Deep learning for protein secondary structure prediction: Pre and post-AlphaFold

Dewi Pramudi Ismi\textsuperscript{a,b}, Reza Pulungan\textsuperscript{a,*}, Afiahayati\textsuperscript{a}

\textsuperscript{a}Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia
\textsuperscript{b}Department of Informatics, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Abstract

This paper aims to provide a comprehensive review of the trends and challenges of deep neural networks for protein secondary structure prediction (PSSP). In recent years, deep neural networks have become the primary method for protein secondary structure prediction. Previous studies showed that deep neural networks have uplifted the accuracy of three-state secondary structure prediction to more than 80%. Favored deep learning methods, such as convolutional neural networks, recurrent neural networks, inception networks, and graph neural networks, have been implemented in protein secondary structure prediction. Methods adapted from natural language processing (NLP) and computer vision are also employed, including attention mechanism, ResNet, and U-shape networks. In the post-AlphaFold era, PSSP studies focus on different objectives, such as enhancing the quality of evolutionary information and exploiting protein language models as the PSSP input. The recent trend to utilize pre-trained language models as input features for secondary structure prediction provides a new direction for PSSP studies. Moreover, the state-of-the-art accuracy achieved by previous PSSP models is still below its theoretical limit. There are still rooms for improvement to be made in the field.

© 2022 The Author(s). Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

1. Introduction .......................................................................................................................... 6272
2. General description .............................................................................................................. 6273
3. Data ..................................................................................................................................... 6274
   3.1. Training data .................................................................................................................. 6274
   3.2. Test data ....................................................................................................................... 6274
4. PSSP in pre-AlphaFold publication ..................................................................................... 6275
   4.1. Features ....................................................................................................................... 6275
   4.2. Model architectures ...................................................................................................... 6276
4.3. Performance .................................................................................................................... 6278
5. New approaches for feature generation ............................................................................... 6279
6. AlphaFold and its impact .................................................................................................... 6280
7. PSSP in post-AlphaFold publication ................................................................................... 6281
   7.1. Taking MSA as direct input to the PSSP model ............................................................ 6281
   7.2. Improving the quality of sequence profiles ................................................................. 6281
   7.3. Exploiting protein language models (LMs) as input features ...................................... 6281
   7.4. PSSP studies aiming at other objectives ...................................................................... 6282
8. Summary and outlook ......................................................................................................... 6282
  CRediT authorship contribution statement .......................................................................... 6283
  Declaration of Competing Interest ...................................................................................... 6283

* Corresponding author.
E-mail addresses: dewipramudi85@mail.ugm.ac.id (D.P. Ismi), pulungan@ugm.ac.id (R. Pulungan), afia@ugm.ac.id (Afiahayati).

https://doi.org/10.1016/j.csbj.2022.11.012
2001-0370© 2022 The Author(s). Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
1. Introduction

Proteins play important roles for living organisms due to their diverse functions, for example, acting as a catalyst in cell metabolism, playing an essential role in DNA replication, forming cell structures, forming living tissues, and constructing antibodies for the immune system. Protein has four different levels of structure: primary structure, secondary structure, tertiary structure, and quaternary structure. The protein’s primary structure is the amino acid sequence composing its polypeptide chain. The protein’s secondary structure is a local conformation formed within the polypeptide chain due to the hydrogen bonds between atoms in the backbone. The tertiary structure of protein exposes the three-dimensional structure of the protein in the physical space. The quaternary structure of protein is made up of multiple polypeptide chains that come together.

Proteins function differently from one another due to variations in their structures, mainly due to the folds that make up varying tertiary structures [1,2]. Since the function of a particular protein is influenced by its tertiary structure, understanding the protein’s tertiary structure is necessary to reveal its functionality. Hence, protein tertiary structure prediction is a crucial task in structural bioinformatics. Prior to the invention of AlphaFold [3], predicting protein tertiary structure directly from its primary structure was a challenging problem [4–7]. Thus, protein structure prediction was divided into subsidiary tasks that are easier to accomplish and beneficial for ultimate protein tertiary structure prediction. Researchers attempted to build models that solve these subsidiary tasks.

Protein secondary structure prediction (PSSP) is one of the subsidiary tasks of protein structure prediction and is regarded as an intermediary step in predicting protein tertiary structure. If protein secondary structure can be determined precisely, it helps to predict various structural properties useful for tertiary structure prediction. For example, secondary structures affect structural folds in the polypeptide chain [8,9], and the same protein secondary structures define similar folds in the protein’s polypeptide chain [10]. PSSP started when Pauling et al. [11] suggested α-helices and β-sheets as the dominant conformations even before the first protein structure was revealed. The first protein structure was revealed using an X-ray by Kendrew et al. [12]. PSSP has gone through five generations since then [13]. Table 1 shows the generations and their respective traits.

Leveraging evolutionary information as an input feature and utilizing deep neural networks (DNN)-based architectures has become the primary approach done by PSSP researchers in recent years. Three main reasons that warrant the use of DNN in PSSP are: (1) the increasing number of protein sequences in the Protein Data Bank (PDB), (2) the need to understand and capture long-range interactions in protein sequences, and (3) the capability of DNN to observe the underlying characteristics and hidden patterns in protein sequences.

Considering the immense usage of DNN in PSSP, we aim to provide a comprehensive literature review on PSSP studies that leverage DNN. Although there have been several publications on the topic, such as [39–42], several key points differentiate this review from previous reviews, namely:

1. This review focuses on the implementation and architecture details of DNN methods utilized in PSSP;
2. This review discusses various approaches that have been applied to improve PSSP accuracy, such as multi-task learning, ensemble learning, iterative learning, attention mechanism, and language modeling;
3. This review discusses the more recent and advanced methods employed in PSSP, such as the involvement of pre-trained language models (LMs) and knowledge distillation; and
4. This review discusses the progress and perspectives of PSSP in two different time plots, namely before and after the invention of AlphaFold.

This review is organized as follows: general description, data, PSSP in pre-AlphaFold publication, new approaches for feature generation, AlphaFold and its impacts, PSSP in post-AlphaFold publication, and summary and outlook.

### Table 1

| Generation | Year       | Traits                                                                 | PSSP studies                  |
|------------|------------|------------------------------------------------------------------------|-------------------------------|
| 1st        | 1960s, 1970s | Utilizing statistical properties of amino acids to determine SS.       | [14–16]                       |
| 2nd        | 1980s, 1990s | Utilizing statistical analysis of residue window (several adjacent residues) as the features to predict SS. Employing graph theory, neural networks, and nearest neighbour algorithms. | [17–21]                       |
| 3rd        | 1990s, 2000s | Utilizing evolutionary information derived from multiple sequence alignment (MSA) as input feature for PSSP models. Achieving 77% of prediction accuracy (Q3). | [22, PHD [23], PSIPRED [24], JPRED [25]] |
| 4th        | 2000s      | Utilizing auxiliary input features besides the evolutionary information, such as conserved domain profiles, frequent patterns, physico-chemical properties of amino acids, predicted torsion angles, predicted residue contact maps, and predicted residue solvent accessibility (RSA). Small improvements in Q3 accuracy have been made although they did not go beyond 80%. | [26–31]                       |
| 5th        | 2010s, 2020s | Utilizing sophisticated deep learning architectures. The Q3 accuracy achieved exceeds 80%. | SPIDER [32], RaptorX-Property [33], SPIDER3 [34], MUFOLD-SS [35], SPOT-1D [36], NetSurfP-2.0 [37], cCRNN [38], SecNet [13] |
2. General description

PSSP can be seen as a classification problem that categorizes an amino acid residue to the type of secondary structure in which it is located. A typical PSSP model takes a sequence of amino acids as input and returns a sequence of the corresponding secondary structure (see Fig. 1). There are two different types of PSSP: three-state PSSP and eight-state PSSP. In three-state PSSP, the secondary structure elements consist of helix (H), sheet (E), and coil (C). Helix and sheet are the two main conformations suggested by Pauling et al. [11]. Coil (C) denotes an amino acid that does not fit both H and E. In eight-state PSSP, proposed by Kabsch and

Fig. 1. A PSSP model takes sequences of amino acids as an input and produces the sequences of corresponding secondary structure elements.

Fig. 2. The general framework of PSSP models: two phases in model development, namely training and evaluation. The training dataset is used to build the model, while the test dataset is used to confirm the performance of the trained model.
Sander [43], the secondary structure elements consist of α-helix (H), 3_10-helix (G), parallel/anti-parallel β-sheet conformation (E), isolated β-bridge (B), bend (S), turn (T), β-helix (I), and coil (C). The three-state PSSP is a coarse-grained classification of secondary structure elements, whereas the eight-state PSSP provides more specific traits of the secondary structure elements. Several transformation rules to map the eight-state secondary structure to the three-state secondary structure are proposed by researchers, including DSSP [43], DEFINE [44], and Rost and Sander [45].

In 2020, four-state and five-state PSSP alternatives were proposed by Shapovalov et al. [13]. The reason for proposing four-state and five-state PSSP was because isolated β-bridge (B) and bend (S) have a small number of samples and low true-positive rates. In five-state PSSP, B and S are considered as C, whereas in four-state PSSP, B, S, and G are considered as C. Moreover, 75% of π-helix (I) was located at the beginning or the end of an α-helix structure (H), so it was categorized as H.

A PSSP model takes a sequence of amino acids as input. Features representing the sequence of amino acids are generated and used to train the PSSP model. Like other machine learning-based systems, DNN-based PSSP models are developed through the training and evaluation phase. The general framework of DNN-based PSSP is shown in Fig. 2. Based on the framework depicted in the figure, three aspects play essential roles in producing high prediction accuracy: the training data, the features representing the sequence of amino acids, and the model’s architecture. A typical PSSP study usually tackles one or more aspects.

A DNN-based PSSP model incorporates one or more deep learning methods in its architecture, such as recurrent neural networks (RNN), including long short-term memory (LSTM) and gated recurrent unit (GRU), convolutional neural networks (CNN), inception networks, and other networks. Although most previous PSSP studies focused on developing novel architectures, several PSSP studies focused on different objectives. Several studies focused on proposing new feature representations, such as new amino acid encoding schemes [46–48], and other features such as protein language models [49,50]. Several studies focused on evaluating the evolutionary information used as input of the PSSP model, for example [51–55]. Other studies proposed novel architectures as well as input features, for example [36,56]. Besides the model’s architecture and features, several studies also proposed new training and test datasets that eliminate data redundancy such that the models trained using these datasets can achieve a good performance [13,57,58].

The performance of PSSP models is usually assessed by predicting the secondary structures of benchmark test datasets. It is measured by the prediction accuracy, namely the percentage of correctly classified amino acid residues in the test datasets. Q3 denotes the accuracy of three-state PSSP, whereas Q8 denotes the accuracy of eight-state PSSP. In addition to accuracy, several PSSP studies also used other standard classification performance metrics, such as precision, recall, and F1 score. PSSP model performance is also measured by segment overlap (SOV). SOV calculates segment-based accuracy that tolerates false prediction at the segment’s boundary but penalizes false prediction in the middle of the segment [59]. In 1999, Zemla et al. [60] proposed SOV99, which normalizes the SOV score such that the final SOV score is on a percentage scale. SOV is unable to extend allowance when more residues can be predicted correctly in a particular segment. This issue becomes the drawback of SOV and SOV_Refine was proposed by Liu and Wang [61] to tackle the issue. Although less prevalent in PSSP, Kappa Performance was used to measure the performance of the PSSP model proposed by [62], and Matthew’s correlation coefficient [63] was used in [64].

3. Data

3.1. Training data

The development of DNN-based PSSP models requires two different datasets: training and test datasets. There are two different fashions with which the researchers obtain training and test datasets. The first is to collect the datasets from the protein database by themselves. The second is to use the benchmark datasets previously collected by other researchers. Researchers self-collected datasets from the protein data bank (PDB) [65] to build their PSSP models [13,34,35,37,47,53,56,66–75]. In this case, the size of culled datasets varies depending on the date when the data collecting is performed.

Protein data used to train the DNN-based PSSP models are typically obtained using the PISCES CullPDB server [76]. A sequence identity cutoff of 25–30% usually applies. Besides using the PISCES CullPDB server, protein sequences are also obtained from ASTRAL [77]. Several PSSP studies used datasets extracted from ASTRAL including [78,68,79]. Proteins from the Genbank database [80] were used as training data to build the PSSP model developed by Xavier and Thirunavukarasu [81]. Li et al. [82] predicted the secondary structure of transmembrane proteins and took the protein sequences from the OPM database [83].

Instead of performing self data assembly, some researchers prefer to use existing datasets from previous researchers. Several well-known benchmark datasets have been used to train DNN-based PSSP (see Table 2).

AlQuraishi [57] published the ProteinNet dataset, a standardized dataset for training and evaluating PSSP models. It was motivated by the availability of standard datasets in computer vision enabling researchers to build models and evaluate their performances using standardized data. The ProteinNet dataset consists of protein sequences, MSAs, position-specific scoring matrices (PSSMs), and training-validation-test splits. The dataset was designed to support the critical assessment of protein structure prediction (CASP) [87]. AlQuraishi [57] collected all protein sequences available in the protein databases before the date of each CASP challenge commencing. The dataset is designed to provide the complete set of protein sequences needed to build PSSP.

Table 2

| Dataset         | Notes                                      | Collected by |
|-----------------|--------------------------------------------|--------------|
| CullPDB 6125    | The dataset consists of 6,125 proteins.    | [76]         |
| CB6133          | The dataset consists of 6,133 proteins obtained from CullPDB server but it contains duplicates. | [84]         |
| CB6133−filtered | Result after filtering CB6133 dataset for redundancy with CB813 [65]. | [84]         |
| CullPDB 5926    | The dataset is the CB6133 dataset with the duplicates removed. | [84]         |
| CB513           | The dataset is a combination of CB396 and RS126 with removal of 9 proteins with 3D score > 5. | [85]         |
| CB396           | The dataset consists of non-redundant proteins that are different from proteins in RS126 [45]. | [85]         |
| ASTRAL          | The dataset contains 59,514 proteins.     | [77]         |
| TR4590          | The dataset consists of 4,590 proteins taken from CullPDB server. | [32]         |
| TR9993          | The dataset consists of 9,993 proteins taken from CullPDB server in 2017. | [74]         |
| 4Prot           | The dataset consists of 11,765 proteins collected from the DSSP database [43] in 2008. | [86]         |
| SPOT-1D         | The dataset consists of 10,029 proteins obtained from CullPDB server (2017). | [36]         |
models to solve the respective CASP challenge. For example, the dataset provides protein sequences and structures available in the databases before the date of the CASP10 challenge to train the PSSP models to be tested on CASP10.

Although there are various benchmark datasets for PSSP model training, previous PSSP researchers tend to use different training datasets to maximize the performance of their PSSP models, i.e., using a combination of several benchmark datasets or culling the training dataset from the protein database by themselves.

### 3.2. Test data

Although PSSP researchers may take the test data from the same source as the training data by culling a portion of the training data as the test dataset, several independent benchmark test datasets are widely used to compare the performance among PSSP models. CASP datasets from the biennial CASP events [87] are notable benchmark test datasets for PSSP studies. There have been 14 CASP events conducted to date; hence there are 14 CASP datasets publicly available [87]. Besides CASP datasets, several benchmark test datasets used to evaluate PSSP performances are shown in Table 3.

### 4. PSSP in pre-AlphaFold publication

This section focuses on the DNN-based PSSP models released within five years before the publication of AlphaFold (2016–2021). The discussion covers the features representing amino acid sequences, the architecture of PSSP models, and the performance achieved by those models.

#### 4.1. Features

The features representing amino acid sequences can usually be divided into three types: standard features, single-sequence, and other features.

**Standard features.** The standard input features representing an amino acid sequence in DNN-based PSSP models are the one-hot encoding of amino acids, sequence profiles, and physicochemical properties of amino acids. A one-hot encoding of amino acids is a matrix of size $n/20$, where $n$ denotes the number of amino acids in the sequence, and 20 denotes the number of types of amino acids. Matrix entries corresponding to the amino acid types are filled with 1, while the remaining entries are filled with 0.

The sequence profile is derived from MSA and represents evolutionary information of the sequence. Two different sequence profiles are commonly used in previous PSSP works: position-specific scoring matrix (PSSM) and hidden Markov model (HMM) profiles. PSSM exhibits the probability of each amino acid residue at each position in the sequence based on the MSA. On the other hand, HMM profile offers state transition probability and emission probability as additional information in the sequence profile. PSSM is usually obtained using PSI-BLAST [92], while the HMM profile is usually obtained using HHBlits [93]. Besides PSI-BLAST and HHBlits, MMSeq2 [94], an optimized search for massive datasets, is also used to generate sequence profiles. The details of one-hot encoding of amino acids and PSSP are depicted in Fig. 3.

**Table 3**

| Dataset      | Notes                                           | Collected by |
|--------------|------------------------------------------------|--------------|
| CB513        | The dataset is a combination of CB396 and RS126 with the removal of 9 proteins with SD score $>5$. | [85]          |
| RS126        | The dataset consists of 126 non-homologous proteins. | [45]          |
| 2SPDB        | The dataset contains 1,673 proteins with sequence homology of 25%. | [88]          |
| EVA set      | The dataset contains 218 protein sequences collected in 2001. | [89]          |
| MANESH       | The dataset contains 215 protein chains.       | [90]          |
| TS115        | The dataset contains 115 protein chains released after January 2016. | [39]          |
| TS1199       | The dataset contains 1,199 proteins deposited in PDB before July 2015. | [74]          |
| TS1250       | The dataset contains 1,250 proteins deposited in PDB after July 2015. | [74]          |
| TS-Hard      | The dataset is subset of TS1250, contains 280 sequences with potential homologous proteins removed (E value cutoff of 0.1). | [74]          |
| TEST2016     | The dataset contains 1,213 proteins collected between June 2015 and February 2017. | [36]          |
| TEST2018     | The dataset contains 230 proteins collected between January and July 2018. | [36]          |
| CAMEO        | The dataset contains 93 proteins.              | [91]          |

**Fig. 3.** One hot encoding of amino acids and position specific scoring matrix (PSSM).
Sequence profiles have been used as fundamental features for PSSP models. A sequence profile is generated based on the known protein sequences in the target dataset, i.e., UniRef90. With the significant growth of the number of known protein sequences in the target dataset, the computational resources and time needed to perform PSSP, especially for generating sequence profiles, also increase. Juan et al. [95] proposed an approach to increase the speed of PSSP without compromising accuracy. The approach was carried out by reducing the size of the target dataset by taking a random sampling and reducing the sequence homology of the target dataset to 25% of the sequence identity. This study showed that reducing the size of the target dataset decreased the time cost of PSSP, whereas reducing the sequence homology of the target dataset improved the complexity of the generated PSSM and enhanced the PSSP accuracy.

Several studies performed critical appraisals on the use of sequence profiles in PSSP. Urban et al. [55] performed a study to critically evaluate the role of sequence profiles in the performance of PSSP models. Sequence profiles have shown the ability to increase the performance of PSSP models. A significant accuracy gap exists between the PSSP models using sequence profiles and those using only amino acid sequences. However, its underlying reason has yet to be clearly explained because only amino acid sequences play a role when protein folds are constructed. Moreover, the study was also based on the observation that although training and test datasets used in the prediction have less than 25% of sequence identity, sequences in training and test datasets are likely to come from the same family [96–98]. In that case, a high profile similarity between training and test datasets occurs, resulting in evaluation bias. The prediction accuracy of models utilizing sequence profiles is enhanced by the redundancy found in the sequence profiles of the training and test datasets which lead to an invalid evaluation result. Urban et al. [55] also proposed an evaluation protocol, named EVALPro, to measure the performance of the PSSP models with the assessment of profile similarity between training and test proteins. EVALPro employed Gaussian process regression (GPR) to portray the relationship between the profile similarity of training and test proteins and the accuracy gained.

In addition to the one-hot encoding of amino acids and the sequence profiles, the physicochemical properties of amino acids are commonly used as an input feature for the PSSP model. The physicochemical properties include the steric parameter (graph shape index), polarizability, normalized Van der Waals volume, hydrophobicity, isoelectric point, helix probability, and sheet probability. The values of those parameters usually are those specified in Table 1 of [28].

**Single-sequence prediction.** Several studies have begun to develop PSSP models that only accept one-hot encoding of amino acids and do not use sequence profiles. It is motivated by the argument that protein structures are only influenced by the amino acid sequences and not by evolutionary information. The PSSP models in this category include SPIDER3-Single [74], ProteinUnet [99], SPOT-1D-Single [100], and S4PRED [70]. With only using amino acid encoding, the highest reported Q3 accuracy is 75.3%, achieved by S4PRED on the CB513 dataset [70].

**Non-standard features.** Besides the standard and sequence-only features, previous PSSP studies also used non-standard features and evaluated their impacts on the performance of the PSSP model. Hanson et al. [36] proposed a predicted contact map [101] as an additional input feature in the PSSP model and showed that using a contact map successfully increases accuracy. A conservation score was proposed as an input feature for eCRRNN [38]. However, the effect of this conservation score on eCRRNN accuracy was not mentioned.

### 4.2. Model architectures

Various deep learning methods have been employed in the architecture of previous PSSP models. We collected all the PSSP publications, including journal papers and conference proceedings during 2016–2021 and we summarized the usage of popular deep learning methods in **Fig. 4**. The chart shows that CNN and RNN (in-
cluding LSTM and GRU) are the two most dominant methods implemented in DNN-based PSSP models.

**Extracting local contexts using convolutional networks.** A protein secondary structure element is a local conformation formed in the polypeptide chain of the protein, composed of several amino acids. In physical space, the amino acids that make up a single secondary structure element interact with each other. From the computational point of view, these interactions, in the form of hydrogen bonds, are often invisible in the input features fed to PSSP models. Therefore, identification of local contexts/interactions on amino acid sequences needs to be done by PSSP models, and CNN has been extensively used in previous PSSP models for this purpose.

Both one-dimensional (CNN-1D) and two-dimensional CNN (CNN-2D) have been employed in previous PSSP models. CNN-1D is used more frequently since the input of a PSSP model is in the form of sequences. However, several models used CNN-2D in their architectures. CNN-2D is used in PSSP model architecture to extract temporal and spatial features of the input sequences better. Feature vectors (PSSM and one-hot encoding) of a fixed length residue window were employed in [102–104,82] as input to the two-dimensional CNN.

DeepACLSTM [105] used asymmetric convolutional filters, including two different filters, namely $1 \times 2d$ and $k \times 1$, to extract the local dependency feature of the amino acid sequence. These asymmetric convolutional filters are implemented using CNN-1D and CNN-2D. Shapovalov et al. [13] proposed a PSSP model that employed four stacked convolutional layers in its architecture. Jalal et al. [106] used multi-input CNN layers and merged the convolution outputs of each input channel.

Several studies proposed modified CNN architectures. Lin et al. [107] implemented multilayer shift and stitch in deep convolutional networks for their PSSP model to accelerate training and testing time. CNNH_PSS [108] used a highway to connect two adjacent convolutional layers to send information from the first layer to the second layer. Long and Tian [73] proposed a PSSP model with context convolutional networks, in which standard convolutional and dilated convolutional operations were joined. OCLSTM [68] optimized the parameters used in CNN using Bayesian optimization. IGRED [66] combined CNN and graph convolutional networks.

Various architectures derived from standard CNN, such as ResNet, inception networks, U-shape networks, and fractal networks, were also employed in previous PSSP models. SPOT-1D [36] and SPOT-1D-Single [100] exploited residual networks in their architectures. Residual networks were also used in [109]. Inception networks have been utilized in several PSSP models, such as MUFOLD-SS [109] and SAINT [110]. DNSSS [67] developed several model architectures and used various types of convolutional networks, including standard CNN, ResNet (residual block), InceptionNet (inception block), convolutional residual memory (CRM), and FractalNet. The U-shape networks have been utilized in ProteinUNet [99].

The kernel size used in convolutional layers in a PSSP model varies, i.e., 3, 5, 7, and 11. Previous researchers determined the most appropriate kernel size by experimentation. It means that the kernel size used in PSSP models is not standardized, and different architectures may use different kernel sizes. SecNet [13] was built with stacked CNN layers. Experimentation on SecNet found that a kernel size of 7 achieved the highest prediction accuracy, and lower accuracy was achieved using larger and smaller kernel sizes. The kernel size below 7 has worse performance than that above 7. It was probably because the hydrogen bonds in the helix are formed between $3, 4, 5$ residues apart. Moreover, the average length of sheets is about six residues [111]. Although Shapovalov et al. [13] did not recommend kernel size below 7, several previous works, including [38,35], used kernel sizes of 3 and 5 and achieved their best performances. Li and Yu [112] employed multiscale CNN layers with a kernel size of 3, 7, and 11. The three feature maps obtained are concatenated together as the local contextual feature vector. Although the kernel sizes mentioned above are widely used in previous PSSP models, larger kernel sizes are used in NetSurfP-2.0 [37] (129 and 257) and OPUS-TASS [56] (11, 21, 31, 41 and 51). Besides the kernel size, the number of filters used in convolutional layers in previous PSSP models also varies.

**Capturing long-range interactions using RNN.** In addition to identifying the local context/interactions in amino acid sequences, a PSSP model must also identify long-range interactions between amino acid residues. In sheet secondary structure, the interactions of amino acid residues (forming the hydrogen bonds) occur between distant amino acids. RNN is widely used in PSSP models to identify these long-range interactions. Furthermore, bidirectional RNNs have been used to capture forward and backward interactions. Two kinds of RNN implementation are used for this case: LSTM and GRU.

The backbone architecture of SPIDER3 [34] and SPIDER-Single [74] employed two stacked LSTM layers and two fully connected layers. Hu et al. [72] utilized ensemble learning of five base learners, and each of the base learners consisted of stacked bidirectional LSTM layers and fully connected layers. Hattori et al. [113] and Wang et al. [114] also utilized stacked LSTM layers in their PSSP model architectures. Yang et al. [115] combined bidirectional GRU and batch normalization in their PSSP model. Lyu et al. [116] built their PSSP model using two stacked bidirectional GRU layers flanked by two multi-layer-perceptron (MLP) layers. De Oliveira et al. [117] used five bidirectional GRU layers in their global classifier and five random forests in their local classifier. Ensemble learning is utilized to combine local and global classifiers.

Many PSSP models integrated both CNN and RNN in their architectures. The combination of CNN and RNN in the architecture of a PSSP model enables the PSSP model to capture both local contexts and long-range interactions.

**Feature extraction using autoencoder.** Besides using CNN and RNN for feature extraction, an autoencoder has been utilized to extract features on protein sequences. Several PSSP models have integrated autoencoder in their architectures [118–121,81].

**Prediction algorithms.** A typical PSSP model consists of feature extraction modules and prediction modules. For the latter part, many researchers tend to use a fully connected layer and a softmax activation function. However, several PSSP studies have attempted to use different algorithms to make secondary structure predictions. Dionysiou et al. [122], Sutanto et al. [123], and Görmez and Aysin [124] used SVM instead of a fully connected layer and a softmax activation function. However, several PSSP studies have attempted to use different algorithms to make secondary structure predictions. Dionysiou et al. [122], Sutanto et al. [123], and Görmez and Aysin [124] used SVM instead of a fully connected layer. Random forest was utilized in several PSSP studies [125–127], while a Bayesian classifier was used in other studies [119,103,121].

**Multitask learning.** Several previous PSSP researchers designed their models to be able to perform more than one prediction task. For example, a particular model predicts not only the protein secondary structure but also other structural features, such as solvent accessibility, dihedral angles, protein disorder, and protein structural classes. Solvent accessibility measures the exposure of amino acid residues to solvent, and it is essential for understanding and predicting protein structure, function, and interactions [128]. Torsion angles ($\phi$ and $\psi$) provide a continuous representation of the local conformations [129] rather than the discrete secondary structures. Moreover, these continuous representations are potential for predicted local structure in fragment-free tertiary-structure prediction. Hence, it is advantageous to use protein structural properties in addition to secondary structures for protein tertiary structure prediction. Motivated by that, previous PSSP researchers
performed multitask learning to predict not only the secondary structure but also other protein structural features.

Li and Yu [112] developed a PSSP model that predicts both secondary structure and solvent accessibility. SPIDER3 [34] produced several prediction outputs: secondary structure, solvent accessibility, contact number, half-sphere exposure, and dihedral angles. NetSurfP-2.0 [37] was developed using a single model to predict three-state PSSP, eight-state PSSP, relative solvent accessibility, dihedral angles, and protein disorder. MASSP [82] was designed to be able to predict both residue-level structural attributes (secondary structure, location, orientation, and topology) and protein-level structural classes (bitopic, α-helical, β-barrel, and soluble).

**Ensemble learning.** Ensemble learning is an approach in machine learning that combines several base classifiers/predictors to achieve better prediction performance. Previous PSSP researchers have used ensemble learning to boost the accuracy gained by their models. Basri et al. [110] developed an ensemble of multiple artificial neural networks, where each network can have distinct parameters and architecture. eCRRN [38] employed ten independently trained CRNN models. PORTER5 [131] employed an ensemble of seven BRNN networks. SPOT-1D [36] leveraged the ensemble of several deep learning networks, including LSTM- BRNN and ResNet. Guo et al. [132] leveraged the ensemble learning of conditionally parameterized convolutional networks (CondGCNN) and bidirectional LSTM. Cheng et al. [126] used bagging and AdaBoost with several feed-forward neural networks as the base classifiers.

**Iterative learning.** SPIDER3 [34] initiated iterative learning to predict protein secondary structure. The model training took four iterations, for which the earlier iteration’s output and the original input features were used as the input for the next iteration. The original input features of SPIDER3 include physicochemical properties of amino acids, PSSM, and HMM profiles.

**Attention mechanism.** PSSP is similar to machine translation in NLP whereby both cases use sequences as input and output. Previous PSSP models had adopted approaches and methods used in the machine translation field. One of the methods adopted by PSSP models is the attention mechanism. The first PSSP model utilizing the attention mechanism was proposed by Drori et al. [133], where the attention mechanism proposed in [134] was used. SAINT [110] proposed a new PSSP model architecture by augmenting self-attention (transformer) [135] in Deep3I (deep inception inside inception) architecture. OPUS-TASS [56] utilized modified self-attention (transformer) by dismissing the decoder part. TMPSS [69] utilized a shared attention mechanism initially proposed in [136] for multi-way multi-language machine translation. The attention mechanism was also used in the PSSP model proposed by Guo et al. [132].

**Miscellaneous.** Wang et al. [137] integrated a deep convolutional neural network with conditional random field (CRF) to model complex sequence-structure relationships and interdependency between neighboring amino acids. Instead of using real-valued neural networks, complex-valued neural networks were utilized in [140] and applied to a small training set (compact model) of the CB513 dataset. Zhao et al. [79] applied a generative adversarial networks (GAN) to perform feature extraction on the original PSSM data and fed the result into CNN. Yavuz et al. [141] proposed using a cloning selection algorithm in the PSSP model architecture to improve the data before the classification process.

**4.3. Performance**

The performance of previous notable PSSP models using hybrid features is shown in Table 5. The performance of single-sequence-based PSSP models is shown in Table 6.

### Table 4

Deep learning methods involved in previous notable PSSP models.  

| Author         | Model         | Deep learning architecture | Attention |
|----------------|---------------|----------------------------|-----------|
| Lin et al. [107]| MUST-CNN      |                            |           |
| Wang et al. [137]| DeepCNF      |                            |           |
| Yang et al. [138]| SPIDER2      |                            |           |
| Heffernan et al. [34]| SPIDER3       |                            |           |
| Wang et al. [120]| SSREDNs      |                            |           |
| Zhang et al. [38]| eCRRNN       |                            |           |
| Zhou et al. [108]| CNNH_PSS     |                            |           |
| Guo et al. [104]| 2C-BRNNs     |                            |           |
| Fang et al. [35]| MUFOID-SS    |                            |           |
| Heffernan et al. [74]| SPIDER3-Single|                            |           |
| Klausen et al. [37]| NetSurfP-2.0|                            |           |
| Hu et al. [139]| DeepNRCN     |                            |           |
| Hanson et al. [36]| SPOT-1D      |                            |           |
| Guo et al. [104]| DeepACLTSM   |                            |           |
| Xu et al. [56]| OPUS-TASS    |                            |           |
| Uddin et al. [110]| SAINT       |                            |           |
| Shapovalov et al. [13]| SecNet      |                            |           |
| Li et al. [82]| MASSP        |                            |           |
| Moffat and Jones [70]| S4PRD       |                            |           |
| Liu et al. [69]| TMPS         |                            |           |
| Kowalowska et al. [99]| ProteinInet|                            |           |
| Singh et al. [100]| SPOT-1D-Single|                            |           |
| Zhao and Liu [68]| OLSTM        |                            |           |
| Lyu et al. [116]| MLPNN        |                            |           |
| Guo et al. [67]| DNS2         |                            |           |
| Görmez et al. [66]| IGPRD       |                            |           |

GC: graph convolutional networks; CN: convolutional neural networks; RN: residual neural networks; IN: inception networks; US: U-shape networks; FN: fractal networks; AE: auto-encoder; LS: long short-term memory; GR: gated recurrent unit; FF: feed-forward neural networks; FC: fully connected layers; CR: conditional random field; Sf: self attention (transformer); Glb: global attention.
Both tables show that previous PSSP models are trained on non-uniform datasets. As the training data determine the capability of PSSP models to extract patterns from the input sequences and influence the performance of the classification task, a fair performance comparison cannot be made in this setting. Moreover, using sequence profiles as input of the PSSP models may lead to an evaluation bias primarily when redundancy of profiles between the training dataset and test dataset occurs [55]. Hence, a proper comparison among PSSP models could be made if two conditions are met: (1) PSSP models are trained with identical training datasets, (2) profile redundancy between training and test dataset is diminished.

5. New approaches for feature generation

The favorable outcome of language models (LMs) in NLP has driven researchers to implement LMs in protein structure prediction. Researchers have developed protein LMs to learn the biolog-

| Year | PSSP model | Training set | Test set | Q3 | Q8 | SOV |
|------|------------|--------------|----------|----|----|-----|
| 2016 | MUST-CNN   | 4Prot, CullPDB | CB513    | -  | 68.40 | - |
|      | DeepCNF    | CullPDB 6125 | CullPDB test set | 85.40 | 75.20 | 86.70 |
|      |            |              | CB513    | 82.30 | 68.30 | 84.80 |
|      |            |              | CASP10   | 84.40 | 71.80 | 85.70 |
|      |            |              | CASP11   | 84.70 | 73.30 | 86.50 |
|      |            |              | CAMEO    | 84.50 | 72.10 | 85.50 |
| 2017 | SPIDER2    | 5,789 sequences | TS1199   | 82.00 | -    | - |
|      | SPIDER3    | TR4590       | TS115    | 83.90 | -    | - |
|      | SSREDNs    | CullPDB      | CB513    | 82.90 | 68.20 | - |
|      |            |              | CullPDB test set | 84.20 | 73.10 | 78.20 |
| 2018 | CNNH_PSS   | CB6133       | CB6133   | -   | 74.00 | - |
|      | eCRRNN     | TR12148      | CB513    | 87.30 | 74.00 | 84.20* |
|      |            |              | CASP10   | 87.80 | 73.30 | 85.10* |
|      |            |              | CASP11   | 85.90 | 73.90 | 85.00* |
|      |            |              | CASP12   | 83.70 | 70.70 | 81.70* |
|      | 2C-BRNNs   | CB6133       | CASP10   | -   | 74.50 | - |
|      |            |              | CASP11   | -   | 72.60 | - |
|      |            |              | CB6133   | -   | 75.70 | - |
|      |            |              | CB513    | -   | 70.20 | - |
| 2019 | NetSurfP-2.0 | 10,337 sequences | CB513   | 85.40 | 72.30 | - |
|      | DeepNRCN   | CullPDB      | TS115    | 85.70 | 75.00 | - |
|      | SPOT-1D    | 10,029 sequences | CB513   | 82.40 | 71.10 | - |
|      |            |              | CASP10   | 86.49 | 76.47 | - |
|      | DeepACLSTM | CB6133       | CASP11   | 85.20 | 74.51 | - |
|      |            |              | CASP12   | 83.36 | 72.10 | - |
| 2020 | OPUS-TASS  | SPOT-1D dataset | TEST2018 | 86.84 | 76.59 | - |
|      | SAINT      | SPOT-1D dataset | TEST2018 | 87.79 | 78.01 | - |
|      |            |              | TEST2016 | -   | 77.73 | - |
|      |            |              | TEST2018 | -   | 76.09 | - |
|      |            |              | CASP12   | -   | 74.17 | - |
|      |            |              | CASP13   | -   | 74.78 | - |
|      |            |              | CASP-FM  | -   | 72.25 | - |
|      | SecNet     | 8,563 sequences | Test2018 | 84.00 | 73.00 | - |
| 2021 | TMPSS      | 811 TM sequences | TEST50  | 84.00 | -   | - |
|      | OCLSTM     | 15,696 sequences | CASP10  | 84.68 | -   | - |
|      |            |              | CASP11   | 82.36 | -   | - |
|      |            |              | CASP12   | 82.91 | -   | - |
|      |            |              | 2SPDB    | 84.21 | -   | - |
|      |            |              | CB513    | 85.08 | -   | - |
|      | MLPRNN     | CB6133-filtered | CB513   | 83.32 | 70.59 | - |
|      | DNSS2      | 4,872 sequences | DNS2 Test_Set | 86.64 | - | 75.57 |
|      |            |              | CB513    | 82.56 | -   | 73.12 |
|      |            |              | CASP11   | 82.84 | -   | 74.34 |
|      |            |              | CASP12   | 80.95 | -   | 71.76 |

*SOV’99
The objective of using masked language modeling is to reveal the relationship/dependencies between the masked and non-masked regions. The embedding features derived from these LMs carry contextual features of the amino acid residues [143]. These embedding features may replace sequence profiles for the prediction input.

Heinzinger et al. [142] adopted the bidirectional language model ELMo (Embeddings from Language Models) [144] to build a protein representation in the form of continuous vectors (embeddings) called as SecVec (sequence-to-vector). They trained ELMo on UniRef50 and assessed the predictive capability of the embeddings by applications to per-residue (word-level) and per-protein (sentence-level) tasks. For the per-residue task, they evaluated PSSP performance given one-hot amino acid encoding. SecVec embeddings, evolutionary profiles, or their combinations as inputs. SecVec embeddings produced the best performance for prediction without evolutionary information, although it did not improve over the best existing method using evolutionary information. Regarding the computation time, generating SecVec embedding is about 60-fold faster than generating HMM profiles using HHblits.

ProtTrans [49] trained six LMs in NLP, including T5 [145], Electra [146], BERT [147], ALBERT [148], Transformer-XL [149], and XLNet [150] on protein sequences. Five models, including ProtTXL, ProtBert, ProtXLNet, ProtAlbert, and ProtElectra, are trained using Uniref100 [151]. Besides Uniref100, ProtBert and ProtXL are also trained using BFD [152]. ProtT5 is trained on UniRef50 [151] and also trained on BFD. ProtTrans initially tokenizes the protein sequences. For the per-residue task, the token will be each amino acid. It then adds positional encoding for each respective residue. Afterward, the resulting vectors are passed through one of the ProtTrans models to create embeddings for each input token. The last hidden state of the Transformer’s attention stack of the models is used as input for the task-specific predictors (i.e., three-state PSSP model). ProtTrans experiments showed that the Q3 accuracy on CASP12 achieved by using ProtTrans embeddings is higher than using word2vec embeddings. Using the same test set (CASP12), the ProtTXL version is less accurate compared to an existing ELMo/LSTM-based solution (SeqVec) [142], while all other Transformer-models outperformed the SeqVec model. Moreover, ProtTrans resulted in lower Q3 accuracy than NetSurfP-2.0 [37], a biennial protein structure competition held by the international community. AlphaFold can predict not only the 3D structure of single-chain proteins but also protein complexes [158].

### 6. AlphaFold and its impact

AlphaFold [3] became a revolution in the field of structural bioinformatics after it was able to predict 3D coordinates of protein structures with high accuracy. AlphaFold succeeded in predicting protein structures in the critical assessment of protein structure prediction round 14 (CASP14) [87], a biennial protein structure competition held by the international community. AlphaFold can predict not only the 3D structure of single-chain proteins but also protein complexes [158].

AlphaFold generates MSA from the genetic database search, residue pairs, and structural templates from the structural database search. Residue pairs and structural templates make up the pair representation. AlphaFold’s architecture consists of two main networks: Evoformer and the structure module. Evoformer consists of 48 blocks containing attention-based and non-attention-based components. Each block of the Evoformer updates the MSA and the pair representation and facilitates communication between them. Evoformer performs three updates to update the MSA representation: row-wise gated self-attention, column-wise gated self-attention, and transition. The first step incorporates information from the pair representation as input to synchronize the MSA with the updated pair representation from the previous block. For each step, the output is added to the input and the result of this addition is fed into the next step. The result of the MSA update is then injected into the pair representation.

Evoformer utilizes a triangle portrayal to depict the pair representation into a 3D structure. This triangle is a graph of three different nodes. The nodes represent amino acid residues, whereas the edges represent the connectivity between amino acid residues. Evoformer performs several operations to update the pair representation, including: injection of the updated MSA into the pair representation by executing the outer product mean operation, two non-attentional triangle multiplicative updates (using ‘outgoing’ edges and ‘incoming’ edges), two attentional triangle updates (self-attention around starting node and self-attention around terminating node), and transition.

Each block of the Evoformer is followed by a structure module. The objective of the structure module is to generate the 3D representation of large protein sequences in the databases [142,49,50]. The embedding features derived from these LMs carry contextual features of the amino acid residues [143]. These embedding features may replace sequence profiles for the prediction input.

### Table 6

| Year | PSSP model          | Training set                  | Test set                  | Accuracy [%] |
|------|---------------------|-------------------------------|---------------------------|--------------|
| 2018 | SPIDER3-Single      | TR9993                        | TS1199                    | 72.56        |
|      |                     |                               | TS1250                    | 72.52        |
|      |                     |                               | TS-Hard                   | 73.24        |
| 2020 | ProteinUnet         | TR9961                        | TS1197                    | 72.66        |
| 2021 | S4PRED              | 10,143 labelled sequences     | CBS13                     | 75.30        |
|      | SPOT-1D-Single       | ProteinNet dataset            | SPOT-2016                 | 74.29        |
|      |                     |                               | SPOT-2018-HQ              | 73.65        |
|      |                     |                               | SPOT-2018                 | 73.71        |
|      |                     |                               | SPOT-2018-HQ              | 72.12        |

The ProtTrans and ESM-1b embedding features have been utilized to train the following PSSP models, NetSurfP-3.0 [154], DML_SS [143], and SPOT-1D-LM [155]. Moreover, these embedding features have been employed to train the contact map prediction model, SPOT-Contact-LM [156], and the inter-residue distance predictor [157].
tation of the input protein by using the MSA and the pair represen-
tation that the Evoformer has manipulated. Moreover, the structure
module also takes backbone frames as input. The backbone frames
represent each residue as one free-floating unit of the protein
polypeptide backbone. The geometry of N-C module also takes
backbone frames as input. The backbone frames
representation of the input protein by using the MSA and the pair representa-
tion predictors, and side-chain angles predictor. The IPA is a
geometry-aware attention operation that updates the MSA’s single
representation that the Evoformer has manipulated. Moreover, the structure
module consists of three parts: Invariant Point Attention (IPA), relative rotation and translation
predictors, and side-chain angles predictor. The IPA is a
geometry-aware attention operation that updates the MSA’s single
representation, which is then used to update the backbone frames
via rotation and translation operations. Together with the updated
backbone frames, the single representation of the MSA predicts the
side-chain angles and the position of atoms composing each residue.

The success of AlphaFold in predicting the 3D structure of pro-
teins is induced by the combination of bioinformatics and physical
approaches it implemented. Evoformer successfully produces rep-
resentations that reveal spatial and evolutionary relationships of
the residues. Moreover, these spatial and evolutionary relations-
ships are used by the structure module to capture the orientation of
the protein backbone, thus the 3D structure of the protein can
be revealed. With the invention of AlphaFold, the prediction of the
3D structure of proteins from their amino acid sequences is
already practicable. It directly predicts the tertiary structure of proteins by using only their primary structures, bypassing the intermediate tasks in protein structure prediction, such as PSSP, folds prediction, and prediction of protein structural features.

This advancement in protein structure prediction pioneered by
AlphaFold undoubtedly raises the question of whether PSSP and the prediction of protein structural features are still necessary since predicting protein tertiary structure from its amino acid sequence becomes practical. AlphaFold is powerful in predicting protein ter-
tary structures; however, since AlphaFold relies on MSA, it is partic-
ularly beneficial for homologous proteins. AlphaFold performance is
imperfect in predicting the structure of proteins with no prior evolu-
tionary information (proteins without known homologs) [159,160],
whereas proteins without known homologs are currently estimated
at 20% of all metagenomic protein sequences [161] and about 11% of
eukaryotic and viral proteins [162]. This case of low-quality MSA will
lead to less accuracy in structure prediction. Moreover, leveraging MSA requires high computing resources, especially to generate MSA from massive collections of protein sequences from multiple
protein databases. PSSP with evolutionary information as input
may fall for the same problem. Here, PSSP and prediction of protein structural features, especially ones that do not rely on evolutionary
information might still be needed for proteins without known homologs.

7. PSSP in post-AlphaFold publication

Although AlphaFold is a breakthrough in protein structure pre-
diction due to its highly accurate prediction, we found that PSSP
studies continue after the release of AlphaFold. This section dis-
cusses the PSSP models proposed by researchers after AlphaFold
was published (2021). PSSP studies in the post-AlphaFold era focus
on one or more objectives including: (1) taking MSA instead of
PSSM as direct input to the PSSP model, (2) improving the quality of
sequence profiles, (3) exploiting protein language models (LMs) as alternative input features, and (4) other objectives.

7.1. Taking MSA as direct input to the PSSP model

AlphaFold has successfully used MSA instead of sequence pro-
files to produce accurate structure prediction. Another study by
Ju et al. [163] suggested the use of MSA as a direct input for the
PSSP model concerning that sequence profiles derived from MSA
(PSSM and HMM profiles) may not be able to represent residue
mutations and correlations. The model, called Seq-SetNet, per-
formed two stages, encoding and aggregation, to deduce the struc-
tural properties of each amino acid in the input sequence. The
encoding module processes each homologue sequence from the
MSA independently and produces a contextual vector in which each position in the vector covers the information of its surround-
ings. The encoding module is constructed of 1D convolutional
residual networks with 8 residual blocks. The aggregation module
aggregates the outputs of the encoder module, namely the context-
tual features extracted from all the homologous proteins. The
element-wise maximum function is applied to the contextual fea-
tures of the homologue proteins to perform the aggregation. The
experiment result showed that Seq-SetNet outperformed the base-
line PSSP model, which used handcrafted features, such as PSSM and
HMM profiles.

7.2. Improving the quality of sequence profiles

Considering the low-quality sequence profiles that proteins
without known homologs may have, a recent PSSP study by Wang
et al. [164] proposed PSSM-Distil, which conducts knowledge dis-
tillation to produce enhanced PSSM leveraging student–teacher
networks. PSSM-Distil basically performs several steps. First, it
trains the teacher network with high-quality PSSM. Secondly, it
down-samples the high-quality PSSM to get the low-quality one
by using prior statistics and then, the low-quality PSSM and pre-
trained BERT are used to train a network producing an enhanced
PSSM. Thirdly, it performs secondary structure classification using
enhanced PSSM on a student network. It also performs the sec-
ondary structure classification on the teacher network. A loss func-
tion based on the loss of the student network and the loss of the
teacher network is calculated to measure the PSSP performance. The experiment result showed that using PSSM-Distil instead of
the standard PSSM increased the accuracy of PSSP in low-quality
PSSM cases in BC40 and CB513 datasets. Subsequently, the same
researchers also proposed a dynamic scoring matrix (DSM)-Distil
to replace PSSM and other widely used features [165]. This feature
leveraged the pre-trained BERT to construct a dynamic scoring
matrix (DSM) and performed knowledge distillation on the DSM.

7.3. Exploiting protein language models (LMs) as input features

Following the success of language models in NLP, protein lan-
guage models (LMs) have appeared to be an alternative to replace
the evolutionary information as an input feature for PSSP models.
DML_SS [143] utilized embedding features derived from the Prot-
Trans language model (ProtT5-XL-U50). DML_SS used a deep cen-
troid model based on deep metric learning in its model
architecture. This deep centroid model is used to learn deep
embedding to keep the samples with the same labels close to each
other in the embedding space while samples with different labels
remain distant from each other. DML_SS employed embedding
networks consisting of inception networks, convolutional
networks, and fully connected layers. DML_SS proved that by using
embedding features, its prediction accuracy in the CB513 test data-
set is higher than if it used hybrid features (sequence profile fea-
tures and physicochemical properties).

LMs have also been utilized to improve the performance of
NetSurfP-2.0. NetSurfP-3.0 [154] employed the architecture of
NetSurfP-2.0 [37] and utilized latent sequence representation
derived from the ESM-1b as the input feature. NetsurfP-3.0 showed
that it surpasses the performance of NetSurfP-2.0 in CB513 and
TS115 test datasets, although it does not perform better on the
They also used the Adjusted Geometric Mean (AGM) to accelerate the secondary structure prediction of large datasets. Moreover, ProteinUnet2 can perform other single-sequence-based PSSP models in terms of Q8 accuracy in the same test dataset.

Table 7 shows the performance of LM-based PSSP models. Similar to Table 5 and Table 6 cases, a fair comparison of LM-based PSSP models can be conducted when the models are trained using identical training datasets.

### 7.4. PSSP studies aiming at other objectives

The length of input sequences fed into DNN-based PSSP models may vary due to the different amino acid compositions of the input proteins. Thus, padding is a common practice conducted to have equal size inputs passed into the networks. Padding is done by adding zero values into the sequences shorter than a predetermined size such that they are adjusted to that size. However, this standard practice raises an issue: the networks will change the feature vectors corresponding to the padded positions to non-zero vectors during forward propagation, affecting the prediction result of the non-padded positions. To address this issue, Yang et al. [166] proposed a modified 1D batch normalization by exploiting a mask matrix, so that padding positions in the feature vectors do not participate in normalization operations. Hence, the feature vectors in these padding positions remain all-zeros. Furthermore, they [166] also proposed a PSSP model with architecture comprised of CNNs and fully connected layers along with the implementation of masked batch normalization.

Stapor et al. [167] proposed lightweight ProteinUnet2, a PSSP model based on U-net convolutional networks. ProteinUnet2 takes sequence profiles (PSSM and HMM profiles) and SPOT-Contact features as inputs. ProteinUnet2 is developed based on the architecture of ProteinUnet [99], extended with multiple inputs using multiple contractive paths. The experiment shows that ProteinUnet2 achieves a comparable prediction accuracy to SPOT-1D and SAINT. However, ProteinUnet requires less training and prediction time than SPOT-1D and SAINT. Hence, it could be useful for a system with low computing resources. Moreover, ProteinUnet2 can accelerate the secondary structure prediction of large datasets. They [167] also used the Adjusted Geometric Mean (AGM) [168] metric to assess the performance of PSSP models, especially due to the imbalanced dataset available in the field.

### 8. Summary and outlook

PSSP has been a mature research topic in structural bioinformatics. The three-state PSSP has achieved accuracy close to its theoretical limit, 88–89% [170]. However, considering the current protein structure databases, a new study reported that the theoretical limit of PSSP can be extended to 94% [171]. This means that there is still an accuracy gap to fill.

The state of the art of PSSP accuracy is not distributed evenly for all classes (helix/sheet/coil). The prediction accuracy of sheet and coil is not as high as that of helix. The ability to precisely capture long-range interactions, especially in sheet (β-strand), may need to be improved to boost the prediction accuracy, even though previous models had used RNN to handle this problem. Recently, the embedding features derived from protein language models can reveal the contextual information of the amino acid residues [143]. Using embedding features derived from protein language models in NetSurf-P3.0, DML SS, and SPOT-1D-LM has successfully improved the prediction accuracy of the benchmark test datasets. The impact of embedding features on the ability to accurately predict non-helix structures can be further investigated.

On the other side, there is room for improvement in both eight-state PSSP and single-sequence-based PSSP, as prediction accuracy is still limited. The challenge in the eight-state PSSP was the imbalanced samples of each class; some classes lacked samples, such as B, S, G, and I. The eight-state PSSP model performance can be affected by this imbalanced dataset issue. Thus, the methods and techniques to train DNN on this imbalanced dataset case are potential for future work. A new classification of four and five-state PSSP proposed by Shapovalov et al. [13] may become a potential alternative to PSSP.

DNN has been employed in single-sequence-based PSSP, but only a few models have been published. Various deep neural net-
work architectures are feasible to be implemented. Adopting previous DNN-based PSSP models into single-sequence cases is also worthwhile. Considering the recent success of LM-based PSSP models in enhancing the PSSP accuracy, combining LM-derived features and amino acid sequences are potential input features for future PSSP models. Furthermore, the issue remaining in the post-AlphFold era is the low-quality evolutionary information of the proteins without known homologs. Hence, the structure prediction models that exclude evolutionary information will be useful for such proteins. Single-sequence-based PSSP models can be further developed.

CRediT authorship contribution statement

Dewi Pramudi Ismi: Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing. Reza Pulungan: Conceptualization, Methodology, Writing – review & editing. Afiahayati: Conceptualization, Analysis, Methodology, Writing – review.

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.

Acknowledgments

This work is supported in part by the Directorate of Research, Technology, and Community Services, Directorate General of Higher Education, Research, and Technology, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia, through Doctoral Dissertation Research Grant No. 1906/UN1/DITLIT/Dit-Lit/PT.01.03/2022.

References

[1] Breda A, Valadares N.F., de Souza D.N., Garratt R.C. Protein structure, modelling and applications. In: A. Gruber, A.M. Durham, C. Huynh, H.A. del Portillo (Eds.). Bioinformatics in tropical disease research: a practical and case-study approach. National Center for Biotechnology Information (US), Bethesda (MD): 2008. Ch. A06, pp. 137–170.
[2] Branden CI, Tooze J. Introduction to Protein Structure. Garland Science 2012. https://doi.org/10.1201/9781134680896
[3] Junger J, Evans R, Pruzan K, Figurnov M, Ronneberger O, Tunnyashvunskool K, Bates R, Židek A, Potapenko A, Bridgland A, Meyer C, Kohl SA, Ballard AJ, Cowie A, Romera-Paredes B, Nikolov S, Jain R, Adler J, Back T, Petersen S, Reiman D, Clancy E, Zielinski M, Kohli P, Hassabis D. Highly accurate protein structure prediction with AlphaFold. Nature 2021;596(7873):583–9. https://doi.org/10.1038/s41586-021-02931-2
[4] Gibson KD, Scheraga H. Minimization of polypeptide energy. I. Preliminary structures of bovine pancreatic ribonuclease s-peptide. Proc Natl Acad Sci 1967;58(2):420–7. https://doi.org/10.1073/pnas.58.2.420
[5] Levitt M. Protein folding by restrained energy minimization and molecular dynamics. J Mol Biol 1983;170(3):723–64. https://doi.org/10.1016/0022-2836(83)90279-9
[6] Tai C-H, Bai H, Taylor TJ, Lee B. Assessment of template-free modeling in secondary structure prediction. J Mol Biol 1978;120(1):97–120. https://doi.org/10.1016/0022-2836(78)90037-6
[7] Zhou Y, Duan Y, Yang Y, Faraggi E, Lei H. Trends in template/fragment-free prediction of secondary structure, backbone angles, solvent accessibility and contact numbers on the prediction of protein secondary structures. BMC Bioinformatics 2008;9(1):357. https://doi.org/10.1186/1471-2105-9-357
[8] Rost B. [31] PHD: Predicting one-dimensional protein structure by profile-based neural networks. Methods in Enzymology 1996;266:525–39. https://doi.org/10.1016/S0076-6879(96)66033-9
[9] Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 1999;292(2):195–202. https://doi.org/10.1006/jmbi.1999.3091
[10] Pollastri G, Przybylski D, Rost B, Baldi P. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins: Struct, Funct, Bioinf 2002;47(2):228–35. https://doi.org/10.1002/prot.10082
[11] Xiao S-K, Park C-B, Lee S-W. Protein secondary structure prediction using sequence profile and conserved domain profile. In: International Conference on Intelligent Computing. Springer: 2005. p. 1–10.
[12] Birzole F, Kramer S. A new representation for protein secondary structure prediction based on frequent patterns. Bioinformatics 2006;22(21):2628–34. https://doi.org/10.1093/bioinformatics/btl352
[13] Meiler J, Müller M, Zeidler A, Schäfler F. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. Mol Model Annual 2001;7(9):360–9. https://doi.org/10.1002/mdb.200100136
[14] Wood MJ, Hirst JD. Protein secondary structure prediction with dihedral angles. Proteins: Struct, Funct, Bioinf 2005;59(3):476–81. https://doi.org/10.1002/prot.20435
[15] Midic U, Dunker AK, Obadovc Z. Exploring alternative knowledge representations for protein secondary-structure prediction. Int J Data Min Bioinform 2007;1(3):285–97. https://doi.org/10.1504/IJDMB.2007.013089
[16] Momen-Roknabadi A, Sadeghi M, Pezeshki H, Marashi S-A. Impact of residue contact numbers and solvent accessibility on the prediction of protein secondary structures. BMC Bioinformatics 2009;10(1):357. https://doi.org/10.1186/1471-2105-10-357
[17] Heffernan R, Paiwali K, Iyonj J, Dehang A, Sharma A, Wang J, Sattar A, Yang Y, Jey B. Improving prediction of secondary structure, local backbone angles and solvent accessible surface area of proteins by iterative deep learning. Sci Rep 2013;5(1):11476. https://doi.org/10.1038/srep11476
[18] Wang S, Liu W, Liu S, Xu J. RaptorX-Property: a web server for protein structure property prediction. Nucleic Acids Res 2016;44(1W):W430–5. https://doi.org/10.1093/nar/gkw1306
[19] Heffernan R, Yang Y, Paiwali K, Zhou Y. Capturing non-local interactions by local short-term memory recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. Bioinformatics 2017;33(18):2842–9. https://doi.org/10.1093/bioinformatics/btx318
[20] Fang C, Shang Y, Xu D. MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction. Proteins: Struct, Funct, Bioinf 2018;86(5):592–8. https://doi.org/10.1002/prot.25487
[21] Hansson J, Paiwali K, Lofin T, Yang Y, Zhou Y. Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent deep networks. Mol Syst Biol 2019;15(8):e0232528.
