biological processes. While existing clinical and lab-based methods are available, they can be costly and time-consuming. This study introduces DeepQSP, a novel technique for QSP identification, which combines Latent Semantic Analysis (LSA), a word embedding feature extraction method, with classical amino acid-based extraction Pseudo Amino Acid Composition (PAAC), and a convolutional neural network (CNN) classifier. The DeepQSP model was evaluated using a dataset of 440 peptide sequences, achieving impressive performance metrics: 0.9697 accuracy, 0.9655 sensitivity, 0.9730 specificity, and a Matthews correlation coefficient (MCC) of 0.9385. The LSA combined with PAAC improves peptide sequence representation, while the CNN effectively captures complex patterns, leading to accurate QSP identification. These quantified results demonstrate the effectiveness of the DeepQSP method, offering a powerful tool for advancing the study of microbial interaction and quorum sensing. The enhanced identification of QSPs is critical for microbiology and bioengineering, aiding in the understanding of cell-to-cell communication in microorganisms.

1. Introduction

Quorum sensing (QS) is a common biological process in microorganisms, facilitating bacterial cells to communicate and synchronize gene expression through the exchange of chemical signaling molecules [1,2]. Quorum Sensing Peptides (QSP) are small signaling molecules produced and released by bacteria to facilitate communication among microbial community members [3,4]. These peptides play a vital role in governing various bacterial behaviors, including forming biofilms, expressing virulence factors, and regulating gene activity [5]. Quorum sensing enables bacteria to synchronize their actions based on the population density of their peers by detecting the concentration of QSP in their environment. Once the QSP concentration reaches a critical threshold, it triggers specific responses in the bacterial population, allowing them to function as a cohesive group [6]. Quorum sensing is an intriguing mechanism that empowers bacteria to adapt to their surroundings and respond collectively to changing conditions. Researchers have shown keen interest in studying QSP for potential applications in diverse fields, such as medicine, biotechnology, the food industry, agriculture, and environmental science [7–9].

☆ This work was supported in part by funding from the Natural Sciences and Engineering Research Council of Canada (NSERC).

* Corresponding author at: Department of Electrical and Computer Engineering, University of Saskatchewan, 57 Campus Drive, Saskatoon, SK S7H 3W4, Canada.
E-mail addresses: ashikur35-562@diu.edu.bd (M. Ashikur Rahman), m.ali@usask.ca, mamun.ali@ieee.org (M. Mamun Ali), k.ahmed.bd@ieee.org, kawsar.ict@mbstu.ac.bd, k.ahmed@usask.ca (K. Ahmed), imranmahmud@daffodilvarsity.edu.bd (I. Mahmud), francis.bui@usask.ca (F.M. Bui), lic900@usask.ca (L. Chen), santosh@kluniversity.in (S. Kumar), mmoni@csu.edu.au (M.A. Moni).

https://doi.org/10.1016/j.rineng.2024.102878
Received 25 May 2024; Received in revised form 30 August 2024; Accepted 9 September 2024
Available online 13 September 2024

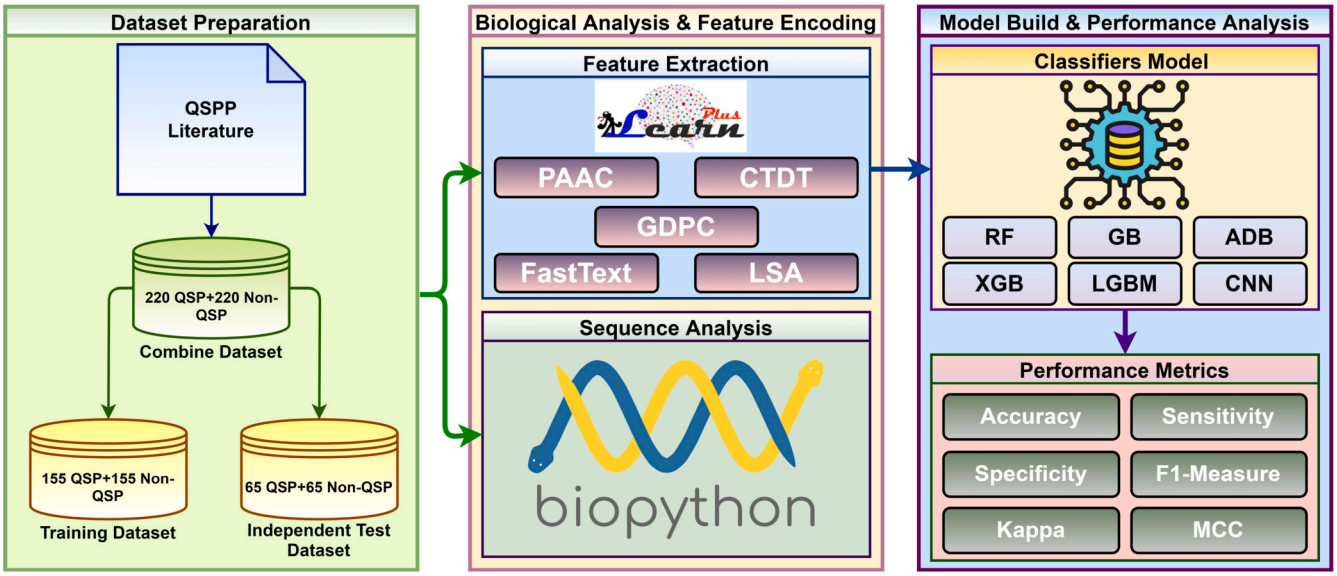


Fig. 1. Research methodology of the study and structural architecture for developing the proposed model, which has been named after DeepQSP.

QSP have the potential to enhance crop health through eco-friendly approaches for managing plant diseases by disrupting quorum sensing in plant pathogens. Moreover, these factors find application in regulating bacterial behavior in food production, effectively preventing the unwanted formation of biofilms and thereby safeguarding the safety and quality of food products [10,11]. Researchers are exploring the use of QSP for the development of innovative antibacterial therapies. These peptides offer a means to interfere with quorum sensing, potentially curbing bacterial pathogens' virulence without outright destroying them. This approach has the potential to mitigate the emergence of antibiotic resistance, a critical concern in modern medicine [12]. Moreover, QSP finds practical utility in the realms of bioprocessing and bioremediation. These factors boost the efficiency of microbial fermentation processes and provide a means to govern the conduct of genetically engineered bacteria, whether for the targeted synthesis of particular compounds or the breakdown of environmental pollutants [13]. QSP is a cornerstone in a broad array of scientific fields, from food and medicine to agriculture and environmental science. As a result, the accurate, reliable, and cost-effective detection of QSP holds profound importance, contributing to scientific progress and the overall welfare of humankind. Conventional laboratory experiments for QSP or peptide sequence identification are both time-consuming and expensive. In this case, the application of machine learning (ML) and computational biology becomes instrumental in addressing this challenge. ML and other computational models can offer more precise and reliable predictions for QSP and protein or peptide sequences [14,15].

In recent years, researchers have undertaken numerous studies to identify QSP using ML and computational models. For instance, Rajput et al. in 2015 introduced the first computational model for QSP prediction, known as QSPpred. The authors used amino acid composition (AAC), amino acid residue position (ACRP), motif identification, physicochemical properties (PCP), and Grand average of hydropathy (GRAVY) for sequence encoding. A support vector machine (SVM) was proposed as the classifier, exhibiting noteworthy performance with the highest accuracy and Mathew's correlation coefficient (MCC) of 93.00% and 0.8600, respectively [16]. However, the model's reliance on multiple encoding methods makes it computationally intensive. Additionally, the SVM classifier, while effective, may not capture complex patterns in the data as effectively as more advanced machine learning techniques like deep learning. A model for predicting QSP, which employed Random Forest (RF) and feature representation learning, termed QSPred-FL. QSPred-FL achieved an accuracy of 94.30% and a MCC of 0.8850 [17]. Despite these improvements, RF models can struggle with high-dimensional data and may not generalize well to unseen data. The feature representation learning technique used in this model, while beneficial, still relies on traditional encoding methods that may not fully capture the complexity of QSP sequences. Another SVM classifier-based model called iQSP predictor with a PCP-based encoding method was proposed by Charoenkwan et al. (2019), with maximum accuracy of 93.00% and MCC of 0.8600 [18]. While the use of PCP-based encoding is a step forward, the model's performance plateaued compared to earlier efforts. The reliance on SVM may limit the ability to learn more complex patterns in QSP sequences, and the encoding methods used may not capture all relevant features. In 2022, Sivaramakrishnan et al. introduced a stacking-based ensemble learning model, known as EnsembleQS, for identifying QSP. Feature encoding methods such as AAC, Dipeptide Composition (DPC), Dipeptide Deviation from Expected Mean (DEM), and Tripeptide Composition (TPC) were utilized. Notably, the EnsembleQS predictor achieved an impressive accuracy of 93.40% and an MCC of 0.9100 [19]. The ensemble approach enhances predictive performance by combining multiple models, but it also increases computational complexity and may suffer from overfitting. Additionally, the employed feature encoding methods, although diverse, are still traditional methods and might not capture deeper contextual information in sequences. Charoenkwan et al. (2023) introduced the PSRQSP model for QSP prediction. Their approach involved the use of propensity scores for 20 amino acids and 400 dipeptides, implemented with a scoring card method to construct PSRQSP. Impressively, the PSRQSP model outperformed existing prediction models, achieving an accuracy of 94.44% [20]. Despite its success, the PSRQSP model's reliance on propensity scores and scoring card methods may limit its ability to generalize across different datasets. Additionally, while the model improved accuracy, it did not significantly innovate in terms of feature extraction or machine learning techniques. It has also been shown that only a limited number of studies have utilized deep learning models, such as CNN, in the bioinformatics field for the prediction of various proteins, peptide, RNA, DNA, and virus sequences [15,24].

Numerous studies have delved into QSP prediction; however, there is still considerable scope for advancing QSP identification. While existing models have achieved notable performance, the reliance is predominantly on traditional amino acid-based feature extraction methods. There is a need for integrating more advanced techniques, such as word embedding feature extraction, which can capture more nuanced patterns in the data. Additionally, exploring the combination of different machine learning and deep learning algorithms can potentially lead to more robust and accurate models. Considering the broad-reaching ap-

plications of QSP in significant domains, there is an urgent demand for a more precise, reliable, and cost-effective predictive model. In light of these imperatives, our research aims to develop a QSP identification model that excels in terms of accuracy and effectiveness, surpassing existing QSP predictors.

The research contribution is to enhance the performance of QSP prediction. This study has accomplished this by combining word embedding feature extraction methods with some amino acid-based feature extractors, creating a more resilient prediction model. This model is further enhanced by the implementation of a diverse number of ML and deep learning (DL) algorithms, ensuring a more precise and robust prediction of QSP. The novelty of this research lies in the integration of advanced word embedding techniques with traditional amino acid feature extraction methods, which has not been extensively explored in the context of QSP prediction. This novel approach leverages the strengths of both techniques to create a more accurate and effective model, pushing the boundaries of current QSP predictive capabilities.

2. Research Methodology

2.1. Dataset Description

To build a more robust and effective QSP predictor in this research, the existing datasets previously utilized by different researchers in their studies were employed [16–20]. The resulting curated dataset contains a total of 440 peptide sequences. Among them, 220 are QSP sequences, and 220 are non-QSP sequences. The entire data set was divided into training and independent test sets for the purposes of this study. The training dataset comprises 310 sequences, with 155 QSP sequences and the remaining being non-QSP sequences. The independent test dataset contains 130 sequences, with an equal number of QSP and non-QSP sequences.

2.2. K-fold Cross Validation (CV)

The 5-fold Cross-Validation (CV) is a commonly employed technique in ML for model evaluation. It involves dividing the dataset into five roughly equal subsets or folds. The training and testing procedure is iterated five times, with each iteration using four folds for training and one fold for testing. This method is particularly useful in classification techniques, as it helps mitigate overfitting.

2.3. Feature Encoding

2.3.1. Pseudo Amino Acid Composition

Pseudo amino acid composition (PAAC) is a computational representation of protein or peptide sequences that encode various properties of amino acids to aid in bioinformatics analyzes, such as prediction of protein structure and prediction of function [21–23]. The PAAC can be defined as:

$$X_c = \frac{f_c}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{\lambda}\theta_j}, (1 < c < 20) \quad (1)$$

$$X_c = \frac{w\theta_{c-20}}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{\lambda}\theta_j}, (21 < c < 20+\lambda) \quad (2)$$

where, X_c represents the components of the PAAC vector, where f_c is the normalized occurrence frequency of the c-th amino acid, and θ_j denotes the sequence-order correlation factors capturing the order of amino acids in the sequence. The sum $\sum_{r=1}^{20} f_r$ accounts for the total frequencies of occurrence of all 20 standard amino acids, while λ defines the maximum sequence distance considered for correlation. The weighting factor w, typically set at 0.05, balances the influence of sequence-order information against amino acid composition.

2.3.2. Composition, Transition, Distribution

Composition, Transition, Distribution, Transition (CTDT) is a feature extraction method in bioinformatics that combines four types of features to represent protein or peptide sequences, aiding in various computational analyses [24,25]. CTDT works as follows:

$$T(r,s) = \frac{N(r,s)+N(s,r)}{N-1}, (r,s)\varepsilon[(polar, neutral), (neutral, hydrophobic), (hydrophobic, polar)] \quad (3)$$

$N(r,s)$ and $N(s,r)$ represent the counts of dipeptides encoded as "rs" and "sr", respectively, in the sequence. "N" corresponds to the length of the sequence.

2.3.3. Grouped Dipeptide Composition

An alternative variant of the Dipeptide Composition descriptor, known as the Grouped Dipeptide Composition Encoding (GDPC), comprises a total of 25 dimensions [26,27]. GDPC is defined as:

$$f(r,s) = \frac{N_{rs}}{N-1}, r,s\varepsilon(g1,g2,g3,g4,g5) \quad (4)$$

Here, N_{rs} denotes the quantity of tripeptides produced by the amino acid type groups denoted by the letters "r" and "s", and N is the total length of a peptide or protein sequence.

2.3.4. FastText

FastText is a word embedding method that represents words as dense vectors, capturing their semantic and syntactic meanings. It is known for its speed and ability to handle out-of-vocabulary words, making it valuable for various natural language processing tasks [28,29]. In recent years, FastText has gained popularity among researchers for encoding biological sequences [30–32]. The FastText works based on the following equation:

$$E(w) = \sum_{g\varepsilon G_w} Z_g \quad (5)$$

Here, $E(w)$ represents the word vector for word w, G_w is the set of subword (character) n-grams for the word w, Z_g represents the vector for subword g in G_w.

2.3.5. Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a word embedding method that represents words as vectors in a high-dimensional space, capturing their semantic relationships and meanings by analyzing the co-occurrence patterns of words in a large corpus of text [33,34]. In the field of biological sequence data encoding, researchers have explored the use of LSA word embedding techniques in their studies [35,36]. LSA is characterized by two key components: the term-document matrix and singular value decomposition (SVD) [37]. These two terms are defined as follows:

$$M = \begin{bmatrix} m_{11}, m_{12}, \ldots, m_{1n}, \\ m_{21}, m_{22}, \ldots, m_{2n} \\ \ldots \ldots \ldots \\ m_{m1}, m_{m2}, \ldots, m_{mn} \end{bmatrix} \quad (6)$$

$$M = U \times \Sigma \times V^T \quad (7)$$

In this case, the matrices U, Σ, and V^T have dimensions $(m \times k), (k \times k)$, and $(k \times n)$, respectively. The chosen reduced dimensionality, $'k'$, corresponds to the number of latent semantic dimensions selected for embedding.

2.4. Development of DeepQSP using Deep Learning

Deep learning, a subset of machine learning, is a powerful tool in computational biology for analyzing complex datasets. It involves training artificial neural networks on large data sets to recognize patterns and make predictions. In this study, the deep learning method is utilized to identify QSPs, which play a crucial role in bacterial communication. The approach leverages the ability of neural networks to process and learn from the intricate patterns in peptide sequences, enabling more accurate and efficient identification compared to traditional methods. The developed deep learning-based method is named DeepQSP, where the CNN layer is employed. Then the performances of the DeepQSP are compared with different machine learning to ensure that the proposed DeepQSP is highly capable of identifying QSPs from protein sequences. The structural architecture of this study has been represented in Fig. 1.

CNN was used to develop DeepQSP for the identification of Quorum Sensing Peptides. The process starts with the input layer, which applies convolution using 64 filters of kernel size 2. This operation can be mathematically represented as [54,55]

$$y = f(W \times x + b) \tag{8}$$

where x is the input, W represents the weights of the filters, and b is the bias, and $\times$ denotes the convolution operation. The ReLU activation function is then applied for non-linear transformation. The ReLU activation function can be expressed as follows [55]:

$$f(x) = max(0, x) \tag{9}$$

Subsequent hidden layers, with 128 filters each and ReLU activation, further process these features. The pooling layer, with a kernel size of 4, reduces the spatial dimensions (downsampling), which can be represented as [55,56]

$$P(x)_{ij} = \max_{k,l \in [1,2,3,4]} x_{4i+k,4j+l} \tag{10}$$

where $P(x)_{ij}$ is the pooled output. Afterward, a flattened layer is employed to convert all the previous layer's output from pooled feature maps into a single long continuous linear vector. The network then includes dense layers with 128 and 64 nodes respectively, again using ReLU activation, to interpret these features. The final output layer with a sigmoid activation function, is used for binary classification. The mathematical expression of sigmoid function is as follows [55–57]:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{11}$$

The model is trained over 40 epochs with batches of 64 samples each, optimizing the network's ability to identify QSPs from the input data accurately. This architecture is a strategic blend of convolutional layers and dense layers, using non-linear activation functions for complex pattern recognition in peptide sequences. The structural architecture of the proposed model is illustrated in Fig. 2.

2.5. Applied Machine Learning Algorithms

After developing the optimized deep learning model, DeepQSP, to identify QSPs, we applied various machine learning algorithms and compared the performance of the proposed DeepQSP model with the five outperformed machine learning algorithms to justify that our proposed model is the most promising solution to identify QSPs. For selection of the model this study focuses on the performances of the applied models and used Randomized Search CV to choose the specific parameters of the models. A brief description of the applied machine learning models is provided in the following subsections. The selected parameters of the applied models have been represented in Table 1.

Table 1
Parameters of the applied models of this study.

Model	Parameter			
CNN	Layer	Filters	Kernel Size/Pool size	Activation function
	Input	64	2	relu
	Hidden-1	128	2	relu
	Hidden-2	128	2	relu
	Pooling		4	
	Dense-1	128		relu
	Dense-2	64		relu
	Output	1		sigmoid
	epoch: 40; batch size: 64			
RF	n_estimators = 200, max_depth = 5			
GB	n_estimators = 200, learning_rate = 0.5, random_state = 50			
ADB	n_estimators = 200, learning_rate = 0.1, random_state = 50			
XGB	n_estimators = 200,max_depth = 5, learning_rate = 0.1			
LGBM	learning_rate = 0.1,max_depth = 5,random_state = 50			

2.5.1. Random Forest (RF)

An RF is a powerful and versatile ML algorithm used for both classification and regression tasks. It is an ensemble learning method that combines the predictions of multiple decision trees to improve accuracy and reduce overfitting [38]. The key advantages of RF are its ability to handle high-dimensional data, handle missing values, and provide a measure of feature importance. It also tends to be more robust and less prone to overfitting compared to individual decision trees. RF is widely used in various fields, including finance, healthcare, and image recognition, due to its reliability and excellent performance in a wide range of applications [39,40].

2.5.2. Gradient Boosting (GB)

Gradient Boosting (GB) is a powerful ML technique for regression and classification tasks. It is a group learning method that builds a strong predictive model by combining the predictions of multiple weak learners, usually decision trees, sequentially [41]. The key advantage of Gradient Boosting is its ability to create highly accurate models, even when dealing with complex relationships in the data. It is robust against overfitting and can handle both numerical and categorical features [42].

2.5.3. AdaBoost (ADB)

AdaBoost (ADB), a short form of Adaptive Boosting, is an ensemble ML technique used primarily for binary classification tasks. It aims to improve the accuracy of weak classifiers by combining them into a strong classifier [43,44]. ADB is particularly useful when dealing with complex data where simple models struggle. It adapts by giving more emphasis to difficult-to-classify examples, effectively creating a strong model from a collection of weak ones. While ADB can be sensitive to noisy data and outliers, it is a popular choice in many practical applications, including face detection and text classification, due to its ability to improve classification accuracy significantly [45,46].

2.5.4. XGBoost (XGB) Classifier

XGBoost (XGB), a short form of Extreme Gradient Boosting, is a highly efficient and popular ML algorithm known for its exceptional performance in various tasks, particularly in classification and regression. It is an advanced implementation of the gradient boosting framework with several optimizations and features, making it a preferred choice among data scientists and ML practitioners [47,48]. XGB is a versatile and powerful tool in the field of ML, known for its ability to produce accurate and robust models across a wide range of applications, from natural language processing to image classification and beyond [49].

2.5.5. LightGBM (LGBM) Classifier

LightGBM (LGBM), short form Light Gradient Boosting Machine, is a high-performance gradient boosting framework designed for efficient and accurate machine learning tasks, particularly in the domains of classification, regression, and ranking. LGBM is known for its speed,

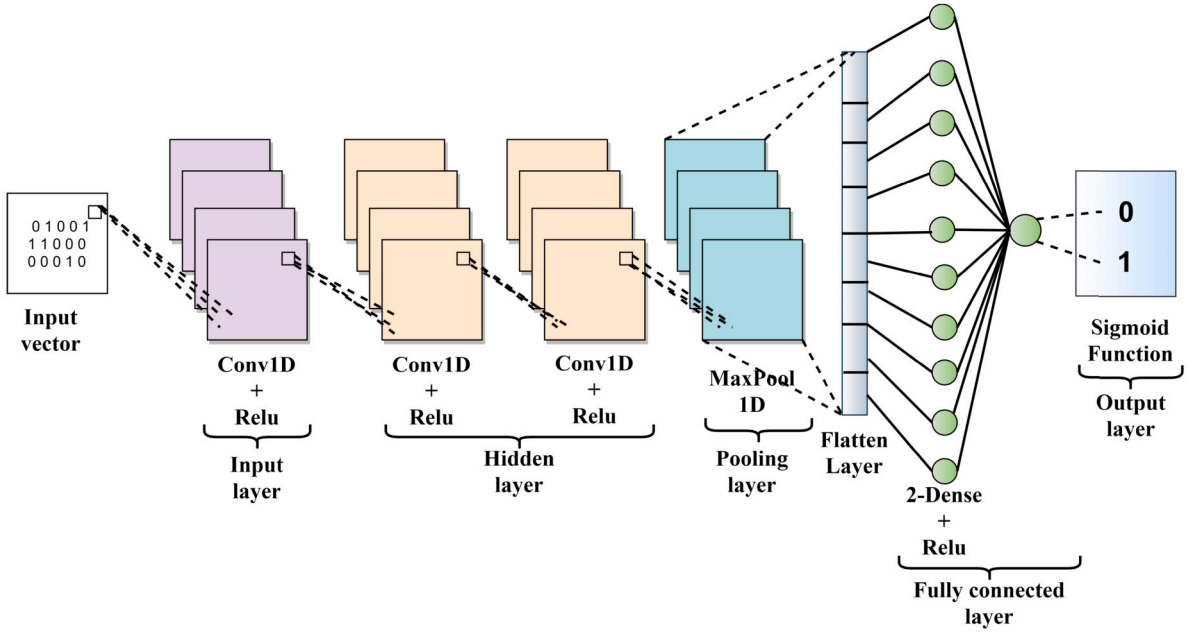


Fig. 2. Structural architecture of the proposed model DeepQSP, which has been developed using CNN architecture.

memory efficiency, and ability to handle large datasets [50–52]. LGBM has gained popularity in ML competitions and real-world applications due to its exceptional speed and performance [53].

2.6. Evaluation Metrics

Evaluating the performance of deep learning (DL) and ML models is a critical step in assessing their effectiveness and predictive capabilities. The choice of evaluation metrics depends on the nature of the problem at hand, whether it is classification, regression, or another specific task. In the assessment of the applied models, a set of six performance metrics was employed to gauge their effectiveness. The accuracy metric measures the proportion of correctly classified instances relative to the total number of instances [58]. Specificity is calculated as the count of correct negative predictions divided by the total number of actual negative instances [59]. Sensitivity is calculated as the number of true positive predictions divided by the total number of actual positive instances [60].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{12}$$

$$Specificity = \frac{TN}{TN + FP} \tag{13}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{14}$$

Another performance metric is the F1-Measure. The F1-Measure is a metric that combines precision and recall to provide a balanced assessment of a model's performance in classification tasks [59]. Kappa Statistics assesses agreement between raters or classifiers, accounting for chance agreement, in various tasks such as inter-rater reliability or classification model evaluation [61]. The Matthews Correlation Coefficient (MCC) is a correlation measure that falls within the range of -1 to +1, effectively quantifying the strength of association between two variables [60].

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{15}$$

$$KappaStat = \frac{observed\ accuracy - expected\ accuracy}{1 - expected\ accuracy} \tag{16}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{17}$$

Here, TP stands for true positive, TN represents true negative, FP corresponds to false positive, and FN denotes false negative.

3. Result Analysis and Discussion

In this work, a two-stage analysis was conducted. Initially, a biological analysis of the peptide sequences was performed using the Biopython software package [62]. Subsequently, we employed Python version 3.10.12 within the Google Colab environment to develop and assess the model's performance.

3.1. Analysis of Peptide Sequence

Fig. 3 shows the percentage of amino acids in the datasets used. Based on the figure, the F (phenylalanine) has the highest percentage value among the twenty amino acids, considering the QSP sequence. G (glycine) shows the maximum percentage among other amino acids for the non-QSP sequence. The lowest percentage value among the amino acids is W (tryptophan) for the non-QSP. For the QSP sequence, H (histidine) has the lowest percentage. Here it can be seen that the percentage value of the non-QSP sequence is significantly higher than the QSP sequence.

3.2. Analysis of the Classifiers Algorithm Result

In this study, 5-fold Cross-Validation (CV) was applied to construct classification models for the training dataset. The utilized datasets are balanced, with an equal number of QSP and non-QSP sequences, eliminating the need for any balancing method. After applying 5-fold CV to the training dataset, the classification model was tested on an independent dataset. In addition, the feature extractor was merged, combining 3 amino acid property-based extractors with 2-word embedding techniques, resulting in six different combinations of feature extraction methods. The results of the 5-fold CV and independent test are represented in Table 2 and Table 3.

Table 2 presents the cross-validation (CV) results for various classifier models using different feature extraction methods. The LSA+GDPC feature extractor method yields the highest performance across nearly all evaluation metrics when used with the CNN classification model. Specifically, this combination achieves an accuracy of 0.9672, a sensitivity of 1.00, a specificity of 0.9355, an F1-measure of 0.9677, and MCC

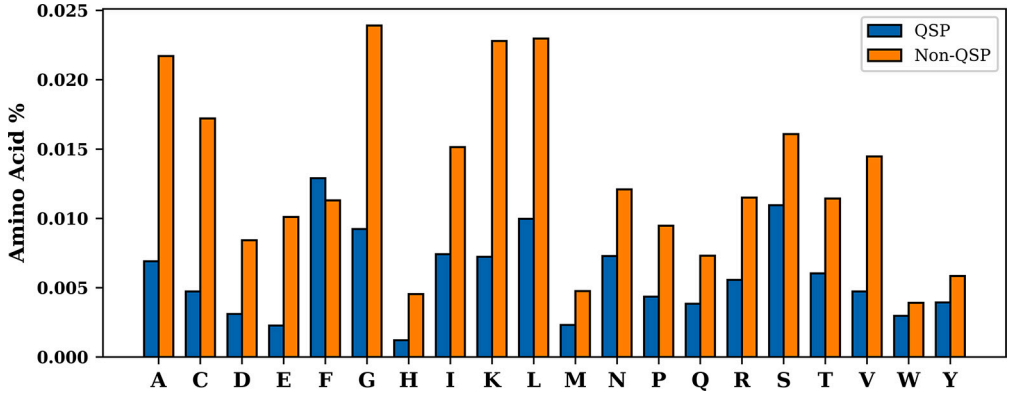


Fig. 3. Percentage of the amino acids in the used QSP dataset.

Table 2
5-fold CV results of the different combined feature extractors.

Extractor	Classifiers	Accuracy	Sensitivity	Specificity	F1-Measure	MCC	Kappa
LSA + PAAC	RF	0.7857	0.8366	0.7355	0.795	0.5748	0.5717
	XGB	0.8084	0.8104	0.8064	0.8078	0.6169	0.6169
	LGBM	0.8084	0.8104	0.8064	0.8078	0.6169	0.6169
	GB	0.8344	0.8301	0.8387	0.8328	0.6688	0.6688
	ADB	0.8279	0.8758	0.7806	0.8349	0.6592	0.656
	CNN	0.9672	0.9667	0.9677	0.9667	0.9344	0.9344
LSA + CTDT	RF	0.8006	0.7671	0.8312	0.786	0.6003	0.5996
	XGB	0.7879	0.7397	0.8312	0.7687	0.5744	0.5729
	LGBM	0.8072	0.8219	0.7937	0.8027	0.615	0.6144
	GB	0.8235	0.8082	0.8375	0.8138	0.6462	0.6461
	ADB	0.7908	0.7603	0.8187	0.7762	0.5805	0.5801
	CNN	0.918	0.963	0.8823	0.9123	0.8357	0.8398
LSA + GDPC	RF	0.8377	0.8456	0.8302	0.8344	0.6755	0.6752
	XGB	0.8961	0.8725	0.9182	0.8904	0.7923	0.7917
	LGBM	0.9091	0.8926	0.9245	0.9048	0.8181	0.8178
	GB	0.8896	0.8658	0.9119	0.8836	0.779	0.7787
	ADB	0.8474	0.8121	0.8805	0.8374	0.6951	0.6939
	CNN	0.9672	1	0.9355	0.9677	0.9365	0.9345
FastText + PAAC	RF	0.8562	0.8973	0.8187	0.8562	0.716	0.713
	XGB	0.8693	0.8699	0.8687	0.8639	0.7382	0.7382
	LGBM	0.8529	0.863	0.8437	0.8485	0.7061	0.7057
	GB	0.866	0.8767	0.8562	0.8619	0.7322	0.7319
	ADB	0.8431	0.8699	0.8187	0.841	0.688	0.6865
	CNN	0.9344	0.931	0.9375	0.931	0.8685	0.8685
FastText + CTDT	RF	0.8006	0.7671	0.8312	0.786	0.6003	0.5996
	XGB	0.7876	0.7397	0.8312	0.7687	0.5744	0.5729
	LGBM	0.8072	0.8219	0.7937	0.8028	0.615	0.6144
	GB	0.8235	0.8082	0.8375	0.8138	0.6462	0.6461
	ADB	0.7908	0.7603	0.8187	0.7762	0.5805	0.5801
	CNN	0.9508	0.9687	0.931	0.9538	0.9017	0.9012
FastText + GDPC	RF	0.8333	0.8278	0.8387	0.8306	0.6666	0.6666
	XGB	0.8431	0.841	0.8452	0.841	0.6862	0.6862
	LGBM	0.8268	0.8212	0.8322	0.8239	0.6535	0.6535
	GB	0.8497	0.8344	0.8645	0.8456	0.6994	0.6992
	ADB	0.8203	0.8079	0.8322	0.816	0.6405	0.6404
	CNN	0.9016	0.9286	0.8788	0.8965	0.8047	0.803

and Kappa values of 0.9365 and 0.9345, respectively. In contrast, the LSA+PAAC feature encoding method, while matching the LSA+GDPC method's accuracy of 0.9672, surpasses it in specificity, achieving a score of 0.9677. However, this comes with a slightly reduced sensitivity of 0.9667, an F1-measure of 0.9667, and MCC and Kappa values of 0.9344. The lowest performance is observed with the Random Forest (RF) classifier when paired with the LSA+PAAC feature extractor. This combination results in the minimum scores for accuracy of 0.7857, sensitivity of 0.8366, specificity of 0.7355, F1-measure of 0.7950, MCC of 0.5748, and Kappa of 0.5717.

Table 3 presents the independent test results of this study, highlighting the performance of various feature encoding methods and classifier models. The LSA+PAAC and FastText+PAAC encoding methods achieved the highest accuracy, both recording 0.9697 on the CNN model. For the LSA+PAAC extractor with the CNN model, the sensitivity was 0.9730, the specificity 0.9655, the F1-measure 0.9730, and the Kappa and MCC scores were both 0.9385. Similarly, the FastText+PAAC method on the CNN model delivered impressive results, with a sensitivity of 0.9855, specificity of 0.9524, F1-measure of 0.9714, Kappa score of 0.9392, and MCC of 0.9396. In contrast, the XGB classifier recorded the lowest accuracy, at 0.7954, along with an MCC of 0.5920 and a Kappa score of 0.5903. The ADB classifier demonstrated the lowest sensitivity and F1-score, both at 0.7833, when using the LSA+CTDT feature extractor. Additionally, the lowest specificity was observed in the RF model with the LSA+GDPC encoding method, which registered a score of 0.7714.

Fig. 4, illustrates the ROC curves along with the AUC scores for the six classifiers used in this study. In subplot (A), the CNN model achieved the highest AUC score of 0.995, while the ADB model recorded the lowest AUC score of 0.893. For the independent test results shown in

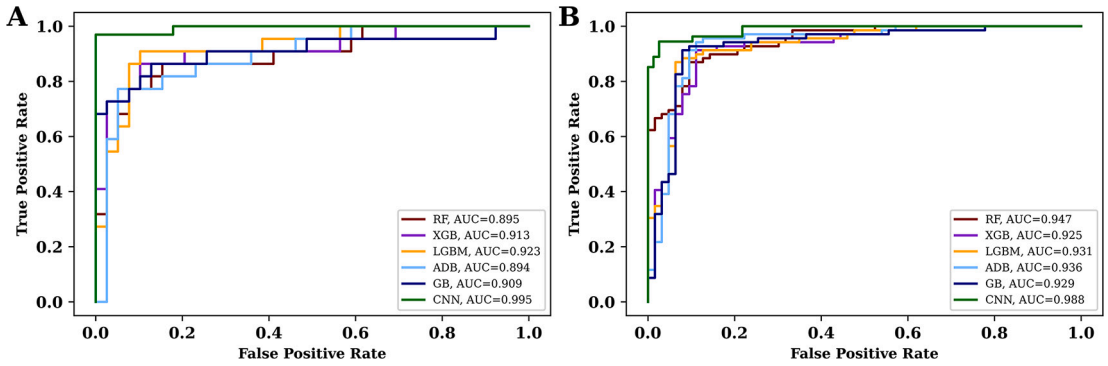


Fig. 4. ROC curve for the different ML classifiers on LSA+PAAC feature encoding method. Subplot(A) shows the result of 5-flod CV and subplot(B) shows the result of the independent test.

Table 3
Independent Test results of the different combined feature extractors.

Extractor	Classifiers	Accuracy	Sensitivity	Specificity	F1-Measure	MCC	Kappa
LSA + PAAC	RF	0.8485	0.8955	0.8	0.8571	0.6994	0.6965
	XGB	0.8561	0.8358	0.8769	0.855	0.713	0.7122
	LGBM	0.8712	0.8658	0.8769	0.8722	0.7425	0.7424
	GB	0.9015	0.9104	0.8923	0.9037	0.803	0.8029
	ADB	0.9091	0.9254	0.8923	0.9118	0.8184	0.818
	CNN	0.9697	0.973	0.9655	0.973	0.9385	0.9385
LSA + CTDT	RF	0.8257	0.8167	0.8333	0.8099	0.6492	0.6491
	XGB	0.7954	0.8167	0.7778	0.784	0.592	0.5903
	LGBM	0.8257	0.85	0.8055	0.816	0.6529	0.651
	GB	0.8182	0.8333	0.8055	0.8064	0.6365	0.6353
	ADB	0.803	0.7833	0.8194	0.7833	0.6028	0.6028
	CNN	0.9318	0.9077	0.9552	0.9291	0.8644	0.8635
LSA + GDPC	RF	0.8257	0.8871	0.7714	0.8271	0.6592	0.6531
	XGB	0.9015	0.9032	0.9	0.896	0.8026	0.8025
	LGBM	0.947	0.9516	0.9428	0.944	0.8937	0.8936
	GB	0.9242	0.9032	0.9428	0.918	0.848	0.8476
	ADB	0.9091	0.9193	0.9	0.9048	0.8182	0.8178
	CNN	0.9621	0.9851	0.9385	0.9635	0.9251	0.9242
FastText + PAAC	RF	0.8257	0.863	0.7797	0.8456	0.6465	0.6458
	XGB	0.8257	0.8356	0.8135	0.8414	0.6482	0.6481
	LGBM	0.8182	0.863	0.7627	0.84	0.631	0.6298
	GB	0.8485	0.8493	0.8474	0.8611	0.6948	0.6945
	ADB	0.8333	0.863	0.7966	0.8513	0.6621	0.6618
	CNN	0.9697	0.9855	0.9524	0.9714	0.9396	0.9392
FastText + CTDT	RF	0.8712	0.8356	0.9152	0.8777	0.7466	0.7424
	XGB	0.8712	0.8767	0.8644	0.8827	0.74	0.7399
	LGBM	0.9015	0.8904	0.9152	0.9091	0.8026	0.8017
	GB	0.8864	0.863	0.9152	0.8936	0.7742	0.772
	ADB	0.8864	0.8082	0.983	0.8872	0.7901	0.7749
	CNN	0.9318	0.9846	0.8806	0.9343	0.8687	0.8638
FastText + GDPC	RF	0.8485	0.8529	0.8437	0.8529	0.6967	0.6967
	XGB	0.8409	0.897	0.7812	0.8531	0.6844	0.6805
	LGBM	0.8788	0.9118	0.8437	0.8857	0.7583	0.7569
	GB	0.8864	0.9265	0.8437	0.8936	0.7742	0.772
	ADB	0.8182	0.8676	0.7656	0.831	0.6377	0.635
	CNN	0.9394	0.9687	0.9118	0.9394	0.8805	0.8789

subplot (B), the CNN model again secured the highest AUC score at 0.988, whereas the XGB model had the lowest AUC score, registering at 0.925. Notably, in both subplots, the CNN model consistently achieved the highest AUC score, demonstrating its superior performance among the tested classifiers.

Fig. 5 compares the accuracy and MCC of six feature extraction methods across six classifier models, resulting in a total of thirty-six models. According to the figure, the LSA+GDPC_CNN and LSA+PAAC_CNN models occupy the top two positions in the 5-fold cross-validation (CV) in terms of accuracy and MCC. For the independent test, the FastText+PAAC_CNN and LSA+PAAC_CNN models lead in both accuracy and MCC performance metrics. Notably, the LSA+PAAC_CNN model consistently ranks among the top models in both the 5-fold CV and the independent test. In contrast, the LSA+PAAC_RF and FText+CTDC_XGB models rank at the bottom for accuracy and MCC in the 5-fold CV. Additionally, in both subplot (C) and subplot (D), the LSA+CTDC_XGB model is positioned at the bottom, reflecting its lower performance in both the 5-fold CV and the independent test.

Fig. 6 presents the sensitivity and specificity comparison of the thirty-six models. According to the figure, in the 5-fold cross-validation (CV), the LSA+PAAC_CNN model achieved the highest specificity among all models. However, for sensitivity in the 5-fold CV, the LSA+PAAC_CNN model ranked third, with the LSA+GDPC_CNN model taking the top position. In the independent test, the FastText+CTDC_GB model ranked highest for specificity, with the LSA+PAAC_CNN model coming in second place. For sensitivity in the independent test, the FastText+PAAC_CNN model took the lead, while the LSA+PAAC_CNN model secured the fourth position among all the models.

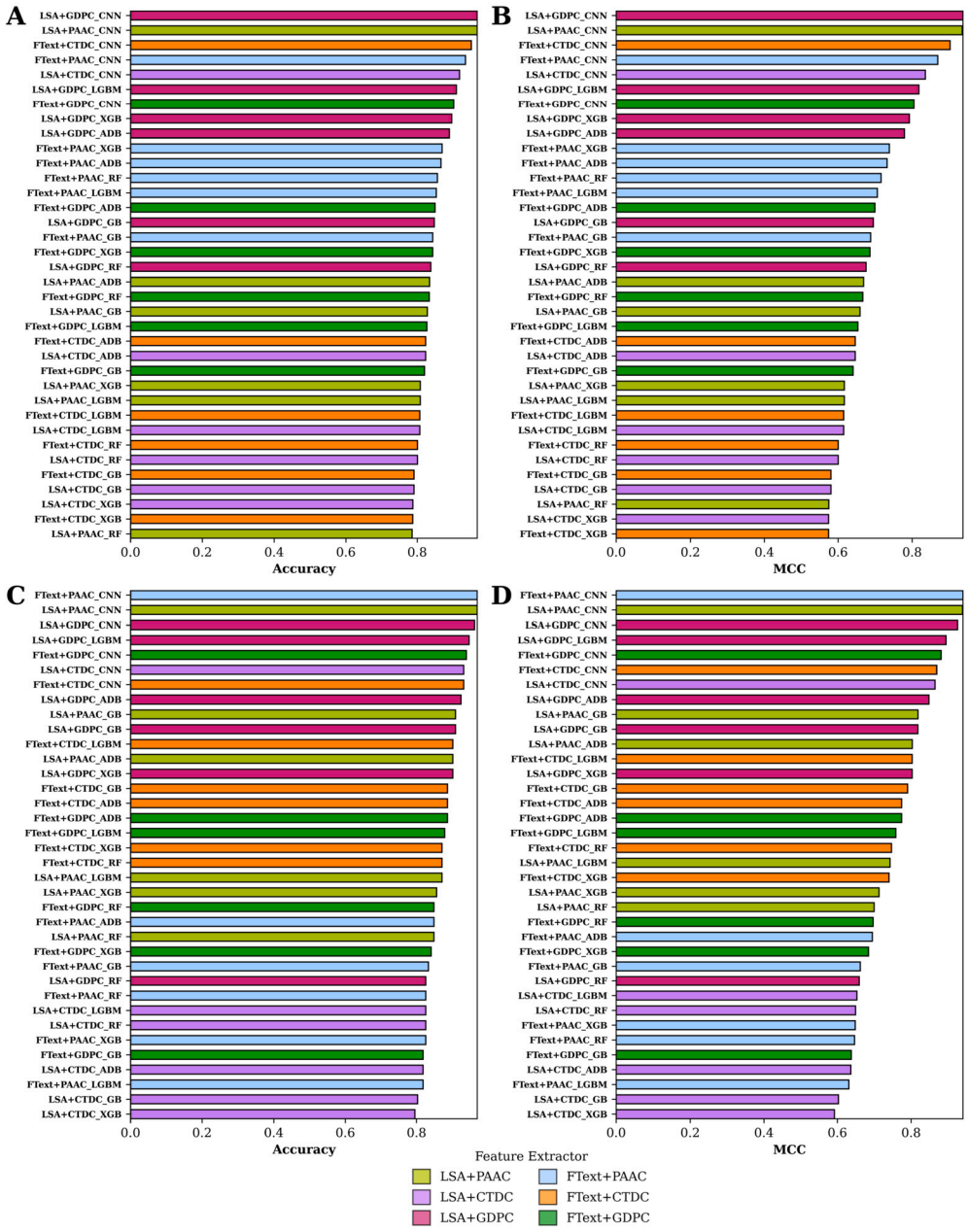


Fig. 5. Comparison of the Accuracy and MCC of the combined feature encoding techniques and applied ML algorithm. Subplot(A) and subplot(B) for the 5-fold CV result. Subplot(C) and subplot(D) for the independent test result.

3.3. Discussion

The identification of quorum-sensing peptides is vital for comprehending and managing microbial activities and driving progress across diverse sectors, including healthcare, food, agriculture, drug discovery, biotechnology, and environmental science. Based on the analysis of the results discussed above, it can be concluded that the LSA+PAAC_CNN model demonstrates superior performance compared to all other models applied in the study. Though some models perform well for 5-fold CV and some models perform well in independent tests, the LSA+PAAC_CNN model outperforms on both the 5-fold CV and independent tests, considering all the evaluation metrics. The results of our developed DeepQSP prediction model on the independent test set are as follows: an accuracy of 0.9697, a specificity of 0.9730, a sensitivity of 0.9655, an F1-measure of 0.9730, and a kappa score and MCC score both equal to 0.9385. Additionally, the AUC score for our proposed DeepQSP model on the independent test is 0.988.

According to Table 4 and Fig. 7, the performance of our proposed DeepQSP model significantly improves over the existing QSP prediction model. Our DeepQSP model outperforms other existing prediction models by a margin of 0.0253 to 0.0697 in terms of accuracy.

Although our developed DeepQSP outperforms other QSP predictor models, there are still some limitations and future scope for this study. First, the dataset used in this study is relatively small, consisting of only 440 instances. A larger dataset would provide a more comprehensive representation of QSP sequences, improving the model's ability to generalize and perform accurately on unseen data. Next, while the combination of word embedding and amino acid-based feature extraction has shown promising results, there may still be other feature extraction methods that could further enhance the model's performance. Deep learning models, including CNNs, are often considered "black boxes" due to their complexity, making it difficult to interpret the specific features or patterns that drive their predictions. Accordingly, our future

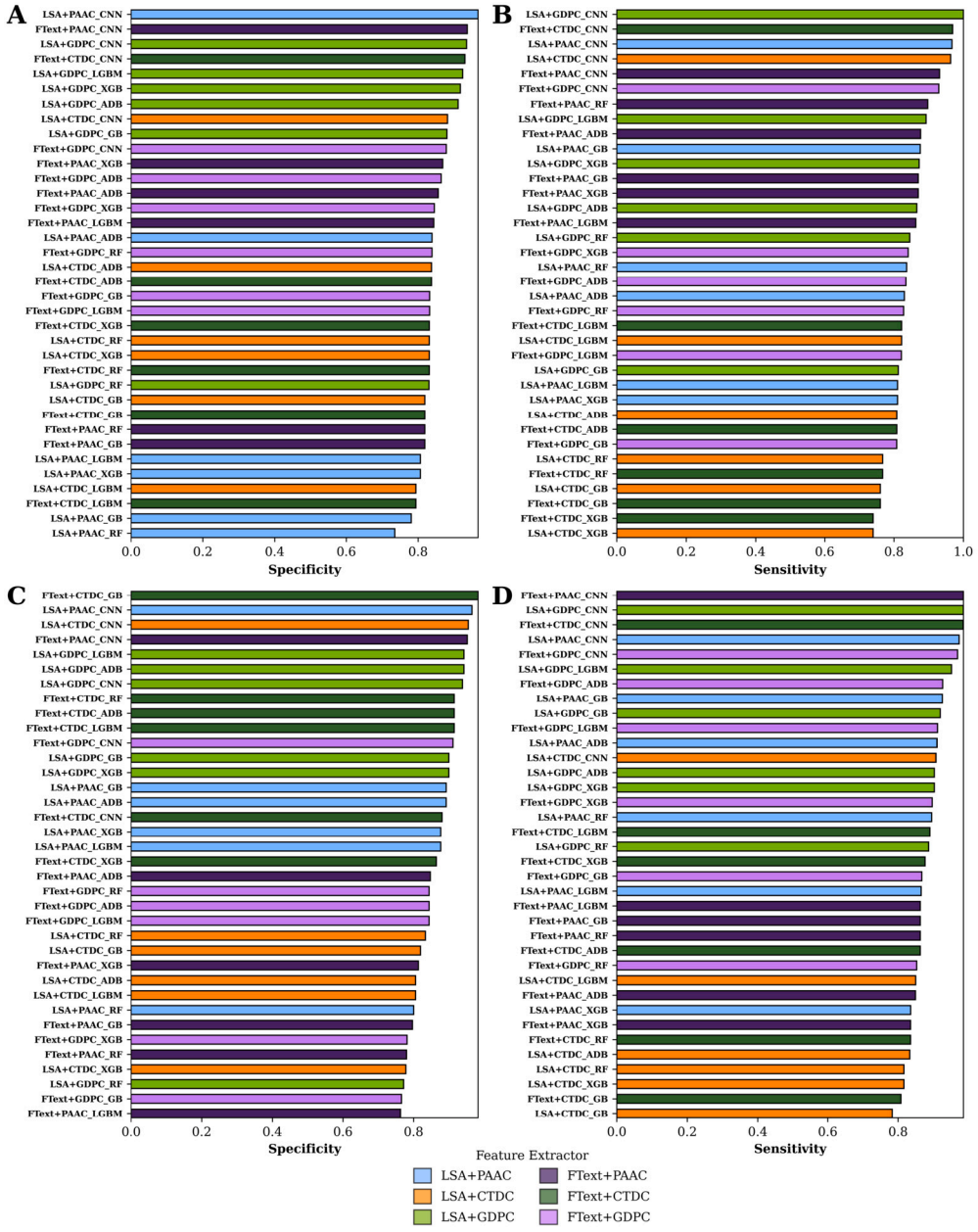


Fig. 6. Comparison of the specificity and sensitivity of the combined feature encoding techniques and applied ML algorithm. Subplot(A) and subplot(B) for the 5-fold CV result. Subplot(C) and subplot(D) for the independent test result.

Table 4
Comparing DeepQSP with existing other QSP prediction models.

Prediction Model	Accuracy	Specificity	Sensitivity	Kappa	MCC
QSPpred [16]	0.9	-	-	-	0.8
QSPred-FL [17]	0.925	-	-	-	0.86
iQSP [18]	0.93	0.935	0.925	-	0.86
EnsembleQS [19]	0.934	-	-	-	0.91
PSRQSP [20]	0.9444	1	0.882	-	0.893
DeepQSP [this work]	**0.9697**	**0.973**	**0.9655**	**0.9385**	**0.9385**

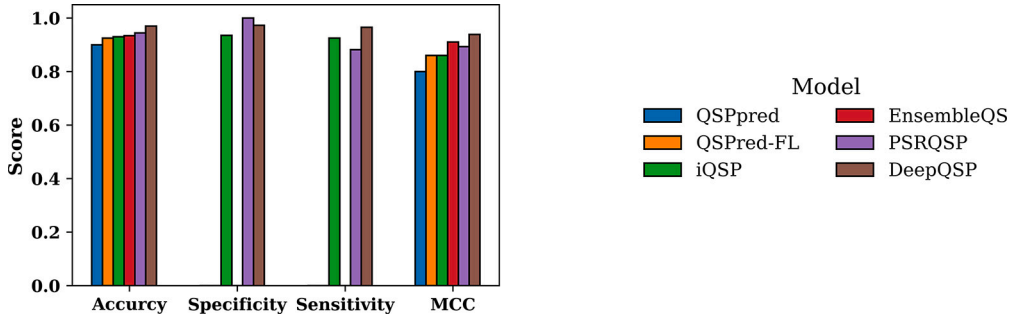


Fig. 7. Performance comparison of DeepQSP with previous existing studies.

work includes building a more reliable and robust QSP predictor model, with the collection of a larger and more diverse QSP dataset. This should help in improving the generalizability and performance of the model in different microbial communities and contexts. The implementation of a web server based on the developed DeepQSP predictor is planned. This should provide an accessible tool for researchers and practitioners in the bioengineering field and the broader scientific community, facilitating the use of our model for various applications. Future work will also explore the integration of additional feature extraction methods, such as structural and functional properties of QSPs, to further enhance the model's predictive capabilities. The use of transfer learning techniques will be investigated to leverage knowledge from related domains and enhance the model's performance with limited data. Ongoing efforts will focus on optimizing the model's architecture and training process to reduce computational requirements and enhance efficiency, making it more accessible for practical applications. Furthermore, future research will explore methods to improve the interpretability of our DeepQSP model, enabling researchers to understand the key features and patterns driving the model's predictions. Techniques such as attention mechanisms or explainable AI methods could be investigated. By addressing these limitations and pursuing these future research directions will further enhance the capabilities of the DeepQSP model, significantly contributing to the field of microbial communication research and its applications in biotechnology, healthcare, and beyond.

4. Conclusion

In summary, the fusion of Convolutional Neural Network (CNN) models with word embedding feature encoding techniques and classical amino acid based encoding methods represents a pioneering approach for Quorum Sensing Peptides (QSP) prediction. This unique amalgamation enhances our comprehension of microbial communication, offering significant improvements in unraveling the complexities of Quorum Sensing. The Convolutional Neural Network-driven model, in conjunction with word embedding, provides a resilient and adaptable tool with diverse research applications, paving the way for innovative therapies in healthcare and reducing reliance on broad-spectrum antibiotics to combat global antibiotic resistance. Our model achieved an impressive accuracy of 0.9697 and a Matthews Correlation Coefficient of 0.9385, marking a transformative phase in microbial communication research with wide-ranging benefits for human health, the environment, and scientific advancement. Further research will focus on expanding the model to predict QSPs in more diverse microbial environments and exploring the integration of additional biological factors into the model. Additionally, future studies will aim to validate the model's predictions experimentally, thereby strengthening its application in biotechnology, healthcare, and other fields.

Consent to Participate

Not applicable.

Consent to Publish

Not applicable.

Ethical Approval

Not applicable.

CRediT authorship contribution statement

Md. Ashikur Rahman: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Data curation. **Md. Mamun Ali:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Kawsar Ahmed:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Imran Mahmud:** Writing – review & editing, Project administration. **Francis M. Bui:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition. **Li Chen:** Writing – review & editing, Validation, Supervision, Funding acquisition. **Santosh Kumar:** Writing – review & editing, Project administration. **Mohammad Ali Moni:** Writing – review & editing, Validation, Supervision, Project administration.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This work was supported, in part, by funding from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Appendix A

See Table 5.

References

[1] W.C. Fuqua, S.C. Winans, E.P. Greenberg, Quorum sensing in bacteria: the LuxR-LuxI family of cell density-responsive transcriptional regulators, J. Bacteriol. 176 (2) (1994) 269–275.

[2] M. MB, Bassler BL. Quorum sensing in bacteria, Annu. Rev. Microbiol. 55 (2001) 165–199.

Table 5
List of Abbreviations.

Short Form	Abbreviations
QSP	Quorum Sensing Peptides
LSA	Latent Semantic Analysis
PAAC	Pseudo Amino Acid Composition
CNN	Convolutional Neural Network
MCC	Matthews correlation coefficient
QS	Quorum sensing
ML	Machine Learning
AAC	Amino Acid Composition
ACRP	Amino Acid Residue Position
PCP	Physicochemical Properties
GRAVY	Grand average of hydropathy
SVM	Support Vector Machine
RF	Random Forest
DPC	Dipeptide Composition
DEM	Dipeptide Deviation from Expected Mean
TPC	Tripeptide Composition
DL	Deep Learning
CV	Cross-Validation
CTDT	Composition, Transition, Distribution, Transition
GDPC	Grouped Dipeptide Composition
SVD	Singular Value Decomposition
GB	Gradient Boosting
ADB	Adaptive Boosting
XGB	Extreme Gradient Boosting
LGBM	Light Gradient Boosting Machine
Amino acid	A: alanine C: cysteine D: aspartic acid E: glutamic acid F: phenylalanine G: glycine H: histidine I: isoleucine K: lysine L: leucine M: methionine N: asparagine P: proline Q: glutamine R: arginine S: serine T: threonine V: valine W: tryptophan Y: tyrosine

[3] G.M. Dunny, B.A. Leonard, Cell-cell communication in gram-positive bacteria, Annu. Rev. Microbiol. 51 (1) (1997) 527–564.

[4] C.M. Waters, B.L. Bassler, Quorum sensing: cell-to-cell communication in bacteria, Annu. Rev. Cell Dev. Biol. 21 (2005) 319–346.

[5] S.T. Rutherford, B.L. Bassler, Bacterial quorum sensing: its role in virulence and possibilities for its control, Cold Spring Harb. Perspect. Med. 2 (11) (2012) a012427.

[6] C.D. Sifri, Quorum sensing: bacteria talk sense, Clin. Infect. Dis. 47 (8) (2008) 1070–1076.

[7] N. Mangwani, H.R. Dash, A. Chauhan, S. Das, Bacterial quorum sensing: functional features and potential applications in biotechnology, J. Mol. Microbiol. Biotechnol. 22 (4) (2012) 215–227.

[8] S. Wu, J. Liu, C. Liu, A. Yang, J. Qiao, Quorum sensing for population-level control of bacteria and potential therapeutic applications, Cell. Mol. Life Sci. 77 (2020) 1319–1343.

[9] A.R. Horswill, P. Stoodley, P.S. Stewart, M.R. Parsek, The effect of the chemical, biological, and physical environment on quorum sensing in structured microbial communities, Anal. Bioanal. Chem. 387 (2007) 371–380.

[10] G.R. Abbamondi, G. Tommonaro, Research progress and hopeful strategies of application of quorum sensing in food, agriculture and nanomedicine, Microorganisms 10 (6) (2022) 1192.

[11] P.N. Skandamis, G.J.E. Nychas, Quorum sensing in the context of food microbiology, Appl. Environ. Microbiol. 78 (16) (2012) 5473–5482.

[12] P.V. Bramhachari (Ed.), Implication of Quorum Sensing and Biofilm Formation in Medicine, Agriculture and Food Industry, Springer, 2019.

[13] D. Sarkar, K. Poddar, N. Verma, S. Biswas, A. Sarkar, Bacterial quorum sensing in environmental biotechnology: a new approach for the detection and remediation of emerging pollutants, in: Emerging Technologies in Environmental Bioremediation, Elsevier, 2020, pp. 151–164.

[14] M. Alsanea, A.S. Dukyil, Riaz B. Afnan, F. Alebeisat, M. Islam, S. Habib, To assist oncologists: an efficient machine learning-based approach for anti-cancer peptides classification, Sensors 22 (11) (2022) 4005.

[15] Y. Pu, J. Li, J. Tang, F. Guo, DeepFusionDTA: drug-target binding affinity prediction with information fusion and hybrid deep-learning ensemble model, IEEE/ACM Trans. Comput. Biol. Bioinform. 19 (5) (2021) 2760–2769.

[16] A. Rajput, A.K. Gupta, M. Kumar, Prediction and analysis of quorum sensing peptides based on sequence features, PLoS ONE 10 (3) (2015) e0120066.

[17] L. Wei, J. Hu, F. Li, J. Song, R. Su, Q. Zou, Comparative analysis and prediction of quorum-sensing peptides using feature representation learning and machine learning algorithms, Brief. Bioinform. 21 (1) (2020) 106–119.

[18] P. Charoenkwan, N. Schaduangrat, C. Nantasenamat, T. Piacham, W. Shoombuatong, iQSP: a sequence-based tool for the prediction and analysis of quorum sensing peptides using informative physicochemical properties, Int. J. Mol. Sci. 21 (1) (2019) 75.

[19] M. Sivaramakrishnan, R. Suresh, K. Ponraj, Predicting quorum sensing peptides using stacked generalization ensemble with gradient boosting based feature selection, J. Microbiol. 60 (7) (2022) 756–765.

[20] P. Charoenkwan, P. Chumnanpuen, N. Schaduangrat, C. Oh, B. Manavalan, W. Shoombuatong, PSRQSP: an effective approach for the interpretable prediction of quorum sensing peptide using propensity score representation learning, Comput. Biol. Med. 158 (2023) 106784.

[21] K.C. Chou, Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology, Curr. Proteomics 6 (4) (2009) 262–274.

[22] K.C. Chou, Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes, Bioinformatics 21 (1) (2005) 10–19.

[23] K.C. Chou, Prediction of protein cellular attributes using pseudo-amino acid composition, Proteins, Struct. Funct. Bioinform. 43 (3) (2001) 246–255.

[24] F.A. Mostafa, Y.M. Afify, R.M. Ismail, N.L. Badr, Deep learning model for protein disease classification, Curr. Bioinform. 17 (3) (2022) 245–253.

[25] C. Wang, J. Wu, L. Xu, Q. Zou, NonClasGP-Pred: robust and efficient prediction of non-classically secreted proteins by integrating subset-specific optimal models of imbalanced data, Microbial genomics 6 (12) (2020).

[26] Z. Chen, P. Zhao, F. Li, T.T. Marquez-Lago, A. Leier, J. Revote, Y. Zhu, D.R. Powell, T. Akutsu, G.I. Webb, K.C. Chou, iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data, Brief. Bioinform. 21 (3) (2020) 1047–1057.

[27] Z. Chen, P. Zhao, F. Li, A. Leier, T.T. Marquez-Lago, Y. Wang, G.I. Webb, A.I. Smith, R.J. Daly, K.C. Chou, J. Song, iFeature: a python package and web server for features extraction and selection from protein and peptide sequences, Bioinformatics 34 (14) (2018) 2499–2502.

[28] Y. Didi, A. Walha, A. Wali, COVID-19 tweets classification based on a hybrid word embedding method, Big Data Cogn. Comput. 6 (2) (2022) 58.

[29] S. Selva Birunda, R. Kanniga Devi, A review on word embedding techniques for text classification, in: Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020, 2021, pp. 267–281.

[30] N.Q.K. Le, T.T. Huynh, Identifying SNAREs by incorporating deep learning architecture and amino acid embedding representation, Front. Physiol. 10 (2019) 1501.

[31] L. Shi, B. Chen, August. LSHvec: a vector representation of DNA sequences using locality sensitive hashing and FastText word embeddings, in: Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, 2021, pp. 1–10.

[32] H. Lv, F.Y. Dao, H. Zulfiqar, H. Lin, DeepIPs: comprehensive assessment and computational identification of phosphorylation sites of SARS-CoV-2 infection using a deep learning-based approach, Brief. Bioinform. 22 (6) (2021) bbab244.

[33] M. Naili, A.H. Chaibi, H.H.B. Ghezala, Comparative study of word embedding methods in topic segmentation, Proc. Comput. Sci. 112 (2017) 340–349.

[34] A. Joshi, V. Tripathi, K. Patel, P. Bhattacharyya, M. Carman, Are word embedding-based features useful for sarcasm detection?, arXiv preprint, arXiv:1610.00883, 2016.

[35] M.T. Patrick, K. Raja, K. Miller, J. Sotzen, J.E. Gudjonsson, J.T. Elder, L.C. Tsoi, Drug repurposing prediction for immune-mediated cutaneous diseases using a word-embedding–based machine learning approach, J. Invest. Dermatol. 139 (3) (2019) 683–691.

[36] C.M. Liu, V.D. Ta, N.Q.K. Le, D.A. Tadesse, C. Shi, Deep neural network framework based on word embedding for protein Glutarylation sites prediction, Life 12 (8) (2022) 1213.

[37] S.C. Tékouabou, Ş.C. Gherghina, H. Toulni, P.N. Mata, J.M. Martins, Towards explainable machine learning for bank churn prediction using data balancing and ensemble-based methods, Mathematics 10 (14) (2022) 2379.

[38] D. Denisko, M.M. Hoffman, Classification and interaction in random forests, Proc. Natl. Acad. Sci. 115 (8) (2018) 1690–1692.

[39] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, S. Homayouni, Support vector machine versus random forest for remote sensing image classification: a meta-analysis and systematic review, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13 (2020) 6308–6325.

[40] P. Yang, D. Wang, W.B. Zhao, L.H. Fu, J.L. Du, H. Su, Ensemble of kernel extreme learning machine based random forest classifiers for automatic heartbeat classification, Biomed. Signal Process. Control 63 (2021) 102138.

[41] C. Bentéjac, A. Csörgő, G. Martínez-Muñoz, A comparative analysis of gradient boosting algorithms, Artif. Intell. Rev. 54 (2021) 1937–1967.

[42] B. Ma, F. Meng, G. Yan, H. Yan, B. Chai, F. Song, Diagnostic classification of cancers using extreme gradient boosting algorithm and multi-omics data, Comput. Biol. Med. 121 (2020) 103761.
[43] E.O. Ogunseye, C.A. Adenusi, A.C. Nwanakwaugwu, S.A. Ajagbe, S.O. Akinola, Predictive analysis of mental health conditions using AdaBoost algorithm, ParadigmPlus 3 (2) (2022) 11–26.
[44] M. Khan, K. Malik, Sentiment Classification of Customer's Reviews About Automobiles in Roman Urdu, Advances in Information and Communication Networks: Proceedings of the 2018 Future of Information and Communication Conference (FICC), vol. 2, Springer International Publishing, 2019, pp. 630–640.
[45] H. Liu, X. Zhang, X. Zhang, PwAdaBoost: possible world based AdaBoost algorithm for classifying uncertain data, Knowl.-Based Syst. 186 (2019) 104930.
[46] L. Hao, G. Huang, An improved AdaBoost algorithm for identification of lung cancer based on electronic nose, Heliyon 9 (3) (2023).
[47] M. Zivkovic, N. Bacanin, M. Antonijevic, B. Nikolic, G. Kvascev, M. Marjanovic, N. Savanovic, Hybrid CNN and XGBoost model tuned by modified arithmetic optimization algorithm for COVID-19 early diagnostics from X-ray images, Electronics 11 (22) (2022) 3798.
[48] S. Gündoğdu, Efficient prediction of early-stage diabetes using XGBoost classifier with random forest feature selection technique, Multimed. Tools Appl. (2023) 1–19.
[49] A. Asselman, M. Khaldi, S. Aammou, Enhancing the prediction of student performance based on the machine learning XGBoost algorithm, Interact. Learn. Environ. (2021) 1–20.
[50] D.D. Rufo, T.G. Debelee, A. Ibenthal, W.G. Negera, Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM), Diagnostics 11 (9) (2021) 1714.
[51] S. Abenna, M. Nahid, A. Bajit, Motor imagery based brain-computer interface: improving the EEG classification using delta rhythm and LightGBM algorithm, Biomed. Signal Process. Control 71 (2022) 103102.
[52] S. Zhang, Y. Yuan, Z. Yao, J. Yang, X. Wang, J. Tian, Coronary artery disease detection model based on class balancing methods and LightGBM algorithm, Electronics 11 (9) (2022) 1495.
[53] J. Zhang, D. Mucs, U. Norinder, F. Svensson, LightGBM: an effective and scalable algorithm for prediction of chemical toxicity–application to the Tox21 and mutagenicity data sets, J. Chem. Inf. Model. 59 (10) (2019) 4150–4158.
[54] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, Y. Miao, Review of image classification algorithms based on convolutional neural networks, Remote Sens. 13 (22) (2021) 4712.
[55] A. Darwish, D. Ezzat, A.E. Hassanien, An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis, Swarm Evol. Comput. 52 (2020) 100616.
[56] W. Lin, T. Tong, Q. Gao, D. Guo, X. Du, Y. Yang, G. Guo, M. Xiao, M. Du, X. Qu, Alzheimer's Disease Neuroimaging Initiative, Convolutional neural networks-based MRI image analysis for the Alzheimer's disease prediction from mild cognitive impairment, Front. Neurosci. 12 (2018) 777.
[57] S. Sandhiya, U. Palani, An effective disease prediction system using incremental feature selection and temporal convolutional neural network, J. Ambient Intell. Humaniz. Comput. 11 (11) (2020) 5547–5560.
[58] P. Charoenkwan, C. Nantasenamat, M.M. Hasan, M.A. Moni, B. Manavalan, W. Shoombuatong, StackDPPIV: a novel computational approach for accurate prediction of dipeptidyl peptidase IV (DPP-IV) inhibitory peptides, Methods 204 (2022) 189–198.
[59] M.M. Ali, K. Ahmed, F.M. Bui, B.K. Paul, S.M. Ibrahim, J.M. Quinn, M.A. Moni, Machine learning-based statistical analysis for early stage detection of cervical cancer, Comput. Biol. Med. 139 (2021) 104985.
[60] M.M. Ali, B.K. Paul, K. Ahmed, F.M. Bui, J.M. Quinn, M.A. Moni, Heart disease prediction using supervised machine learning algorithms: performance analysis and comparison, Comput. Biol. Med. 136 (2021) 104672.
[61] A.E. Mohamed, Comparative study of four supervised machine learning techniques for classification, Int. J. Appl. 7 (2) (2017) 1–15.
[62] Acessed on 4 October 2023 [Online]. Available, https://biopython.org/wiki/ProtParam?fbclid=IwAR1JWQK34HyW30afjY2bGLZzkq900sPU019z7KZVQtj1ocfk_v16JPBoKFI.