Computational Biology and Chemistry 91 (2021) 107436

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/cbac

diSBPred: A machine learning based approach for disulfide bond prediction

Avdesh Mishra [a,1], Md Wasi Ul Kabir [b,1], Md Tamjidul Hoque [b,*]

[a] Department of Electrical Engineering and Computer Science, Texas A&M University-Kingsville, Kingsville, TX, USA
[b] Department of Computer Science, University of New Orleans, New Orleans, LA, USA

ARTICLE INFO

Keywords:
Machine learning
Disulfide bond prediction
Protein structure
Protein sequence

ABSTRACT

The protein disulfide bond is a covalent bond that forms during post-translational modification by the oxidation of a pair of cysteines. In protein, the disulfide bond is the most frequent covalent link between amino acids after the peptide bond. It plays a significant role in three-dimensional (3D) *ab initio* protein structure prediction (*aiPSP*), stabilizing protein conformation, post-translational modification, and protein folding. In *aiPSP*, the location of disulfide bonds can strongly reduce the conformational space searching by imposing geometrical constraints. Existing experimental techniques for the determination of disulfide bonds are time-consuming and expensive. Thus, developing sequence-based computational methods for disulfide bond prediction becomes indispensable. This study proposed a stacking-based machine learning approach for disulfide bond prediction (*diSBPred*). Various useful sequence and structure-based features are extracted for effective training, including conservation profile, residue solvent accessibility, torsion angle flexibility, disorder probability, a sequential distance between cysteines, and more. The prediction of disulfide bonds is carried out in two stages: first, individual cysteines are predicted as either bonding or non-bonding; second, the cysteine-pairs are predicted as either bonding or non-bonding by including the results from cysteine bonding prediction as a feature.

The examination of the relevance of the features employed in this study and the features utilized in the existing nearest neighbor algorithm (NNA) method shows that the features used in this study improve about 7.39 % in jackknife validation balanced accuracy. Moreover, for individual cysteine bonding prediction and cysteine-pair bonding prediction, diSBPred provides a 10-fold cross-validation balanced accuracy of 82.29 % and 94.20 %, respectively. Altogether, our predictor achieves an improvement of 43.25 % based on balanced accuracy compared to the existing NNA based approach. Thus, diSBPred can be utilized to annotate the cysteine bonding residues of protein sequences whose structures are unknown as well as improve the accuracy of the aiPSP method, which can further aid in experimental studies of the disulfide bond and structure determination.

1. Introduction

Disulfide bonds in proteins, also known as disulfide bridge or SS-bond, are formed between the thiol (-SH) groups of cysteine residues by oxidative folding. After the peptide bond, the disulfide bond is the most common covalent connection between cysteine residues in proteins (Mossuto, 2013). Disulfide bonds play a significant role in stabilizing proteins thermodynamically by decreasing the entropy of the unfolded state, increasing mechanical stability, and confining conformational changes by imposing geometrical constraints on the protein backbone (Fass, 2012; Chuang et al., 2003). Accurate identification of disulfide bonds can significantly reduce the large and convoluted conformational search space of possible protein conformation and subsequently facilitate the *ab initio* protein structure prediction (*aiPSP*) (Márquez-Chamorro and Aguilar-Ruiz, 2015; Huang et al., 1999) for an improved prediction of 3D protein structure. For example, Yang et al. developed a machine learning approach for disulfide bridge prediction and integrated the outcome with QUARK simulations for better accuracy (Yang et al., 2015). The usefulness of the disulfide bonds has been recognized in various physiological function such as cell death (Nakamura and Lipton, 2009), hemostasis (Hogg, 2009), G-protein-receptors (Wess et al., 2008) and growth factors (Guo et al., 2010), and pathological processes such as tumor immunity (Dranoff, 2009) and neurodegenerative misfolding diseases (Mossuto, 2013; Nakamura and Lipton, 2009).

However, the existing experimental techniques such as X-ray

* Corresponding author.
E-mail address: thoque@uno.edu (M.T. Hoque).
[1] Equal Authorship.

https://doi.org/10.1016/j.compbiolchem.2021.107436
Received 24 June 2020; Received in revised form 28 December 2020; Accepted 10 January 2021
Available online 27 January 2021

crystallography (Sutton et al., 2013), mass spectrometry (Sun and Smith, 1988), NMR (Mobli and King, 2010), and radiation experiment (Chaudhuri et al., 2001) for the determination of disulfide bonds require time-consuming and expensive apparatus. Hence, it is important to develop computational methods for the fast and effective identification of cysteine disulfide bonds at the proteome level. The computational method refers to a wide variety of approaches that capture various information, such as structural, sequential, and other proteins' physicochemical properties. Several attempts have been made in identifying disulfide bonds, and several computational methods have been proposed in the literature for analyzing them (Fariselli and Casadio, 2001; Niu et al., 2013; Song et al., 2007; Vincent et al., 2008; Tsai et al., 2005; Cheng et al., 2005; Lin et al., 2012).

Disulfide bonding state prediction of individual cysteines in proteins is necessary to predict disulfide connectivity. Numerous studies on computational methods can predict disulfide bonding state in the literature (Muskal et al., 1990; Fiser et al., 1992; Fariselli et al., 1999; Fiser and Simon, 2000). Moreover, one of the first computational approaches for disulfide bond prediction was presented by Fariselli and Casadio (2001), where they reduce disulfide connectivity to the graph matching problem in which vertices represent oxidized cysteines and edges between the associated pair of cysteines are labeled with the contact potential. Consequently, the Monte Carlo simulated annealing method was used to find the optimal values of contact potentials. Finally, the disulfide bonds were located by finding the maximum weighted perfect matching. Following this work, several neural network-based methods (Cheng et al., 2005; Vullo and Frasconi, 2004; Ferrè and Clote, 2005) were proposed.

Conversely, Ferre and Clote developed a web server called DiANNA 1.1 (Ferrè and Clote, 2006), which uses a support vector machine (SVM) with a spectrum kernel for the classification of cysteines into reduced (free), half-cysteine (involved in disulfide bond), or metallic ligand-bound state. Likewise, Song et al. developed a disulfide connectivity predictor using a support vector regression trained on multiple sequence feature vectors and predicted secondary structures (Song et al., 2007). Rubinstein and Fiser (2008) developed yet another method that analyzes correlated mutation patterns in multiple sequence alignments to predict disulfide bonds. For the proteins with two, three, and four disulfide bonds, their method's prediction accuracy is 73 %, 69 %, and 61 %, respectively, which indicates that their program is limited for proteins with a fewer number of disulfide bonds. Vincent et al. introduced a method for predicting disulfide bridges using two decomposition kernels to measure the similarity between protein sequences according to the amino acid environments around cysteines (Vincent et al., 2008).

In the recent past, Zhu et al. (2010) applied both global and local sequential and structural features of proteins to predict disulfide bonds using support vector regression, which achieved the prediction accuracy of about 76 %. In their work, the authors highlight the use of three different filter-based feature selection methods, namely, variance score, Laplacian score, and Fisher score. Lin and Tseng developed a method for disulfide bonding connectivity pattern prediction using the coordinates of the alpha-carbon of each residue to compute the normalized pair distance and use it as an input to the SVM (Lin and Tseng, 2010). More recently, Niu et al. (2013) devised a method for inter- and intra-chain disulfide bond prediction using the nearest neighbor algorithm (NNA) by optimal feature selection based on maximum relevance and minimum redundancy (mRMR) followed by incremental feature selection (IFS). This method utilizes features such as sequence conservation, residual disorder, and amino acid factor for an inter-chain disulfide bond prediction. Moreover, for an intra-chain disulfide bond prediction, the features used for inter-chain along with the sequential distance between a pair of cysteines are utilized. Table 1 summarizes the recent works on disulfide bond prediction.

In this work, we established an advanced machine learning technique called stacking to predict disulfide bonds. The prediction of disulfide bonds is achieved in two stages: first, the bonding state of individual cysteines is predicted; second, the disulfide bonding of cysteine-pairs is predicted by including the results from individual cysteine bonding state prediction as a feature. Various useful sequence and structure-based features are extracted and used for individual cysteine and cysteine-pairs disulfide bond prediction. For predicting single or individual cysteine bonding, features such as residue profile, physiochemical profile, conservation profile, structural profile, flexibility profile, and position-specific energy profile are used. In addition to these features, the sequential distance between a pair of cysteines and individual cysteine bonding state probability is also used for cysteine-pairs bonding prediction. For individual cysteine and cysteine-pair disulfide bonding prediction, *diSBPred* attains 10-fold cross-validation balanced accuracy of 82.29 % and 94.20 %, respectively. Altogether, our method achieves an overall improvement of 43.25 % based on balanced accuracy compared to the existing NNA based approach. These results indicate that the *diSBPred* can be utilized to annotate the sequences whose structure has not been experimentally determined and facilitate the aiPSP method, which can further aid in experimental studies of the disulfide bond structure determination. Fig. 1 illustrates the workflow of our proposed method.

Table 1
Recent works on the prediction of disulfide bonding state.

Authors	Method
Fariselli and Casadio (2001)	Monte Carlo simulated annealing
Cheng et al. (2005)	Neural network
Vullo and Frasconi (2004)	Neural network
Ferrè and Clote (2005)	Neural network
Ferrè and Clote (2006)	Support vector machine (SVM)
Song et al. (2007)	Support vector regression
Rubinstein and Fiser (2008)	Correlated mutation patterns
Vincent et al. (2008)	Decomposition kernels
Zhu et al. (2010)	Support vector regression
Lin and Tseng (2010)	Support vector machine (SVM)
Niu et al. (2013)	Nearest neighbor algorithm (NNA)

2. Materials and methods

This section presents our benchmark/training dataset preparation

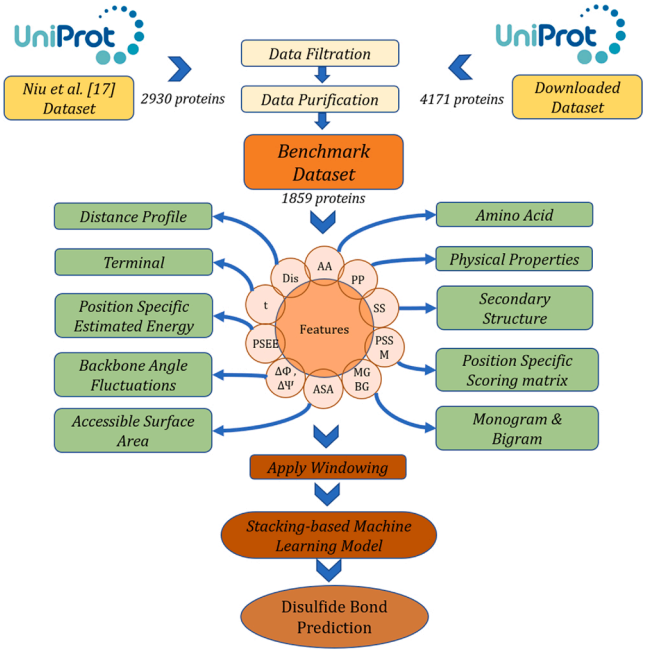

Fig. 1. Illustrates the workflow of the disulfide bond prediction.

approach, feature mining, performance assessment, and machine learning-based disulfide bond predictor development.

2.1. Dataset

We collected the benchmark dataset established previously by Niu et al. (2013). It was reported that the dataset consists of 2930 proteins. However, we were able to collect only 2674 unique protein ids. These unique UniProt ids were searched in the UniProt (Consortium, 2007) database to obtain protein sequences and disulfide annotation. During our search of these proteins in UniProt, we found some of the proteins marked as obsolete. Such obsolete sequences were filtered from the dataset for further consideration. Therefore, the search for 2674 protein ids resulted in 2671 protein sequences. Furthermore, the following filtering strategies were adopted to obtain a good quality training/validation dataset: (i) filter sequences less than 50 amino acid long; (ii) filter sequences containing keyword "alternate" under disulfide labels as it was seen that the same residue was labeled to form a disulfide bond with multiple other cysteines within the sequence which could mislead the learning algorithm; (iii) filter sequences not containing at least one disulfide bond; (iv) filter sequences containing non-standard amino acids and (v) filter sequences which contain "?" or ">" character instead of the residue index of the disulfide bonding cysteine. This resulted in 2276 protein sequences, and we call this set "Set-A."

Next, we downloaded 4171 protein sequences containing disulfide bonds of any experimentally validated type and sequence length < 5000 from the UniProt database. As employed above, a similar filtration strategy was used to obtain a good quality sequence of 3474 proteins and we call this set "Set-B." Next, the sequences from Set-A, which are common to Set-B, were removed from Set-A. This resulted in 49 sequences in Set-A. We also noted that the dataset prepared by Niu et al. consists of homologous sequences. The presence of homologous sequences in the dataset could lead to the design of the biased predictive model. Thus, we utilized BLASTCLUST (Camacho et al., 2009) to cut off those sequences that have $\geq$ 25 % sequence identity to any other in Set-A. This step yielded 33 proteins in Set-A. Likewise, we again used the BLASTCLUST to cut off those sequences that have $\geq$ 25 % sequence identity to any other in Set-B, which yielded 1859 proteins. Finally, the sequences in Set-A and Set-B were combined, and the BLASTCLUST was used again to remove those sequences that have $\geq$ 25 % sequence identity to any other in this combined set. A final dataset, called DBD1866, which consists of 1866 non-homologous protein sequences, is obtained through this process. The DBD1866 consists of 23187 cysteine residues, of which 16104 are disulfide bonding cysteines, and the remaining 7083 are non-disulfide bonding cysteines. We took all 16104 disulfide bonding cysteines and 7083 non-disulfide bonding cysteines as positive and negative samples, respectively, to predict the binding state of individual cysteines.

Moreover, we calculated all cysteine pairs within each sequence for the prediction of disulfide bonds, resulting in a total of 495570 cysteine pairs. Among 495570, 8056 cysteine pairs with known disulfide bonds are considered positive samples, and the remaining 487514 cysteine pairs are considered the candidates for negative samples. The statistics of the sequence distance between paired cysteines shows that for 94.5 % of the positive samples, the sequence distance between paired cysteines is less than 100 residues. Thus, we selected 8056 cysteine pairs as positive samples and 40280 (since 5 folds imply $8056 \times 5 = 40280$) out of 487514 cysteine pairs with a distance of fewer than 100 residues as negative samples to form an imbalanced dataset, which is referred to here as Imb-DBD.

2.2. Feature extraction

To create an effective machine learning method to predict individual cysteines disulfide bonding state and subsequently predict disulfide bonds from sequence information only, we use various features derived from residue profile, physiochemical profile, conservation profile, structural profile, flexibility profile, and position-specific energy profile, described next. Fig. 2 shows the encoding of the protein sequence into a feature vector utilizing various feature extraction tools.

2.2.1. Residue profile

Twenty different standard amino acid (AA) types are encoded using 20 different numerical values yielding one feature per amino acid. This feature is useful to capture the amino acid composition of residues in an environment that is local to the cysteine residue. The importance of this feature in solving problems in bioinformatics has been demonstrated by previous studies (Iqbal et al., 2015; Iqbal and Hoque, 2018; Iqbal and Hoque, 2016). Further, we encoded terminal (T) residues, five residues from N and C terminal by -1.0 to -0.2, and +0.2 to +1.0, separately with a step size = 0.2, giving one feature per residue (Iqbal et al., 2015).

2.2.2. Physiochemical profile

The physiochemical properties (PP) of a protein are determined by the amino acids' corresponding properties in it. The effect of physiochemical properties of amino acids on post-translational modification (PTM) has been presented by previous studies (Zhu et al., 2010; Niu et al., 2010). In this work, five highly compact numeric patterns reflecting polarity, secondary structure, molecular volume, codon diversity, and electrostatic charge are extracted from (Zhu et al., 2010) and used as features to represent the respective properties of each amino acid.

2.2.3. Conservation profile

The evolutionary conservation profile plays an important role in PTM, including disulfide bond formation (Zhu et al., 2010). In this study, the protein sequence's conservation profile is obtained in terms of a normalized position-specific scoring matrix (PSSM) from the DisPredict2 program (Iqbal and Hoque, 2016). DisPredict2 internally executes three iterations of the position-specific iterative blast (PSI-BLAST) (Altschul et al., 1990) against NCBI's non-redundant database to generate a PSSM profile and subsequently normalizes it by dividing each value by a value of 9. The PSSM is a matrix of $L \times 20$ dimensions, which captures the conservation pattern in multiple alignment and stores the scores for each position in the alignment, where L is the length of the protein sequence. High scores indicate more conserved positions whereas, scores close to zero or negative indicate a faintly conserved position. Each amino acid in a protein sequence is encoded by a 20-D feature vector in our study.

The PSSM score was further extended to compute monogram (MG) and bi-gram (BG features. The MG and BG features can be utilized to characterize a protein sequence segment that can be conserved within a fold in terms of transition probabilities from one amino acid to another (Sharma et al., 2014). Thus, these features can be useful in recognizing the evolutionary folded (ordered) or unfolded (disordered) region of proteins that could occur due to the formation of a disulfide bond, which motivated us to utilize them as features in this work. We extracted 1-D MG and 20-D BG features from the DisPredict2 program and used them in this work.

2.2.4. Structural profile

Local structural properties such as predicted secondary structure (SS) and accessible surface area (ASA) of amino acids have been widely used to solve various biological problems, including the prediction of disulfide bonds. Here, we used the DisPredict2 program to obtain predicted ASA and SS probabilities for helix (H), coil (C), and beta-sheet (E) at the residue level. The DisPredict2 program internally uses SPINE-X (Faraggi et al., 2012) program to compute ASA and SS probabilities from a given protein sequence. In addition, a separate set of SS probabilities for E, C, and E at residue level was obtained from BalancedSSP (Islam et al., 2016) program as it provides a balanced prediction of these SS types. It was noted that the BalancedSSP predicts a higher number of beta-sheet

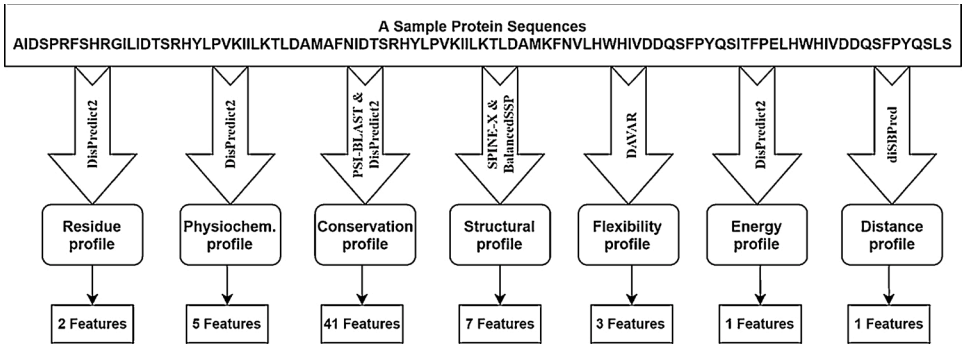

Fig. 2. Illustration of encoding the protein sequence into a feature vector utilizing various feature extraction tools.

correctly compared to the other existing SS predictors. Thus, seven total structural properties, including six predicted SS probabilities and one ASA per amino acid, were extracted and used as a structural profile in this work.

2.2.5. Flexibility profile

Protein molecule shows varying degrees of flexibility throughout their 3D structures, which is often represented by the fluctuation of the Cartesian coordinates of the protein backbone defined by two torsion angles Φ and Ψ. The predictor of backbone torsion angle fluctuation has been useful in the development of several computational methods and predictors of secondary (Islam et al., 2016) and supersecondary (Chen and Kurgan, 2012) structure, short and long disordered regions (Zhang et al., 2012), protein-peptide binding domain (Iqbal and Hoque, 2018), the accessible surface area of protein residues (Tarafder et al., 2018), and more. We obtained two backbone angle fluctuation features, dphi (ΔΦ) and dpsi (ΔΨ) from the DAVAR (Zhang et al., 2010) program.

Moreover, intrinsically disordered region (IDR) in protein exhibit flexibility due to a lack of fixed or ordered 3D structure. Previous studies have demonstrated that IDR contains PTM sites, sorting signals and play an important role in regulating protein structures and functions (Wright and Dyson, 1999; Liu et al., 2002; Tompa, 2002). In this study, we encoded each amino acid in a protein with a disorder probability obtained from a disorder predictor, called DisPredict2, which can predict the protein's disordered regions accurately.

2.2.6. Energy profile

Iqbal and Hoque (2016) proposed a novel approach to estimate position-specific estimated energy (PSEE) of amino acid residue from sequence information alone using contact energy and predicted relative solvent accessibility (RSA). They demonstrated that the PSEE helps identify the structured and unstructured or intrinsically disordered regions of a protein. Furthermore, PSEE can also be used to detect the existence of functional binding regions of a protein. Due to its experimentally proven potential in solving several biological problems, we used the PSEE score per amino acid as a feature in our study.

2.2.7. Distance profile

Through an optimal feature selection using the mRMR technique, Niu et al. demonstrated that the sequence distance between the paired cysteine sites plays a significant role in intra-chain disulfide bond prediction. Thus, we use sequence distance between paired cysteines as one of the features in our study.

2.3. Feature window selection

Here, we applied a widely used feature windowing technique to include the neighboring residue features with the features of the cysteine site to train the predictor. We examined a suitable size of the sliding window that determines the number of residues around a target cysteine residue, which can mediate the interactions between the pairing cysteines. We designed several models with different window sizes (*ws*) (1, 3, 5, and so on). The *ws* of 1 indicates no features from neighboring amino acids will be included with the features of the cysteine site. Whereas, the *ws* of 3 indicates that the features of one amino acid before and after the cysteine site will be included with the features of the cysteine site to train the predictor.

2.4. Feature space

A slightly different set of features were used for individual cysteine bonding state prediction and disulfide bond prediction. Further, the difference in the number of features comes from the windowing of the features. The feature space for individual cysteine bonding state prediction and disulfide bond prediction are discussed in detail below.

2.4.1. For cysteine bonding state prediction

Features including 1 terminal indicator, 20 PSSM scores, 1 MG score, 20 BG score, 6 SS probabilities, 1 ASA, 1 ΔΦ, 1 ΔΨ, 1 disorder probability, 1 PSEE score, totally 53 features were used for cysteine site. Additionally, for each of the neighboring residues, 1 terminal indicator, 20 PSSM scores, 1 MG score, 20 BG score, 6 SS probabilities, 1 ASA, 1 ΔΦ, 1 ΔΨ, 1 disorder probability, 1 PSEE score, 1 amino acid type, 5 physiochemical properties, totally 59 features were used. Thus, for *ws* of 1, the cysteine residue is represented by 53 features, whereas, for *ws* of 3, the cysteine residue is represented by $2 \times 59 + 53 = 171$ features and so on.

2.4.2. For disulfide bond prediction

The absolute values of the sum and difference of the features including 1 terminal indicator, 20 PSSM scores, 1 MG score, 20 BG score, 6 SS probabilities, 1 ASA, 1 ΔΦ, 1 ΔΨ, 1 disorder probability between each pair of cysteine sites, resulting in a total of $2 \times 53 = 106$ features were calculated. Moreover, for each of the neighboring residues around the cysteine sites, the absolute value of sum and difference of the features including 1 terminal indicator, 20 PSSM scores, 1 MG score, 20 BG score, 6 SS probabilities, 1 ASA, 1 ΔΦ, 1 ΔΨ, 1 disorder probability, 1 PSEE score, 5 physiochemical properties, resulting in a total of $2 \times 58 = 116$ features were obtained. Further, the absolute values of the sum and difference of the individual cysteine bonding probabilities were obtained, which give us 2 features. Finally, the sequence distance between the paired cysteine sites was included as a feature. Thus, for *ws* of 1, the feature vector contains $(106 + 2 + 1 = 109)$ features. For, $ws > 1$, the amino acid type of the neighboring amino acids to both the cysteines of the cysteine pair is used as features directly. Thus, for *ws* of 3, the feature vector contains $(116 \times 2 + 106 + 2 + 1 + 2 = 343)$ features and so on.

2.5. Performance evaluation

To evaluate the performance of our predictor, *diSBPred*, we applied 10-fold cross-validation (CV) approach. In a 10-fold CV, the dataset is segmented into 10 equal-size parts. While a fold is set aside for testing, the rest of the folds are used to train the predictor. This process is repetitive until each of the fold is tested once, and subsequently, the test accuracy of each fold is combined to find the average (Hastie et al., 2009). We used various performance evaluation metrics listed in Table 2 to assess our predictor. Moreover, we used a jackknife validation approach to compare our predictor with the existing method. In jackknife validation, every sample is tested by the predictor trained with the samples' remaining in the dataset.

2.6. Stacking framework of disulfide bond prediction

To develop the disulfide bond predictor(*diSBPred*), we adopted an idea of a stacking based machine learning approach (Wolpert, 1992) which, has recently been successfully applied to solve various bioinformatics problems (Iqbal and Hoque, 2018; Mishra et al., 2018; Hu et al., 2015; Nagi and Bhattacharyya, 2013). Stacking is an ensemble-based machine learning approach, which collects information from multiple models in different phases and combines them to form a new model. Stacking is considered to yield more accurate results than the individual machine learning methods as the information gained from more than one predictive model minimizes the generalization error. The stacking framework includes two-levels of classifiers, where the classifiers of the first-level are called base-classifiers, and the classifiers of the second-level are called meta-classifiers. In the first level, a set of base-classifiers C_1, C_2, …, C_N is employed (Džeroski and Ženko, 2004). The base-classifiers' prediction probabilities are combined using a meta-classifier to reduce the generalization error and improve the predictor's accuracy. To enrich the meta-classifier with necessary information on the problem space, the base-level classifiers are selected. Their underlying operating principle is different from one another (Mishra et al., 2018; Nagi and Bhattacharyya, 2013).

To select the classifiers to use in the first and second level of the *diSBPred* stacking framework, we analyzed the performance of eight individual classification methods: *i)* Random Decision Forest (RDF) (Ho, 1995); *ii)* Bagging (Bag) (Breiman, 1996); *iii)* Extra Tree (ET) (Geurts et al., 2006); iv) Neural Network (NN) (McCulloch and Pitts, 1943; Newell, 1969); v) Logistic Regression (LogReg) (Hastie et al., 2009; Szilágyi and Skolnick, 2006); and vi) K-Nearest Neighbor (KNN) (Altman, 1992), vii) Light Gradient Boosting Machine (LightGBM) (Ke et al., 2017) and viii) Support vector machine (SVM) (Vapnik, 1995). The algorithms and their configuration details are briefly discussed below.

Table 2
Name and definition of the evaluation metric.

Name of Metric	Definition
True Positive (TP)	Correctly predicted positive samples
True Negative (TN)	Correctly predicted negative samples
False Positive (FP)	Incorrectly predicted positive samples
False Negative (FN)	Incorrectly predicted negative samples
Recall/Sensitivity/True Positive Rate (SN)	$\frac{TP}{TP+FN}$
Specificity/True Negative Rate (SP)	$\frac{TN}{TN+FP}$
Fall Out Rate /False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
Miss Rate/False Negative Rate (FNR)	$\frac{FN}{FN+TP}$
Accuracy (ACC)	$\frac{TP+TN}{TP+FP+TN+FN}$
Balanced Accuracy (BACC)	$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$
Precision (PR)	$\frac{TP}{TP+FP}$
F1-score (Harmonic mean of precision and recall)	$\frac{2TP}{2TP+FP+FN}$
Mathews Correlation Coefficient (MCC)	$\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FN)\times(TP+FP)\times(TN+FP)\times(TN+FN)}}$

i) RDF: RDF (Ho, 1995) constructs a multitude of decision trees, each of which is trained on a random subset of the training data. The sub-set used to create a decision tree is constructed from a given set of observations of training data by taking 'm' observations at random and with replacement (a.k.a. Bootstrap Sampling). Next, the final predictions are achieved by aggregating the prediction from the individual decision trees. For classification, the final prediction is made by computing the mode (the value that appears most often) of the classes. In our implementation of the RDF, we used bootstrap samples to construct 1,000 trees (n_estimators = 1,000) in the forest, and the rest of the parameters were set to their default value.

ii) Bag: Bag (Breiman, 1996) machine learning algorithm operates by forming a class of algorithms that creates several instances of a base classifier/estimator on random subsets of the training samples and consequently combines their individual predictions to yield a final prediction. It reduces the variance in the prediction. In our study, the BAG classifier was fit on multiple subsets of data using Bootstrap Sampling using 1,000 decision trees (n_estimators = 1,000), and the rest of the parameters were set to their default value.

iii) ET: Extremely randomized tree (ET) classifier (Geurts et al., 2006) operates by fitting several randomized decision trees (a.k.a. extra-trees) on various sub-sets and uses averaging to improve the prediction accuracy and control over-fitting. In our implementation, the ETC model was constructed using 1,000 trees (n_estimators = 1,000), and the Gini impurity index assessed the quality of a split. The rest of the parameters were set to their default value.

iv) NN: Neural networks (NNs) (McCulloch and Pitts, 1943; Newell, 1969) are a non-linear statistical data modeling tool also called artificial neural networks. They are composed of interconnected nodes that can model complex relationships between inputs and outputs. The nodes are called artificial neurons, similar to neurons in the human brain. The connections are called edges and used to transmit signals to nodes/-neurons. Neurons and edges typically have a weight that adjusts as learning proceeds. Recently, a neural network has been applied to massive data sets in a variety of fields such as computer vision (Simonyan and Zisserman, 2015), natural langue processing (Devlin et al., 2019), and protein structure prediction (Senior et al., 2019) and performs very well. We implement a convolutional neural network with three convolution layers and two dense layers using the python Keras library in our implementation.

v) LogReg: LogReg (a.k.a. logit or MaxEnt) (Hastie et al., 2009; Szilágyi and Skolnick, 2006) is a machine learning classifier that measures the relationship between the categorical dependent variable and one or more independent variables by generating an estimation probability using logistic regression. In our implementation, we set all the parameters of LogReg to their default values.

vi) KNN: KNN (Altman, 1992) is a non-parametric and lazy learning algorithm. Non-parametric means it does not make any assumption for underlying data distribution; rather, it creates models directly from the dataset. Furthermore, lazy learning means it does not need any training data points for a model generation rather uses the training data while testing. It works by learning from the K number of training samples closest in the distance to the target point in the feature space. The classification decision is made based on the majority-votes obtained from the K nearest neighbors. Here, we set the value of K to 9 and the rest of the parameters to their default value.

vii) LightGBM: LightGBM (Ke et al., 2017) follows the gradient boosting framework that uses tree-based learning algorithms. The algorithm has a faster training speed, higher efficiency, and lower memory usage. It also supports parallel and GPU learning and capable of handling large-scale data. In our implementation, the LightGBM model was constructed using 1000 trees (n_estimators = 1000), and the rest of

the parameters were set to their default value.

viii) SVM: SVM (Vapnik, 1995) classifier with RBF (radial basis function) kernel operates by maximizing the class-separating hyperplane or the margin between the two classes and penalizes the instances on the wrong side of the decision boundary. Because SVM simultaneously minimizes the empirical classification error (i.e., training error) and generalization error (i.e., test error) by maximizing the geometric margin of the separating hyperplane, it is regarded as an effective technique in hard classification problems, especially in bioinformatics and computation biology area. Kernels in SVM classification refer to the function that is responsible for defining the decision boundaries between the classes. An RBF (radial basis function) kernel is used when the boundaries are hypothesized to be curve-shaped rather than straight. RBF kernel uses two main parameters, gamma and C that are related to the decision region (how spread the region is) and the penalty for misclassifying a data point, respectively. It is crucial to identify the proper combination of SVM parameters (C and γ) to achieve good classification performance. In our implementation, the grid search (Hastie et al., 2009) technique is used to optimize the RBF kernel parameter γ and the cost parameter, C, to achieve the highest 10-fold CV accuracy.

All the classification methods mentioned above are built using python's Scikit-learn, Tensorflow, and Keras library (Pedregosa et al., 2012). In order to design a stacking framework for *diSBPred*, we evaluated the different combinations of base-classifiers and finally selected the one that provided the highest performance.

The set of stacking framework tested are:

i SF1: RDF, LightGBM, LogReg, KNN in base-level and LightGBM in meta-level,
ii SF2: ET, LightGBM, LogReg, KNN in base-level, and LightGBM in meta-level and
iii SF3: Bag, LightGBM, LogReg, KNN in base-level and LightGBM in meta-level.

Here, the choice of base-level classifiers is made such that the underlying principle of learning of each of the classifiers is different from each other (Mishra et al., 2018). For example, in SF1, SF2, and SF3, the tree-based classifiers RDF, Bag, and ET are individually combined with the other two methods LogReg and KNN, to learn different information from the problem-space. Additionally, for each of the combinations SF1, SF2, and SF3, the LightGBM classifier is used both in the base as well as in the meta-level because it performed best among all the other individual methods applied in this work. While examining the 10-fold CVs performance of the above three combinations, we found that the first stacking framework, SF2 attains the highest performance based on the balanced accuracy. Therefore, we employ four classifiers, ET, LightGBM, LogReg, and KNN as the base classifiers and another LightGBM as the meta-classifier in the *diSBPred* stacking framework. Fig. 3 shows the proposed stacking framework of the disulfide bond prediction.

3. Results

Here, first, we discuss the feature importance for different feature profiles and the performance of two different models created using the existing NNA (Niu et al., 2013) based method utilizing the features reported in the same work and the features used in this study. Then, we present the performance of individual cysteine bonding state prediction. Subsequently, we report the disulfide bond prediction performance on

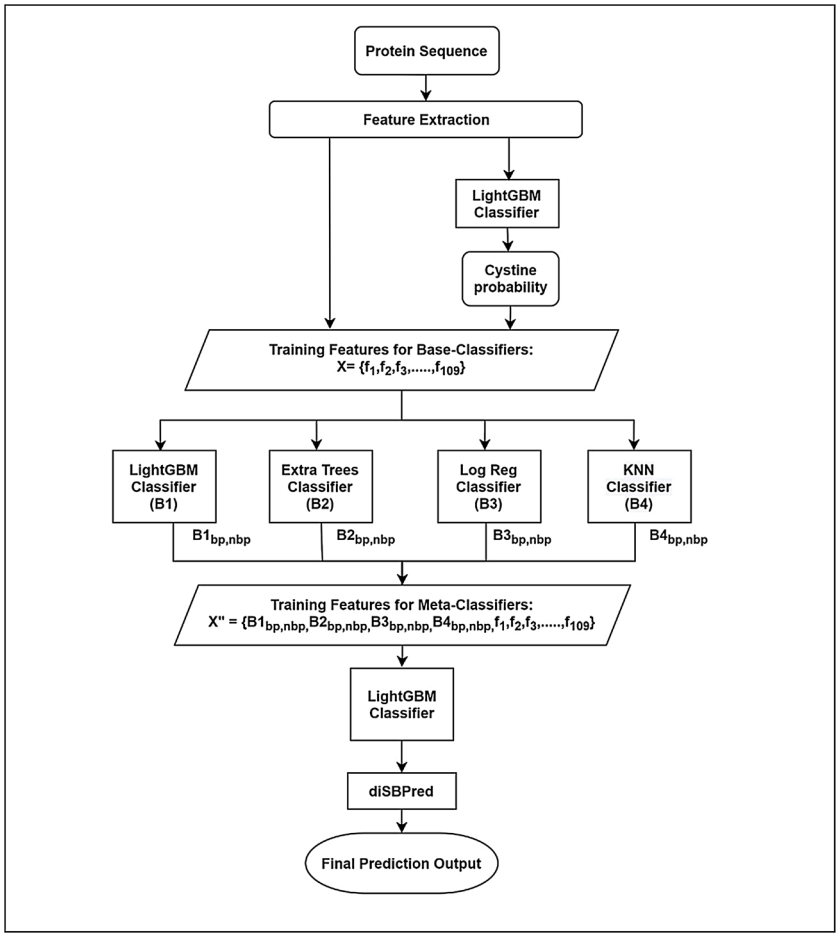

Fig. 3. Illustrates the prediction framework of the disulfide bond prediction.

the Imb-DBD dataset while the predictions from individual cysteine bonding states are included as a feature. Finally, we demonstrate the comparison of our predictor with the existing NNA based predictor.

3.1. Feature importance

To identify the relevance of the features used in our work, we implemented the NNA (Niu et al., 2013) based algorithm locally. We took 2276 proteins that we obtained after the filtration steps mentioned under the Dataset section and further performed BLASTCLUST to remove the homologous sequences that have $\geq$ 25 % sequence identity. This filtration yielded 1217 non-homologous protein sequences. Then, we took all the cysteine pairs corresponding to these 1217 proteins from the original dataset collected from Niu's work. These 1217 proteins consist of 18882 cysteine pairs, of which 3513 were labeled as disulfide bonding (positive samples), and the remaining 15369 were labeled as non-disulfide bonding (negative samples). Niu et al. considered a segment of 9 residues (including cysteine itself in the center, 4 residues upstream, and 4 residues downstream) as the mini-environment of each cysteine. As in Niu's work, for the 4 residues upstream and 4 residues downstream, we extracted the absolute values of the sum and difference of the 20 PSSM score, 5 physiochemical properties, and 1 disorder score. Likewise, the absolute values of the sum and difference of the 20 PSSM score and 1 disorder score were extracted for the cysteine sites.

Additionally, the sequence distance between the paired cysteine sites was also included as a feature. We refer to these features as feature set one (fs1). Subsequently, the fs1 was used to construct a model based on the NNA method. On the other hand, for the same dataset, we extracted all the features for disulfide bond prediction used in this study (see discussion in Section "For disulfide bond prediction"). We refer to these features as feature set two (fs2). Next, we used the fs2 to construct another NNA based model. The jackknife validation balanced accuracy of the two NNA based models (NNA-Model1 and NNA-Model2) constructed using the feature sets fs1 and fs2, respectively, are shown in Table 3.

The comparison in Table 3 shows that NNA-Model2 created using the fs2 (features proposed in this study) provides a 3.07 % improvement over NNA-Model1. This clearly indicates that the features used in this study are useful for the prediction of disulfide bonds and results in better performance.

To find out which feature profile contributed most to our proposed model's performance improvement, we calculated the feature importance for each of the feature profiles using the LightGBM classifier. The logic behind choosing the LightGBM classifier is that it performed best compared to the rest of the individual classifiers studied in this work. Fig. 4 shows the aggregated feature importance for each of the feature profiles used for disulfide bond prediction in our work. The number of times a feature is used to split the data across all trees in the LightGBM (Ke et al., 2017) method is considered as the importance of that feature. Moreover, Fig. 4 shows that the conservation profile feature group is the most important feature, and the residue profile feature group is the least important one.

In addition, we tested the impact of the incremental addition of feature profiles on the performance of the LightGBM classifier. Starting with the least important feature group from Fig. 4, which is residue profile, we build several LightGBM classifiers by adding one feature group in the feature vector at a time through 10-fold cross-validation on the benchmark dataset (Table 4).

Table 3
Comparison of the two NNA based models constructed using the feature sets fs1 and fs2 through jackknife validation.

Metric	NNA-Model1	NNA-Model2	(imp. %)
SN %	43.90	**52.60**	(19.18 %)
SP %	87.62	**88.63**	(1.53 %)
BACC %	65.76	**70.62**	(7.39 %)
ACC %	79.49	**81.93**	(3.07 %)

Here, 'imp.' stands for improvement. The 'imp. %' represents an improvement in percentage achieved by NNA-Model2 over the NNA-Model-1. The best score values are **bold**faced.

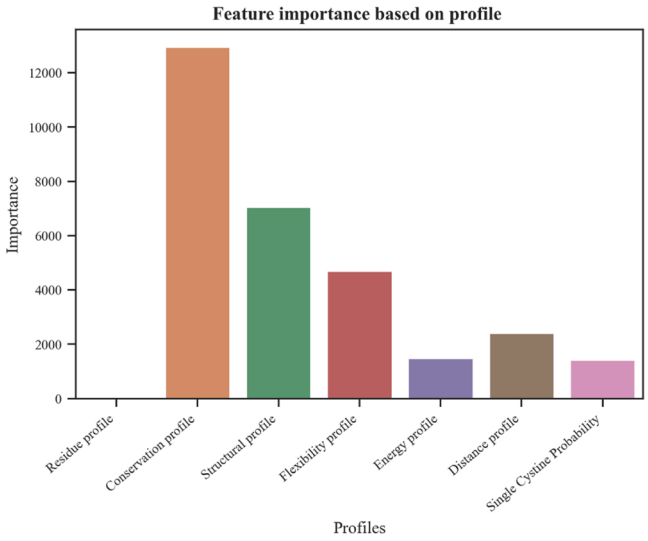

Fig. 4. Feature importance for different feature profiles.

Table 4 shows that incrementally adding the feature group into the feature vector improves the performance of the LightGBM classifier obtained through 10-fold cross-validation on the benchmark dataset. The improvement in the performance of the LightGBM classifier obtained by the sequential addition of feature group into feature vector indicates that all the features implemented in our study are useful. Notably, we can observe that the addition of energy profile features itself improved the balanced accuracy from 53.50%–73.11%. Further, we can also observe from Table 4 that all the feature profiles have some contributions to achieve the final balanced accuracy of 89.62 %. The outcomes in Table 4 also suggest that the feature set that includes individual cysteine bonding state probabilities provides a 1.89 % improvement in balanced accuracy compared to the initial set (residue profile)."

3.2. Cysteine bonding state prediction

The information directly related to the bonding and non-bonding state of the individual cysteine residues might help improve disulfide bond prediction accuracy. Considering the same, we first trained eight different machine learning models (LightGBM, KNN, LogReg, Bag, RDF, ET, SVM, and NN) to predict the bonding non-bonding state of the individual cysteine residues. The performance comparison of the individual classifiers is shown in Table 5.

Table 5 shows that the LightGBM is the best performing classifier among eight different classifiers implemented in our study in terms of balanced accuracy, F1-score, and MCC. So, we developed a LightGBM based predictor to predict the bonding state of the cysteine residues. Then, the prediction probabilities of the bonding state of the cysteine residues were included in the feature set to improve the disulfide bond predictor's performance. To identify the best window size of the features, for which the bonding state predictor yields the highest 10-fold CV balanced accuracy on the DBD1866 dataset, several LightGBM models with different window sizes were created. The performance comparison of the models built using different sliding window sizes is shown in Fig. 5.

The results in Fig. 5 show that the balanced accuracy of the cysteine bonding state predictor improves with the window size increment, which highlights that the inclusion of neighboring residue information helps the predictor learn about a target residue. It is also evident from

Table 4
Contribution of features on the performance of the LightGBM classifier obtained through 10-fold cross-validation on the benchmark dataset.

Feature Set	SN (%)	SP (%)	FPR	FNR	PR (%)	F1-score	MCC	BACC (%)	ACC (%)
Residue profile	5.42	99.60	0.004	0.946	73.08	0.101	0.169	52.51	83.90
+ Single Cystine Probability	8.68	98.32	0.017	0.913	50.87	0.148	0.157	53.50	83.38
+ Energy profile	51.04	95.19	0.048	0.490	67.96	0.583	0.521	73.11	87.83
+ Distance profile	62.92	95.53	0.045	0.371	73.80	0.679	0.624	79.23	90.10
+ Flexibility profile	67.12	96.60	0.034	0.329	79.80	0.729	0.684	81.86	91.69
+ Structural profile	73.05	97.29	0.027	0.269	84.37	0.783	0.746	85.17	93.25
+ Conservation profile	**79.84**	**99.39**	**0.006**	**0.202**	**96.32**	**0.873**	**0.856**	**89.62**	**96.13**

The best score values are **bold**faced.

Table 5
Performance of cysteine bonding state prediction model for different classifiers.

Metric/Methods	LightGBM	KNN	LogReg	Bag	RDF	ET	SVM	NN
SN (%)	92.76	91.70	92.08	92.57	93.25	93.11	**93.99**	91.83
SP (%)	**70.92**	54.89	52.29	68.28	67.03	65.95	53.64	55.46
FPR	**0.291**	0.451	0.477	0.317	0.330	0.341	0.464	0.445
FNR	0.072	0.083	0.079	0.074	0.067	0.069	**0.060**	0.082
PR (%)	**87.88**	82.21	81.44	86.90	86.54	86.14	82.17	82.42
F1-score	**0.903**	0.867	0.864	0.896	0.898	0.895	0.877	0.869
MCC	**0.663**	0.514	0.498	0.639	0.640	0.629	0.543	0.521
BACC (%)	**81.84**	73.29	72.19	80.42	80.14	79.53	73.81	73.64
ACC (%)	**86.09**	80.45	79.92	85.15	85.24	84.81	81.66	80.72

The best score values are **bold**faced.

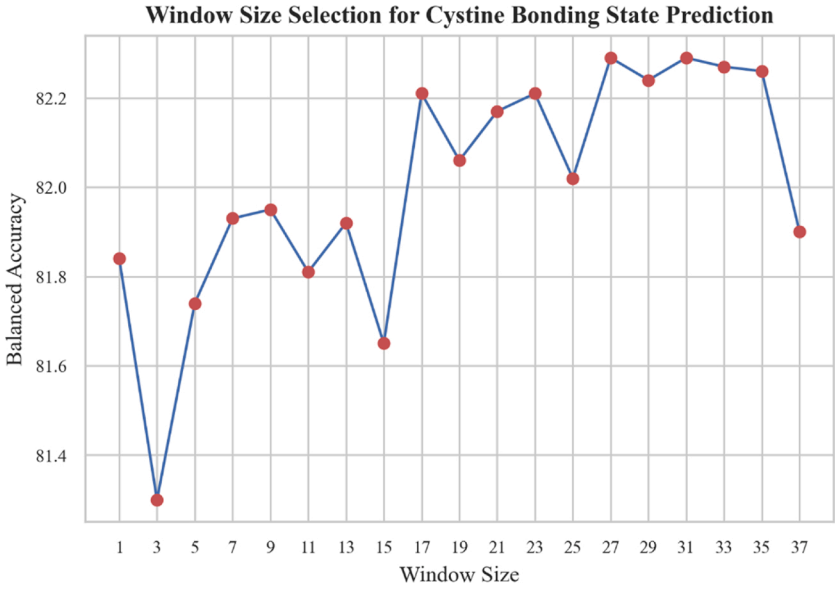

Fig. 5. Performance comparisons of the LightGBM models trained using feature vectors created by applying a sliding window of different sizes. The balanced accuracies of the models for various window sizes are reported.

Fig. 5 that the balanced accuracy significantly increases from window size 1–27. Since the balanced accuracy of the models with higher window sizes ($ws > 27$) differ after second or third decimal places, we compare the MCC and balanced accuracy to select the window size 31 as the best window size for cysteine bonding state prediction. The MCC for the model for window sizes 27 and 31 are 0.689 and 0.692, respectively, while both the models have the same balanced accuracy of 87.29 %.

Moreover, Table 6 shows the values obtained for additional performance measures while performing the 10-fold CV of the model trained with the best window size of 31. This model achieves sensitivity, specificity, precision, f-measure, MCC, BACC and ACC of 95.30 %, 69.28 %, 87.58 %, 0.913, 0.692, 82.29 %, and 87.35 %, respectively. The model's sensitivity is significantly high because the number of positive samples (counts 16104) in the DBD1866 dataset is 2.274 times higher than the

Table 6
Performance of cysteine bonding state prediction model, trained with the best window size.

Metric	Model Trained with the Best Window Size
SN (%)	95.30
SP (%)	69.28
FPR	0.307
FNR	0.047
PR (%)	87.58
F-measure	0.913
MCC	0.692
BACC (%)	82.29
ACC (%)	87.35

number of negative samples (counts 7083).

3.3. Disulfide bond prediction

For the prediction of disulfide bonds, we select the base and meta classifier for our stacking method based on the performance of the individual classifiers. We include the probabilities of the individual cysteine bonding state prediction as a feature for disulfide bond prediction. The results of these experiments are discussed below.

3.4. Selection of classifiers for stacking

To select the methods to use as the base and the meta-classifiers, we analyzed the performance of eight different machine learning algorithms: LightGBM, KNN, LogReg, Bag, RDF, ET, SVM, and NN on the benchmark dataset through a 10-fold CV approach. The performance comparison of the individual classifiers on the benchmark dataset is shown in Table 7.

Table 7 further shows that LightGBM is the best performing classifier among eight different classifiers implemented in our study regarding sensitivity, balanced accuracy, FNR, and F1-score. The LightGBM attains sensitivity, balanced accuracy, accuracy, FNR, F1-score, and MCC of 79.77 %, 89.54 %, 96.05 %, 0.202, 0.871, and 0.852, respectively. As the dataset is highly imbalanced, we consider balanced accuracy as the deciding score as it provides the balanced measure of any predictor trained on an imbalanced dataset. Furthermore, it is evident from Table 7 that the balanced accuracy of the LightGBM is 19.66 %, 12.14 %, 0.19 %, 3.84 %, 4.38 %, 8.48 %, and 8.56 % higher than KNN, LogReg, Bag, RDF, ET, SVM, and NN, respectively. The greater performance of the LightGBM algorithm motivated us to use it both as a base as well as a meta-classier and selection of sliding window size in the diSBPred prediction framework.

To identify the best window size for which the disulfide bonds prediction model yields the highest 10-fold CV balanced accuracy on the Imb_DBD dataset, several LightGBM models for different window sizes were created. Finally, the LightGBM model that corresponds to the best window size for the Imb_DBD dataset was identified. Fig. 6 shows the performance comparison of the LightGBM models created using different window sizes for the Imb_DBD dataset.

Fig. 6 illustrates that the LightGBM model evaluated on window size 13 provides the highest balanced accuracy of 92.66 %. Initially, the performance line indicates increasing performance with the window size, up to the window size 7. With the window size 9 and 11, the model's performance decreases slightly compared to the model with window size 7. However, the next model with window size 13 achieves the highest performance. Moreover, the models with window sizes larger than 9 show decreasing performance. The corresponding MCC for window size 13 is 0.904, which was the highest MCC compared to the other models. Thus, we select the window size 13 to train the stacking based model with LightGBM as the meta classifier on the Imb_DBD dataset for disulfide bond prediction.

We adopted base-classifier selection guidelines based on different underlying principles to select the classifiers to be used at the base-level. Therefore, we used KNN and LogReg as two additional classifiers at the base-level. Then, we added a single tree-based ensemble method out of three methods, RDF, Bag, and ET, at a time as the fourth base-classifier and designed three different combinations of stacking framework, namely SF1, SF2, and SF3. The performance comparison of SF1, SF2, and SF3 stacking framework on the benchmark dataset using 10-fold CV is presented in Table 8. Table 8 demonstrates that SF2 outperforms both SF1 and SF3 in the case of balanced accuracy, FNR, F1-score, MCC, and ACC. Hence, we select SF2, which includes ET, LightGBM, LogReg, and KNN as base-classifiers and another LightGBM as a meta-classifier, as our final predictor.

Here we compare the performance of the proposed method *diSBPred* with an existing NNA (Niu et al., 2013) method proposed by Niu et al. For the sake of appropriate comparison, we implemented the NNA approach locally by removing the inconsistencies in the dataset proposed by Niu et al. For the details of the discrepancies, refer to the Dataset section. The performance comparison of the proposed predictor *diSBPred* with the existing NNA method is presented in Table 9.

Table 9 shows that *diSBPred* achieves an improvement of 102.85 %, 13.38 %, 43.25 %, and 22.82 % based on SN, SP, BACC, and ACC over the NNA method, respectively. Therefore, the proposed approach can predict a greater number of disulfide bonding and non-bonding pairs correctly compared to the existing state-of-the-art method. Tables 3 and 9 show that our method is superior to both the NNA-based models – NNA-Model1 that utilizes the original features proposed by Niu et al.; and NNA-Model2 that utilizes the better features proposed in this work. These results allow us to conclude that our proposed method, the *diSBPred*, significantly outperforms the current state-of-the-art method. Additionally, these outcomes help us summarize that the proposed approach can be effectively applied to annotate disulfide bonding residues of the sequence whose structure is unknown.

Moreover, the prediction from *diSBPred* can be used in the three-dimensional structure prediction of proteins to significantly reduce the conformational search space. The reduction can be achieved by imposing the geometrical constraints on the degree of freedom of the protein's backbone. Further, the predicted bonding information can be incorporated in the applied energy function to rank the structure high that matches the predicted disulfide bonding orientation.

4. Conclusions

We proposed a sequence-based predictor of disulfide bonds using a stacking based machine learning method. To train and validate the proposed approach, we collected a dataset of protein sequences, whose corresponding high-resolution structures are experimentally validated, and each structure has at least a single disulfide bond. For the accurate prediction of the disulfide bond, the computation is carried out in two stages: first, individual cysteines are predicted as either bonding or non-bonding; second, the cysteine-pairs are predicted as either bonding or non-bonding by incorporating the individual cysteine bonding prediction probability as a feature in the second tier. For the individual

Table 7
Performance of disulfide bond prediction model for different classifiers.

Metric/Methods	LightGBM	KNN	LogReg	Bag	RDF	ETC	SVM	NN
SN (%)	**79.77**	51.17	62.15	79.34	72.67	72.07	66.97	67.35
SP (%)	99.30	98.50	97.55	99.40	**99.79**	99.50	98.11	97.61
FPR	0.007	0.015	0.025	0.006	**0.002**	0.005	0.019	0.024
FNR	**0.202**	0.488	0.378	0.207	0.273	0.279	0.330	0.326
PR (%)	95.82	87.18	83.53	96.38	**98.55**	96.64	87.61	84.95
F1-score	**0.871**	0.645	0.713	0.870	0.837	0.826	0.759	0.751
MCC	0.852	0.623	0.675	**0.853**	0.822	0.808	0.727	0.715
BACC (%)	**89.54**	74.83	79.85	89.37	86.23	85.78	82.54	82.48
ACC (%)	96.05	90.61	91.65	**96.06**	95.27	94.93	92.92	92.57

The best score values are **bold**faced.

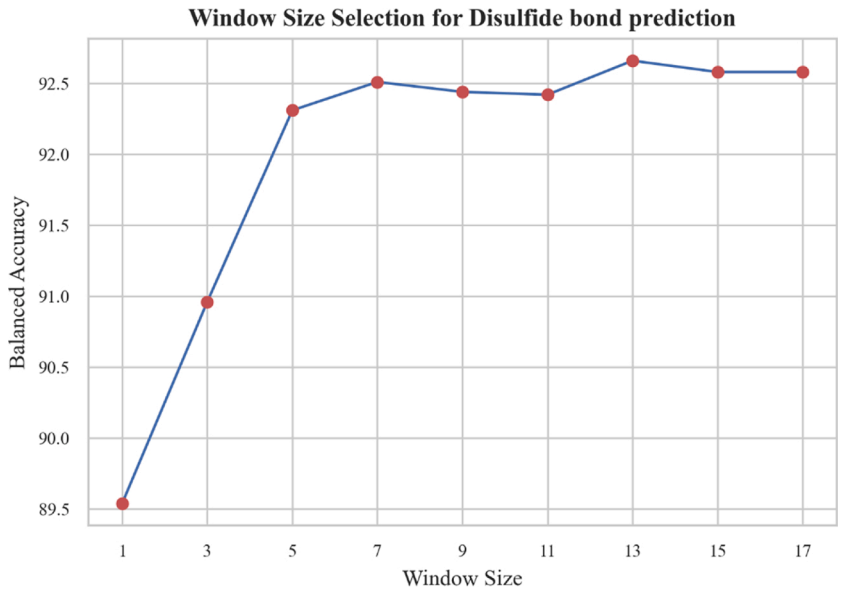

Fig. 6. Performance comparison of the LightGBM models trained using a feature vector created by applying a sliding window of different sizes. The 10-fold CV balanced accuracy of the models on the Imb_DBD dataset for various window sizes is reported.

Table 8
Comparison of different stacking framework with a different set of base-classifiers on the benchmark dataset through a 10-fold CV.

Metric/Methods	SF1	SF2	SF3
SN (%)	88.94	**89.05**	89.01
SP (%)	**99.37**	99.34	99.33
FPR	**0.006**	0.007	0.007
FNR	0.111	**0.109**	0.110
PR (%)	**96.59**	96.44	96.40
F1-score	**0.926**	**0.926**	**0.926**
MCC	**0.913**	**0.913**	0.912
BACC (%)	94.16	**94.20**	94.17
ACC (%)	94.16	**97.63**	97.61

Best score values are **bold**faced Performance comparison with the existing approach.

Table 9
Comparison of the proposed method *diSBPred* with the existing NNA method.

Metric/Methods	NNA	diSBPred	(imp. %)
SN (%)	52.60	**89.05**	(102.85 %)
SP (%)	88.63	**99.34**	(13.38 %)
BACC (%)	70.62	**94.20**	(43.25 %)
ACC (%)	81.93	**97.63**	(22.82 %)

Here, 'imp.' stands for improvement. The 'imp. %' represents an improvement in percentage achieved by *diSBPred* over the NNA. The best score values are **bold**faced.

cysteine bonding state prediction, the dataset containing all the bonding and non-bonding cysteines is used. Furthermore, for the prediction of disulfide bonds, we constructed a benchmark dataset containing bonding and non-bonding pairs in the ratio of 1:5. We used features such as amino acid residue profile, physiochemical profile, conservation profile, structural profile, flexibility profile, and energy profile for individual cysteine bonding state prediction. Moreover, the distance between each pair of cysteines is also used for the disulfide bond prediction.

The proposed approach achieved a prediction balanced accuracy of 82.29 % for individual cysteine bonding state prediction and 94.20 % for disulfide bond prediction. Also, the comparison of the proposed approach with the existing NNA based approach shows that the proposed predictor achieves an improvement of 43.25 % based on BACC (prediction balanced accuracy). These results confirm the robustness of the proposed approach. Moreover, comparative results highlight that the proposed approach significantly outperforms the existing method. Therefore, our approach can be used to effectively annotate the disulfide bonding residue of the protein sequence whose structure is unknown. Besides, the predictor can be useful in reducing the conformational search in the prediction of the three-dimensional structure of the protein by imposing geometrical constraints on the protein-backbone.

Ethical statement

NA (since, no animal or human study is involved.)

Author contributions

Conceived and designed the experiments: AM, MWUK and MTH. Performed the experiments: AM, MWUK. Analyzed the data: AM, MWUK. Contributed reagents/materials/analysis tools: MTH. Wrote the paper: AM, MWUK and MTH.

Availability

Code-data is available here http://cs.uno.edu/~tamjid/Software/diSBPred/code_data.zip

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

The authors gratefully acknowledge the Louisiana Board of Regents through the Board of Regents Support Fund LEQSF (2016-19)-RD-B-07.

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.compbiolchem.20

21.107436.

References

Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. 46, 175–185.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. J. Mol. Biol. 215 (15 May 1990), 403–410.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., et al., 2009. BLAST+: architecture and applications. BMC Bioinformatics 10.

Chaudhuri, A.R., Khan, I.A., Ludueña, R.F., 2001. Detection of disulfide donds in bovine brain tubulin and their role in protein folding and microtubule assembly in vitro: a novel disulfide detection approach. Biochemistry 40, 8834–8841.

Chen, K., Kurgan, L., 2012. Computational prediction of secondary and supersecondary structures. In: Kister, A.E. (Ed.), Protein Supersecondary Structures, vol. 932. Humana Press, Totowa, NJ. Methods in Molecular Biology (Methods and Protocols).

Cheng, J., Saigo, H., Baldi, P., 2005. Large-scale prediction of disulphide bridges using kernel methods, two-dimensional recursive neural networks, and weighted graph matching. Proteins Struct. Funct. Bioinform. 62, 617–629.

Chuang, C.C., Chen, C.Y., Yang, J.M., Lyu, P.C., Hwang, J.K., 2003. Relationship between protein structures and disulfide-bonding patterns. Proteins Struct. Funct. Bioinform. 53, 1–5.

Consortium, T.U., 2007. The universal protein resource (UniProt). Nucleic Acids Res. 35, D193–D197.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 [cs], 2019/05/24/.

Dranoff, G., 2009. Targets of protective tumor immunity. Ann. N. Y. Acad. Sci. 1174, 74–80.

Džeroski, S., Ženko, B., 2004. Is Combining Classifiers with Stacking Better than Selecting the Best One? Mach. Learn. 54, 255–273.

Faraggi, E., Zhang, T., Yang, Y., Kurgan, L., Zhou, Y., 2012. SPINE X: improving protein secondary structure prediction by multi-step learning coupled with prediction of solvent accessible surface area and backbone torsion angles. J. Comput. Chem. 33, 259–267.

Fariselli, P., Casadio, R., 2001. Prediction of disulfide connectivity in proteins. Bioinformatics 17, 957–964.

Fariselli, P., Riccobelli, P., Casadio, R., 1999. Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. Proteins Struct. Funct. Bioinform. 36, 340–346.

Fass, D., 2012. Disulfide bonding in protein biophysics. Annu. Rev. Biophys. 41, 63–79.

Ferrè, F., Clote, P., 2005. Disulfide connectivity prediction using secondary structure information and diresidue frequencies. Bioinformatics 21, 2336–2346.

Ferrè, F., Clote, P., 2006. DiANNA 1.1: an extension of the DiANNA web server for ternary cysteine classification. Nucleic Acids Res. 34, W182–W185.

Fiser, A., Simon, I., 2000. Predicting the oxidation state of cysteines by multiple sequence alignment. Bioinformatics 16, 251–256.

Fiser, A., Cserzo, M., Tudos, E., Simon, I., 1992. Different sequence environments of cysteines and half cystines in proteins. FEBS Lett. 302, 117–120.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63, 3–42.

Guo, Q., Manolopoulou, M., Bian, Y., Schilling, A.B., Tang, W.-J., 2010. Molecular basis for the recognition and cleavages of IGF-II, TGFalpha, and amylin by human insulin-degrading enzyme. J. Mol. Biol. 395, 430–443.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning, 2 ed. Springer-Verlag, New York.

Ho, T.K., 1995. Random decision forests," presented at the document analysis and recognition, 1995. Proceedings of the Third International Conference on, Montreal, Que., Canada.

Hogg, P.J., 2009. Contribution of allosteric disulfide bonds to regulation of hemostasis. J. Thromb. Haemost. 7, 13–16.

Hu, Q., Merchante, C., Stepanova, A.N., Alonso, J.M., Heber, S., 2015. A stacking-based approach to identify translated upstream Open Reading frames in Arabidopsis Thaliana. Presented at the International Symposium on Bioinformatics Research and Applications.

Huang, E.S., Samudrala, R., Ponder, J.W., 1999. Ab initio fold prediction of small helical proteins using distance geometry and knowledgebased scoring functions. J. Mol. Biol. 290, 267–281.

Iqbal, S., Hoque, M.T., 2018. PBRpredict-Suite: a suite of models to predict peptide-recognition domain residues from protein sequence. Bioinformatics pp. bty352-bty352.

Iqbal, S., Hoque, M.T., 2016. Estimation of position specific energy as a feature of protein residues from sequence alone for structural classification. PLoS One 11, e0161452.

Iqbal, S., Mishra, A., Hoque, T., 2015. Improved prediction of accessible surface area results in efficient energy function application. J. Theor. Biol. 380, 380–391.

Islam, M.N., Iqbal, S., Katebi, A.R., Hoque, M.T., 2016. A balanced secondary structure predictor. J. Theor. Biol. 389, 60–71.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al., 2017. LightGBM: a highly efficient gradient boosting decision tree. In: Presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, California, USA.

Lin, H.-H., Tseng, L.-Y., 2010. DBCP: a web server for disulfide bonding connectivity pattern prediction without the prior knowledge of the bonding state of cysteines. Nucleic Acids Res. 38, W503–W507.

Lin, H.-H., Hsu, J.-C., Chen, Y.-F., 2012. Disulfide bonding pattern prediction server based on normalized pair distance by MODELLER. In: Presented at the 2012 International Symposium on Computer, Consumer and Control. Taichung, Taiwan.

Liu, J., Tan, H., Rost, B., 2002. Loopy proteins appear conserved in evolution. J. Mol. Biol. 322, 53–64.

Márquez-Chamorro, A.E., Aguilar-Ruiz, J.S., 2015. Soft computing methods for disulfide connectivity prediction. Evol. Bioinform. 11, 223–229.

McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133, 1943/12/01.

Mishra, A., Pokhrel, P., Hoque, M.T., 2018. StackDPPred: a stacking based prediction of DNA-binding protein from sequence. Bioinformatics 35, 433–441.

Mobli, M., King, G.F., 2010. NMR methods for determining disulfide-bond. Toxicon 56, 849–854.

Mossuto, M.F., 2013. Disulfide bonding in neurodegenerative misfolding diseases. Int. J. Cell Biol. 2013.

Muskal, S.M., Holbrook, S.R., Kim, S.-H., 1990. Prediction of the disulfide-bonding state of cysteine in proteins. Protein Eng. Des. Sel. 3, 667–672.

Nagi, S., Bhattacharyya, D.K., 2013. Classification of microarray cancer data using ensemble approach. Netw. Model. Anal. Health Inform. Bioinform. 2, 159–173.

Nakamura, T., Lipton, S.A., 2009. Cell death: protein misfolding and neurodegenerative diseases. Apoptosis 14, 455–468.

Newell, A., 1969. An introduction to computational geometry. Science 165, 780.

Niu, S., Huang, T., Feng, K.-Y., He, Z., Cui, W., Gu, L., et al., 2013. Inter- and intra-chain disulfide bond prediction based on optimal feature selection. Proetin Peptide Lett. 20, 324–335.

Niu, S., Huang, T., Feng, K., Cai, Y., Li, Y., 2010. Prediction of tyrosine sulfation with mRMR feature selection and analysis. J. Proteome Res. 9, 6490–6497.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al., 2012. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12.

Rubinstein, R., Fiser, A., 2008. Predicting disulfide bond connectivity in proteins by correlated mutations analysis. Bioinformatics (Oxford, England) 24, 498–504, 2008/02/15/.

Senior, A.W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., et al., 2019. Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). Proteins Struct. Funct. Bioinform. 87, 1141–1148, 2019/12/01.

Sharma, A., Dehzangi, A., Lyons, J., Imoto, S., Miyano, S., Nakai, K., et al., 2014. Evaluation of sequence features from intrinsically disordered regions for the estimation of protein function. PLoS One 9 (2), e89890.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs], 2015/04/10/.

Song, J., Yuan, Z., Tan, H., Huber, T., Burrage, K., 2007. Predicting disulfide connectivity from protein sequence using multiple sequence feature vectors and secondary structure. Bioinformatics 23, 3147–3154.

Sun, Y., Smith, D.L., 1988. Identification of disulfide-containing peptides by performic acid oxidation and mass spectrometry. Anal. Biochem. 172, 130–138.

Sutton, K.A., Black, P.J., Mercer, K.R., Garman, E.F., Owen, R.L., Snell, E.H., et al., 2013. Insights into the mechanism of X-ray-induced disulfide-bond cleavage in lysozyme crystals based on EPR, optical absorption and X-ray diffraction studies. Acta Crystallogr. D Biol. Crystallogr. 69, 2381–2394.

Szilágyi, A., Skolnick, J., 2006. Efficient prediction of nucleic acid binding function from low-resolution protein structures. J. Mol. Biol. 358, 922–933.

Tarafder, S., Ahmed, M.T., Iqbal, S., Hoque, M.T., Rahman, M.S., 2018. RBSURFpred: modeling protein accessible surface area in real and binary space using regularized and optimized regression. J. Theor. Biol. 441, 44–57.

Tompa, P., 2002. Intrinsically unstructured proteins. Trends Biol. Sci. 27, 527–533.

Tsai, C.-H., Chen, B.-J., Chan, C.-h., Liu, H.-L., Kao, C.-Y., 2005. Improving disulfide connectivity prediction with sequential distance between oxidized cysteines. Bioinformatics 21, 4416–4419.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer-Verlag.

Vincent, M., Passerini, A., Labbé, M., Frasconi, P., 2008. A simplified approach to disulfide connectivity prediction from protein sequences. BMC Bioinformatics 9.

Vullo, A., Frasconi, P., 2004. Disulfide connectivity prediction using recursive neural networks and evolutionary information. Bioinformatics 20, 653–659.

Wess, J., Han, S.-J., Kim, S.-K., Jacobson, K.A., Li, J.H., 2008. Conformational changes involved in G-protein-coupled-receptor activation. Trends Pharmacol. Sci. 29, 616–625.

Wolpert, D.H., 1992. Stacked generalization. Neural Netw. 5, 241–259.

Wright, P.E., Dyson, H.J., 1999. Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm. J. Mol. Biol. 293, 321–331.

Yang, J., He, B.-J., Jang, R., Zhang, Y., Shen, H.-B., 2015. Accurate disulfide-bonding network predictions improve ab initio structure prediction of cysteine-rich proteins. Bioinformatics 31, 3773–3781.

Zhang, T., Faraggi, E., Xue, B., Dunker, A.K., Uversky, V.N., Zhou, Y., 2012. SPINE-D: accurate prediction of short and long disordered regions by a single neural-network based method. J. Biomol. Struct. Dyn. 29, 799–813.

Zhang, T., Faraggi, E., Zhou, Y., 2010. Fluctuations of backbone torsion angles obtained from NMR-determined structures and their prediction. Proteins Struct. Funct. Bioinform. 78, 3353–3362.

Zhu, L., Yang, J., Song, J.N., Chou, K.C., Shen, H.B., 2010. Improving the accuracy of predicting disulfide connectivity by feature selection. Comput. Chem. 31, 1478–1485.