N-Grams Modeling for Protein Secondary Structure Prediction: Exploring Local Features and Optimal CNN Parameters

Annisa Rizqiana[1] and *Afiahayati*[2*]

[1]Master Program of Computer Science, Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia
[2]Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia

Abstract. This study explores the potential of n-gram modeling in protein secondary structure prediction. Experiments are conducted on three datasets using bigrams, trigrams, and a combination of the best n-grams with PSSM profiles. Optimal parameters for Convolutional Neural Networks (CNNs) are investigated. Results indicate that bigrams outperform trigrams in Q8 accuracy. Adding another feature, that is, PSSM, can improve model performance. Deeper convolution layers and longer convolution sizes enhance accuracy. Both bigrams and trigrams demonstrate similar performance trends, with bigrams slightly more effective. The study offers insights into local feature extraction, which is n-grams for protein modeling. These findings contribute to protein structure analysis and bioinformatics advancements, facilitating improved protein function prediction.
Keywords: Protein secondary structure prediction, n-grams, sequence labeling, convolutional neural network, bioinformatics.

1 Introduction

Proteins are essential macromolecules that play crucial roles in the functioning of living organisms. Their diverse functions include catalyzing biochemical reactions, structural elements, aiding signal transmission, and regulating gene expressions. The specific role of a protein largely depends on its unique three-dimensional structure, which is determined by the linear sequence of amino acids in its polypeptide chain [1].

One of the critical aspects of protein structure is its secondary structure, which refers to the local spatial arrangement of amino acids in the protein chain. The primary types of secondary structures include alpha helices, beta sheets, and random coils or loops. Understanding protein secondary structure is vital because it provides insights into its stability, folding, and function [2]. Wherever the 3-state is extended to an 8-state, that could provide more detailed local structure information, this method is the most often used in PSSP.

Initially, experimental techniques like X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy were used to determine protein structures. While these methods provide highly accurate results, they are laborious, time-consuming, and often require purified protein samples, which can be challenging to obtain [3]. In response to these limitations, bioinformatics has played a vital role in developing computational methods for predicting secondary structures of proteins from their amino acid sequences.

Protein secondary structure prediction (PSSP) is a fundamental task in protein science and computational biology, and it can be used to understand protein 3-dimensional (3-D) structures and, further, to learn their biological functions. This 3-d protein structure can be obtained using cryo-EM methods [4]. In PSSP, each sequence element will be predicted and produced a secondary structure label for each amino acid position in the sequence, known as the sequence labeling task. In the past decade, many methods have been propose for PSSP [2]. Some computational techniques have been use to predict protein structures, like machine learning and neural networks. A standard method used for sequence labeling is Convolutional Neural Network (CNN). CNN is one of the neural networks approach that has emerged as a powerful tool in various image and sequence analysis tasks, including natural language processing. Their ability to automatically learn hierarchical patterns and feature representations from data makes them well-suited for protein secondary structure prediction. CNN is often used in various techniques for PSSP tasks, such as DeepPrime2Sec [1], Generative Stochastic Network [2], multi-input [3], and MUST-CNN [4]. MUST-CNN solves the problem of reducing the length of protein sequences by carrying out the shift and stitch technique, where predicting protein structure requires the same input and output length.

Other than that, protein features also contribute to increasing accuracy. Various protein features like PSSM profiles and physical properties [5] have been trying to improve the study's accuracy. Some embedding methods are also applied to represent amino acid sequences in different formats, ProtVec and ELMo Embedding [1].

* Corresponding author: afia@ugm.ac.id

 https://doi.org/10.1051/bioconf/20237501008

This study will use the n-grams approach [10] to model amino acid sequences. N-grams are usually used in natural language processing tasks. In protein modeling, n-grams play a significant role in capturing short-range sequence patterns. By utilizing n-grams, which are contiguous amino acid sub-sequences, predictive models can identify recurring motifs that contribute to specific structural elements. In protein secondary structure prediction, n-grams refer to a sequence of *n* amino acids that are close together in a protein sequence, where n-grams can help identify patterns or motifs related to a particular secondary structure. However, choosing the correct value of n is critical to getting good results.

We propose a Convolutional Neural Network with n-grams protein modeling to predict secondary structure prediction for Q8 accuracy. We use CNN for PSSP because we saw this task as a sequence labeling task, and many previous studies also used CNN for their method for the PSSP task, as described above. N-grams modeling applies to CullPDB and CB513 datasets, whose *n* values are 2 and 3, commonly referred to as bigrams and trigrams. We investigate the effect of n-grams modeling on secondary structure prediction performance using metrics evaluation, that is, Q8 accuracy, precision, and recall.

This paper is organized as follows: Section 1 presents the introduction and motivation for this work, while Section 2 presents related work. Section 3 offers n-grams dataset modeling. Methods and materials are described in section 4. In section 5, we will discuss the result of the study. And section 6 concludes this paper with possible future work.

2 Related Work

On the other hand, deep learning has shown great promise in representation learning, allowing the discovery of practical features and their mappings directly from data, thereby overcoming the limitations of hand-designed features. In particular, convolutional neural networks (CNNs) have demonstrated remarkable success in image recognition and are being increasingly explored for various bioinformatics applications, including protein secondary structure prediction. By leveraging the hierarchical representations learned from data, CNNs offer the potential to better understand the local features and patterns in protein sequences, thus providing a promising avenue for enhancing the accuracy and performance of protein secondary structure prediction models [11].

Convolutional Neural Networks (CNNs) use in protein secondary structure prediction has been widely studied, with researchers employing different approaches. Among them, exploring local features using n-gram modeling has shown promise. [6] utilized CNN as a classifier for predicting protein secondary structure and compared its performance with PSSM protein profile features, along with a combination of PSSM and features extracted using a Generative Confrontation Network (GCN). Notably, GCN-based protein feature extraction significantly impacted the study.

[7] proposed a combination of CNN and SVM for protein secondary structure prediction using the CullPDB and CB513 datasets. They focused on the Shift and Stitch CNN architecture, which preserved the protein sequence length during training and output. The input dataset included orthogonal input profiles and PSSM profiles with a size of 42. [3] also employed the Shift and Stitch CNN approach, known as MUST-CNN (Multi-input Shift and Stitch CNN), for protein secondary structure prediction using one-hot encoding of protein sequences. The MUST-CNN model utilized 1-dimensional convolutions to preserve an output size equal to the input size through shift and stitch techniques, resulting in improved prediction accuracy.

Moreover, [1][3] investigated biophysical features of amino acids, including flexibility scores, instability, hydrophobicity, hydrophilicity, and surface accessibility, for protein secondary structure prediction. They also explored different amino acid embeddings, such as ProtVec and contextualized embeddings, and found that combining one-hot encoding and PSSM profiles yielded the best results.

In summary, while previous research has extensively explored CNN-based methods and various feature representations for protein secondary structure prediction, the potential of n-gram modeling in capturing local features remains an underexplored area. This study aims to fill this gap by focusing on the n-grams approach and its impact on accurately predicting protein secondary structures. By investigating n-grams as an effective method for extracting local features from protein sequences, this research seeks to contribute valuable insights to bioinformatics and protein structure analysis.

3 N-grams modeling

n-grams is the *n* character of a longer string. Regarding text processing, n-grams are used as a term for adjacent words. N-grams are used in various natural language processing applications, such as language models, sequence modeling, etc. N-grams can provide information about the occurrence and sequence patterns of units in a text, which can be used for prediction, analysis, and other text-related tasks.

N-grams refer to a sequential sequence of n units (such as words or characters) in text. Several *n*-grams include bi-grams (n = 2), tri-grams (n = 3), and others with a higher *n* value. For example, in the word "computer", the bigrams are 'co', 'om', 'mp', 'pu', 'ut', 'te', and 'er'. The trigrams are 'com', 'omp', 'mpu', 'put', 'ute', and 'ter' [8].

Because the protein sequence is the same as the text, which has a sequence pattern that can be learned for various tasks, it is a prediction task in this case. A protein's primary structure, consisting of several amino acids of various lengths, is then cut into overlapping pieces of *n*-grams. In *n*-grams where n is greater than or equal to 2, an empty padding will be added to each sequence's beginning and end so that no information is wasted. An illustration of how n-grams are applied in protein sequence shows in Table 1.

Table 1. Example of *n*-grams modeling in protein sequence.

(a) Primary structure AENFDSRELE
(b) Bigrams ‘ A’, ‘AE’, ‘EN’, ‘NF’, ‘FD’, ‘DS’, ‘SR’, ‘RE’, ‘EL’, ‘LE’, ‘E ’
(c) Trigrams ‘ AE’, ‘AEN’, ‘ENF’, ‘NFD’, ‘FDS’, ‘DSR’, ‘SRE’, ‘REL’, ‘ELE’, ‘LE ’

The amount of padding is determined by Equation 1.

$$Padding = n - (sequence\_length \ \% \ n) \quad (1)$$

4 Data and Methodology

4.1 Dataset

The protein dataset used in these experiments is filtered CullPDB and CB513 dataset from [2] study. Those datasets contain no duplicates. The training and validation sets include 80% and 20% CullPDB, respectively. However, CB513 is used as a test set. Those datasets performed n-grams modeling before they trained to CNN.

The CullPDB dataset has 5365 protein sequences, where the shortest and the longest sequence have 12 and 696, respectively. The CB513 dataset is essential for checking the performance of the model. This dataset has 514 protein sequences, the shortest length is 20, and the longest is 700. The class distribution of both datasets is imbalanced, which will affect the model's accuracy.

4.2 Methodology

The dataset will be extracted to retrieve only the amino acid sequence features and PSSM profiles. The n-grams modeling was applied to a feature set of amino acid sequences with the size of 2 (bigrams) and 3 (trigrams). The two feature sets were tested and compared, and then the best n-grams were combined with the PSSM profile feature set. These steps are illustrated in Figure 1.

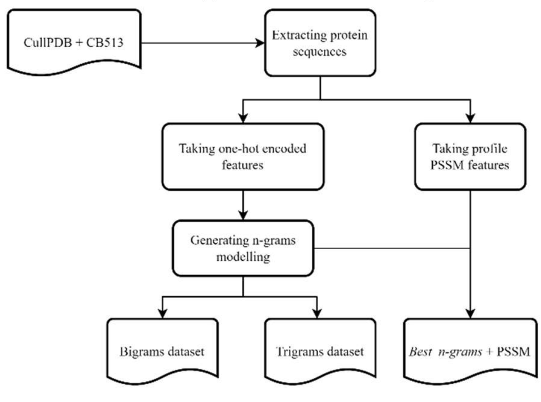

Fig. 1. Generating n-grams dataset

A detailed illustration of how to combine best n-grams with PSSP features can be seen in Figure 2. PSSM features are added at the end of n-grams features so that the total features can be calculated by $(n * i) + j$, where n is the value of n in n-grams, i is the number of feature sets produced by the best n-grams, and j is the number of PSSM feature sets.

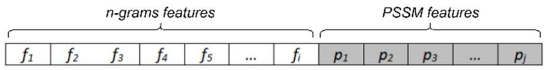

Fig. 2. Combination of n-grams and PSSM features

Shift and stitch CNN architecture, as described in the previous section, has good potential in the protein structure prediction study field because the input and output are expected to have the same length. Shift and stitch restore the length of the protein sequence by multiplying it and going through the processes of convolution and pooling. The architectural design is shown in Figure 3.

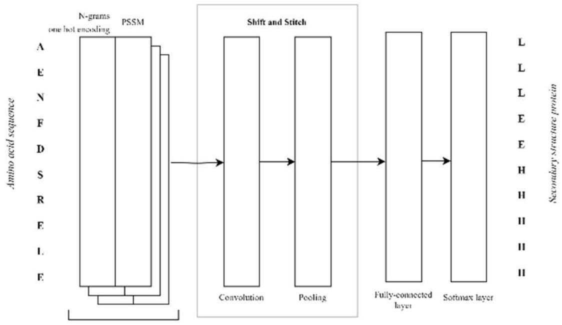

Fig. 3. Architecture for this study

The input layer will receive n-grams data and the best n-grams combination with the PSSM profile. The number of features input will differ depending on the size of n. The shape of the input layer is (700 x f), where f is the number of features. Then, in a shift layer, each amino acid sequence is duplicated and input to the convolution layer. In this layer, the two duplicate sequences will be padded so that the convolutional layer's output is not reduced. Next, the two amino acid sequences are pooled with size 2, where the two sequences will have half the length of the initial amino acid sequence, namely (350 x f), which are then combined through a shift layer to produce an output with the same length as the actual sequence length (700 x f). The result of this stage then becomes the input for the fully-connected and softmax layers, where the output layer will produce the probability prediction.

This study implemented the shift and stitch technique with TensorFlow and Keras libraries by tuning parameters to get the best result from the combination parameters. We then compare the two n-grams modeling to find the best n value for these problems.

5 Results

We conducted experiments on 3 data sets, respectively bigrams, trigrams, and a combination of the best n-grams with PSSM profile features, to find out the potential of n-gram modeling to understand local features of proteins in protein secondary structure prediction.

Table 2. Fixed parameters.

Parameter	Value
Pooling size	2
Fully connected layer	2
Dropout rate	0.2
Activation function	{ReLU, Softmax}

Table 3. Tuned parameters

Parameter	Value
Convolution layer	{2,3,4}
Convolution size	{3,5,7}
Feature maps	{32,64,128}

Table 4. Result in bigrams dataset

Conv Layer	Conv Size	Feature Map	Q8 Accuracy (%)
2	3	32	47.53730903
2	3	64	48.58373149
2	3	128	48.70288445
2	5	32	49.71273521
2	5	64	50.96089188
2	5	128	50.89482687
2	7	32	52.40252463
2	7	64	53.05137734
2	7	128	52.75172536
3	3	32	51.01633929
3	3	64	51.39857252
3	3	128	50.07373326
3	5	32	52.52875597
3	5	64	51.10599894
3	5	128	52.24444051
3	7	32	53.70258951
3	7	64	52.32348257
3	7	128	53.99398337
4	3	32	51.67462986
4	3	64	52.7823984
4	3	128	51.42452663
4	5	32	53.52209049
4	5	64	52.96643662
4	5	128	52.34117855
4	7	32	53.77809237
4	7	64	54.01993747
4	7	128	53.88780747

Before we analyze the potential of n-grams modeling, we first search for the optimal CNN parameters. We divide parameters into two types: fixed parameters, where the parameter values will not change during each training process, and tuned parameters. The set parameters include pool size, the number of fully-connected layers, dropout rate, and the activation function (see Table 2). While the tuned parameter will combine three parameters, including the depth of the convolution layer, the size of the convolution, and the number of feature maps (see Table 3). These three parameters will produce 27 models with 27 parameter combinations for each training dataset.

In this study, we calculate the Q8 accuracy for each experiment. We find that between bigrams and trigrams, bigrams produce the highest accuracy. So, we combine the bigrams dataset with PSSM profile features. The convolution layer parameter increases the Q8 accuracy in almost every depth layer, where the deeper the convolution layer, the higher the resulting accuracy. Details results are presented in Table 4, Table 5, and Table 6, respectively, the bigrams, trigrams, and combination of bigrams and PSSM profiles features.

Table 5. Results in trigrams dataset

Conv Layer	Conv Size	Feature Map	Q8 Accuracy (%)
2	3	32	50.24243497
2	3	64	50.2105822
2	3	128	50.31911756
2	5	32	51.40211172
2	5	64	51.60384593
2	5	128	51.67816906
2	7	32	52.61015749
2	7	64	53.06671386
2	7	128	51.79496254
3	3	32	52.18309444
3	3	64	49.38123046
3	3	128	52.1925323
3	5	32	52.58302365
3	5	64	52.47802749
3	5	128	52.4603315
3	7	32	53.37580369
3	7	64	53.21182092
3	7	128	51.72417861
4	3	32	53.14339645
4	3	64	52.98885153
4	3	128	53.20946145
4	5	32	53.55394325
4	5	64	53.22833717
4	5	128	53.36164691
4	7	32	53.51855129
4	7	64	53.69905032
4	7	128	53.61528933

From these results, it was observed that in the three datasets that were trained, the layer depth that produces the best Q8 accuracy is at depth 4. Likewise, with the convolution size parameter, where the length of the parameter size is directly proportional to the increase in Q8 accuracy. In this case, it can be said that the best kernel length is 7.

Unlike the two parameters mentioned, feature maps cannot provide consistent results for increasing accuracy. It can be observed from the three tables presented the number of feature maps determined by three values gives variations in the results of different Q8 accuracy. Therefore, the best value for the number of feature maps cannot be determined; it is necessary to try a combination of various feature maps to obtain the best Q8 accuracy.

Table 6. Results in bigrams combined with the PSSM dataset

Conv Layer	Conv Size	Feature Map	Q8 Accuracy (%)
2	3	32	62.8136613
2	3	64	63.22066891
2	3	128	63.00123872
2	5	32	62.30991565
2	5	64	63.07202265
2	5	128	63.49790598
2	7	32	64.20338583
2	7	64	64.74842211
2	7	128	64.8215655
3	3	32	64.10310859
3	3	64	63.71733616
3	3	128	63.62295759
3	5	32	61.24815667
3	5	64	65.02211998
3	5	128	64.57264201
3	7	32	65.11177963
3	7	64	63.58992509
3	7	128	65.40435321
4	3	32	64.56792308
4	3	64	65.21913526
4	3	128	63.95800153
4	5	32	58.10299062
4	5	64	63.23364596
4	5	128	64.26355217
4	7	32	65.68512948
4	7	64	65.39963428
4	7	128	64.07361529

As we explained earlier, the number of each data class in the dataset is imbalanced, especially in classes "I" and "B" with relatively small data. This will affect the precision and recall evaluation values of each class. Table 7 and 8 presents the precision and recall for each dataset, respectively.

Table 7. The precision of the best accuracy in each data

	Precision		
	bigrams	**trigrams**	**bigrams + PSSM**
L	0.46	0.45	0.53
B	0	0	0.75
E	0.54	0.58	0.67
G	0.29	0.28	0.39
I	0	0	0
H	0.63	0.60	0.81
S	0.39	0.74	0.47
T	0.38	0.39	0.50

The table shows that the class labels 'L', 'E', and 'H' produce higher precision and recall than other classes due to their appearance in large datasets. Whereas class labels 'B' and 'I' produce very little precision and recall, leading to zero because their appearance in the dataset is very small. This explains that the model failed to predict several minor classes but succeeded in predicting several major classes. This might happen because a minor class with a small number of classes does not have unique features that can distinguish it from other features.

Table 8. Recall of the best accuracy in each data

	Recall		
	bigrams	**trigrams**	**bigrams + PSSM**
L	0.47	0.50	0.58
B	0	0	0.003
E	0.66	0.59	0.84
G	0.03	0.03	0.17
I	0	0	0
H	0.82	0.85	0.91
S	0.03	0.01	0.12
T	0.39	0.36	0.48

We also provide the results of the n-grams modeling we used (bigrams and trigrams). For bigrams, the result of precisions ranges from 0.29 to 0.63 for different secondary structure classes, where the precision (0.63) is achieved for the "H" class, and the lowest precision (0) is obtained for the "B" and "I" classes. While in trigrams, the result of precisions ranges from 0.28 to 0.74 for different secondary structure classes, where the highest precision (0.74) is achieved for the "S" class, and the lowest precision (0) is obtained for the class "I" and "B" classes. Recall for bigrams ranges from 0.03 to 0.82, while trigrams range from 0.01 to 0.85. The highest recall for both n-grams is the "H" class and the lowest recall for are "B" and "I" classes.

6 Discussion

The results of our study show that the percentage of accuracy of the results obtained is consistently below 70%. We consider several things that affect the performance of the models we build.

In our analysis, we compare the use of bigrams and trigrams in the context of our model. The average training loss on bigrams shows slightly better results than on trigrams, with 0.4017 for bigrams and 0.4051 for trigrams. Along with that, we also consider the variability and complexity of the data. Variability in biological data refers to the natural variation in data whereby proteins with similar amino acid sequences can have different secondary structures. The complexity of the data used is multidimensional, thus providing a challenge in establishing an accurate model. We see that the use of trigrams may lead to more specific patterns but may cause the model to be less tolerant of variations in the elements in the data, particularly changes in amino acid sequence. This also affects the model in its less generalization ability. In this context, using bigrams results in a more adaptive model to a wider range of variations. This is shown from the evaluation Q8 accuracy results, which is 54.019% in bigrams and 53.699% in trigrams. Based on these considerations, we decided that better performance by the model resulted from using bigrams. Therefore, bigrams were combined with PSSM and increased the Q8 accuracy up to 65.69%.

Overall, both bigrams and trigrams modeling demonstrate similar performance trends, with bigrams performing slightly better than trigrams in precision and recall for most secondary structure classes. This indicates that the Bigrams method was more effective in correctly predicting the presence of specific secondary structure elements in protein sequences. Our approach may not fully address this problem, but these results provide new insights into the performance of n-grams in PSSP tasks with the constructed model. However, it is important to note that both methods face challenges in predicting secondary structure protein.

7 Conclusion

In conclusion, this study aimed to explore the potential of *n*-gram modeling in understanding local features of proteins for protein secondary structure prediction using a Convolutional Neural Network (CNN). Experiments were conducted on two datasets using bigrams, trigrams, and a combination of the best n-grams with PSSM features. The results showed that bigrams slightly outperformed trigrams regarding accuracy for the protein secondary structure prediction. So, bigrams were combined with PSSM profile features to further improve accuracy.

We conclude that from the analysis of n-grams, bigrams are more effective and have better accuracy than trigrams in protein secondary structure prediction tasks, especially in the model we build. The Q8 accuracy of both n-grams was less than 70%. However, it also underscores the need for further research and optimization, particularly for minor classes with limited instances, to achieve more balanced and accurate predictions. Addressing the class imbalance challenge highlighted within the study could involve employing techniques such as resampling methods, applying ensemble methods, and considering algorithms inherently robust to imbalanced classes.

References

1. P. D. Sun, C. E. Foster, and J. C. Boyington, "Overview of protein structural and functional folds." *Curr. Protoc. Protein Sci.*, vol. Chapter 17, pp. 1–189, 2004, doi: 10.1002/0471140864.ps1701s35.
2. Q. Jiang, X. Jin, S. J. Lee, and S. Yao, "Protein secondary structure prediction: A survey of the state of the art," *J. Mol. Graph. Model.*, vol. 76, pp. 379–402, 2017, doi: 10.1016/j.jmgm.2017.07.015.
3. Y. F. Huang and S. Y. Chen, "Extracting physicochemical features to predict protein secondary structure," *Sci. World J.*, vol. 2013, 2013, doi: 10.1155/2013/347106.
4. K. Murata and M. Wolf, "Cryo-electron microscopy for structural analysis of dynamic biological," *BBA - Gen. Subj.*, vol. 1862, no. 2, pp. 324–334, 2018, doi: 10.1016/j.bbagen.2017.07.020.
5. E. Asgari, N. Poerner, A. C. McHardy, and M. R. K. Mofrad, "DeepPrime2Sec: Deep Learning for Protein Secondary Structure Prediction from the Primary Sequences," *bioRxiv*, 2019, doi 10.1101/705426.
6. J. Zhou and O. G. Troyanskaya, "Deep supervised and convolutional generative stochastic network for protein secondary structure prediction," in *31st International Conference on Machine Learning, ICML 2014*, 2014, vol. 2, pp. 1121–1129.
7. S. I. Jalal, J. Zhong, and S. Kumar, "Protein Secondary Structure Prediction using Multi-input Convolutional Neural Network," in *Conference Proceedings - IEEE SOUTHEASTCON*, 2019, vol. 2019-April. doi: 10.1109/SoutheastCon42311.2019.9020333.
8. Z. Lin, J. Lanchantin, and Y. Qi, "MUST-CNN: A multilayer shift-And-stitch deep convolutional architecture for sequence-based protein structure prediction," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016, pp. 27–34.
9. B. Zhang, J. Li, and Q. Lü, "Prediction of 8-state protein secondary structures by a novel deep learning architecture," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–13, 2018, doi: 10.1186/s12859-018-2280-5.
10. P. Majumder, M. Mitra, and B. B. Chaudhuri, "N-gram : a language-independent approach to IR and NLP," in *ICUKL*, 2002, vol. 2.
11. S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Brief. Bioinform.*, vol. 18, no. 5, pp. 851–869, 2017, doi: 10.1093/bib/bbw068.
12. Y. Zhao, H. Zhang, and Y. Liu, "Protein secondary structure prediction based on generative confrontation and convolutional neural network," *IEEE Access*, vol. 8, pp. 199171–199178, 2020, doi: 10.1109/ACCESS.2020.3035208.
13. V. M. Sutanto, Z. I. Sukma, and A. Afiahayati, "Predicting Secondary Structure of Protein Using Hybrid of Convolutional Neural Network and Support Vector Machine," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 1, pp. 232–243, 2020, doi: 10.22266/IJIES2021.0228.23.
14. W. Cavnar and J. Trenkle, "N-Gram-Based Text Categorization," 2001.