Examination of Different Representations of Proteins Using Protein Ray-based Descriptor and Deep Learning Models

G. Mirceva, A. Naumoski and A. Kulakov
Faculty of computer science and engineering, Ss. Cyril and Methodius University in Skopje, Skopje, Macedonia
georgina.mirceva@finki.ukim.mk, andreja.naumoski@finki.ukim.mk, andrea.kulakov@finki.ukim.mk

***Abstract* - The study of proteins has been of high importance because it is needed to understand the processes in the living organisms in which these molecules are involved. Proteomics is the research area that studies the protein structures. One of the tasks on which proteomics is focused on is solving the protein classification task. Although there are many studies focused on this problem, it is still a popular task because there is still need for faster methods for protein classification. The aim of the study presented in this paper is to develop a fast and accurate protein classification model. For that purpose, for feature extraction we use our protein ray-based descriptor. We use a deep learning architecture for generating prediction model. Besides the standard form of the protein ray-based descriptor, we also consider several other representations of the proteins and make examination which is the most appropriate representation. Some experimental results are given and discussed.**

Keywords - protein structure; protein classification; feature extraction; deep learning

I. INTRODUCTION

Proteomics focuses on the study of protein molecules, which are among the most vital compounds present in the human body. They participate in a variety of cellular processes, making them indispensable for various biological functions. Proteins serve critical roles in biochemical reactions as enzymes, also play a role in the transport of oxygen to cells, have signaling role such as insulin, and serve as antibodies thus providing a defensive mechanism. Due to their importance in the functioning of organisms, there is a significant interest in the proteomics community to understand the structure and functions of proteins. By studying the structure and properties of proteins, researchers can gain valuable insights into their functional roles in different cellular processes.

The development of technology has provided various techniques to determine the structure of protein molecules. These techniques enable the determination of protein structures, which are subsequently deposited in the Protein Data Bank (PDB) [1], [2], serving as the primary repository for this purpose. Protein structures are stored in PDB files, which provide information about the primary, secondary, and tertiary structures of proteins. Despite the vast amount of data available on protein structures, there remains a significant need for research to discover the functional roles of proteins in living organisms.

There are various approaches for determining the functions of proteins. Some approaches rely on the assumption that proteins belonging to the same class share similar functions. Therefore, the protein classification task is of high importance in this regard. Numerous methods have been introduced for classifying protein structures. However, despite these efforts, there is still a significant gap between the number of known protein structures and the number of proteins that are functionally annotated. This highlights the high need for computational methods for fast and accurate classification of proteins.

The Structural Classification Of Proteins (SCOP) [3] is widely recognized as a very important method for protein classification. This method is highly accurate, as it involves visual inspections by human experts in order to make a decision. However, this process is not very fast, which highlights the need for automatic or semi-automatic methods. One such method is the Class, Architecture, Topology and Homologous superfamily (CATH) [4]. CATH uses a semi-automatic approach because it considers manual decision for a particular protein only if the automatic classification is not appropriate.

Also, there are methods that utilize protein sequence alignment to solve the protein classification task. For that purpose, methods like Needleman–Wunch [5], BLAST [6] and PSI-BLAST [7] could be applied. Nevertheless, these methods may not be suitable for proteins with similar structures whose sequences are not so similar. Therefore, it is preferable to align protein structures rather than sequences using methods such as CE [8], MAMMOTH [9] and DALI [10]. Additionally, some methods [11], [12] combine sequence alignment and structure alignment for improved accuracy.

Methods based on feature vector comparison offer a solution to the problem of long classification times using alignment. These methods extract the most relevant characteristics of protein sequences or structures and use them to create a prediction model. Feature vectors can be extracted from protein sequences [13] or structures [14], containing the most informative features for the analysis. The use of these feature vectors reduces the amount of data that's been processed, speeding up the creation of the classification model and reducing the time required for

testing for novel proteins. Different machine-learning algorithms can be used to create the classification model from the extracted feature vectors.

The goal of this research is to develop a fast and accurate method for protein classification. Our attention is on the last category of methods that utilize feature vectors for representing the key characteristics of protein structures. Our previous studies have concentrated on the extraction of feature vectors for protein tertiary structures. In [15], we conducted a comparative analysis of different methods for comparing protein structures. These methods are for solving protein retrieval task, where for a given target protein the most similar proteins from the database are identified. They can also be employed for solving the task of protein classification based on the extracted feature vectors. In this paper, we present a method for solving protein classification task. Our approach involves extracting feature vectors, and to accomplish this, we have selected the protein ray-based descriptor as it has demonstrated high accuracy, compactness, and ease of extraction in our previous research [15]. This descriptor captures information about the geometrical characteristics of the protein structure, specifically how the protein backbone is oriented in space relative to the center of mass. Once the feature vectors have been extracted, we then employ a deep learning architecture to generate classification models. Besides the standard form (variant) of the protein ray-based descriptor, additionally we employed three other variants of the protein ray-based descriptor that we previously used in [16], where several well-known classification methods were used for protein classification.

The rest of this paper is organized as follows. Section 2 provides an overview of the method that is used in this research. We outline the process for extracting the protein ray-based descriptor and present the deep learning architecture used for generating prediction models. We also present the four variants of the protein ray-based descriptor that are considered in this study. In Section 3, we present the experimental results obtained using various neural network settings and we discuss the effect of these settings. Section 4 gives conclusion of the paper and suggests future directions for enhancing this research.

II. The Method Used in This Study

In this paper, we present a method that solves the task for classifying proteins by considering only their tertiary structure. The method comprises two main steps. First, we extract feature vectors by employing our protein ray-based descriptor [15]. Besides its standard form, we also use three additional variants that are obtained from the standard form of the protein ray-based descriptor. Subsequently, we employ deep learning architecture to create a classification model for classification of new proteins. For that purpose, we employ fully connected neural network by using several hidden layers.

The training phase elucidates the process of generating the prediction model. In the testing phase, for a particular test (query) protein the respective class is determined. Both training and testing protein data are preserved in the corresponding PDB files, which are taken from the PDB database [2]. These files contain information regarding the primary, secondary, and tertiary structure of the proteins. In this study, the focus is on the tertiary structure of proteins, thus we inspect the proteins' geometry.

The training phase begins with the extraction of feature vectors for the training proteins. These feature vectors are utilized as samples in the process that is employed in the succeeding stage where the model is generated. Next, the model is created by using deep learning architecture where the weights in the neural network model are adjusted.

Upon creation of the model, it is possible to classify novel proteins. For the query protein, the feature vector is extracted in the same manner as the training proteins. The extracted descriptor is then presented to the model that generates corresponding output (class decision) for that protein. In this research, the classes relate to the SCOP domains used by the SCOP method. We consider only the domain level from the SCOP hierarchy as a level enabling the distinction between the proteins based on their functions. According to this, the query protein is classified into a corresponding SCOP domain.

A. Protein Ray-based Descriptor

Proteins are made of multiple chains, which are folded into specific 3D compositions. The SCOP database [3] provides information about the SCOP domains for the protein chains, so the protein chains are the samples within the dataset that is used. Each protein chain is comprised of amino acid residues, which are connected to form the protein backbone. The amino acid residues within each chain are folded in a particular way. The amino acid residues within each chain consist of various atoms. The Cα atoms within the amino acids connect two consecutive amino acid residues and form the protein backbone.

In a previous study, as documented in [15], we explored various methods for protein structure retrieval. While some of these methods take into account all atoms, the others are focused solely on the Cα atoms. As per the findings presented in [15], the accuracy of the protein structure retrieval process was observed to decline with the inclusion of the remaining atoms. It was concluded that considering only the Cα atoms is the more suitable approach. The protein ray-based descriptor was applied to extract information concerning the 3D coordinates of the Cα atoms, and as a result, the confirmation of the protein backbone is presented. These findings facilitated the creation of a 3D model for the protein, where the protein backbone was defined as a 3D object that occupies a position in the 3D space.

The 3D model created for the protein is scaled ensuring that the Euclidean distance between the most distant Cα atom and the center of mass is 1. With that, we provide scale invariance of the feature vectors. In 3D object retrieval, it is important to provide not only scale invariance but also invariance to translation and rotation. This implies that the same feature vector should be extracted for a given protein chain, regardless of any translations or rotations that are performed. The approach

by which this descriptor is extracted guarantees these properties.

Another challenge in protein structure retrieval is representing protein chains with feature vectors of the same length, considering their differing numbers of Cα atoms. To address this, the protein backbone is interpolated with a fixed number of interpolation points, ensuring that the same number of interpolation points is utilized for each protein chain, regardless of the number of Cα atoms. In [15], we employed two approaches for interpolating the protein backbone. The first approach entailed uniformly interpolating the backbone with interpolation points equidistantly spaced along it, while the second approach involved using more interpolation points in parts of the backbone where consecutive Cα atoms are spaced farther apart. The findings reported in [15] indicate that the feature vectors extracted using uniform interpolation are more accurate. Hence, in this study, we utilize the uniform interpolation of the protein backbone.

The uniform interpolation of the backbone of a given protein involves the initial step of determining the length of the backbone, which is achieved by applying Eq. (1).

$$L = \sum_{i=1}^{N_\alpha - 1} d_{Euclidean}(i, i+1) \tag{1}$$

Here, $d_{Euclidean}(i,i+1)$ represents the Euclidean distance between the i-th and (i+1)-th Cα atoms, and N_α denotes the total number of Cα atoms in the protein chain being analyzed.

In our previous work presented in [15], we conducted an analysis to identify the optimal number of interpolation points for uniform interpolation of the protein backbone. The findings indicated that the most effective number of interpolation points is 64. It was observed that increasing the number of interpolation points above this value did not significantly enhance retrieval accuracy, while using a lower number of interpolation points resulted in a notable decrease in performance. Consequently, in this paper, we have employed 64 interpolation points for uniform interpolation. The interpolation points are uniformly spaced along the curve of the protein backbone, with a distance between two consecutive points equal to $L/(N\text{-}1)$, where L denotes the length of the backbone calculated using Eq. (1), while N denotes the number of interpolation points (64 in this study).

Upon determining the interpolation points, the subsequent step involves extracting the feature vectors. The feature vector, as its name implies, draws inspiration from the ray descriptor [17], which was initially proposed for 3D objects retrieval. The name indicates that rays are "emitted" from the center of mass towards the points that represent the object, which in our case are the interpolation points. The features are obtained by calculating the Euclidean distances between the center of mass and the points that represents the object. This way, the feature vector provides invariance to translation and rotation. The protein ray-based descriptor obtained for a given protein chain elucidates how the backbone of the inspected protein chain is positioned in the space relative to the center of mass.

B. Four Approaches for Extraction of the Protein Ray-Based Descriptor

In the standard form of the protein ray-based descriptor described above, each element of the feature vector pertains to an individual interpolation point. In essence, the protein ray-based descriptor aims to elucidate the traversal of the protein backbone from one concentric sphere to another if we divide the 3D space with concentric spheres. In [18], it is illustrated visually how the protein backbone traverses between these concentric spheres as we go along the backbone.

In [16], we utilized three additional variants of the feature vector obtained based on the Euclidean distances for the interpolation points. In the standard form (standard variant) of the protein ray-based descriptor [15], the feature vector $f_{Eucl} = [f_1, f_2, \ldots, f_N]$ contains the Euclidean distances from the interpolation points towards the center of mass. For example, the feature for the i-th interpolation is $f_i = D_i$, for i=1, 2,…, N, where D_i is the Euclidean distance from the i-th interpolation point to the center of mass. In the other three variant, we consider pairs of consecutive interpolation points. With the second variant, we calculate the difference between the Euclidean distances for two consecutive interpolation points, so the i-th feature is calculated as $diff_i = f_i - f_{i+1} = D_i - D_{i+1}$, i=1, 2,…, N–1. The remaining two variant considers only the magnitude of this change (difference), or only the sign of this change. In the third variant, the i-th feature is calculated as $abs_i = |diff_i|$, i=1, 2,…, N–1. The fourth variant examines whether backbone goes towards the surface or towards the center of mass as we traverse along the backbone without reflecting the quantity of the increase or decrease of the Euclidean distance. With this variant, the i-th feature is calculated as $sign_i = sign(diff_i)$, i=1, 2,…, N–1, where the function $sign(x)$ returns 1 for $x > 0$, 0 for $x = 0$ and -1 for $x < 0$.

C. Deep Learning Models

This study is concerned with the development of classification models through the application of deep learning. Specifically, a fully connected neural network was utilized. The hidden layers are dense layers, wherein each neuron is interconnected with the neurons from the preceding and succeeding layers. This architecture enables a heightened degree of connectivity, resulting in a more intricate representation of the input data, and thereby leading to enhanced performance of the neural network.

The input layer of our neural network model consists of 64 neurons, corresponding to the length of the feature vectors. Each neuron in the hidden layers is activated using the rectified linear unit (ReLU) activation function. The output layer is composed of 150 neurons, with each neuron corresponding to one of the classes, namely SCOP domains in this context. The softmax activation function is employed in the output layer. The optimization algorithm used in this study is stochastic gradient descent (SGD), which is utilized for optimizing the objective

function. Moreover, SGD is also employed as a bias updater. The Adam optimizer is employed as an updater in this study and the learning rate equals 0.001.

In [19], we used the same deep learning architecture to build prediction models. However, in this study besides the standard form of the protein ray-based descriptor we also use the three remaining variants of this descriptor that were described before.

Based on the results obtained in [19], we used the best settings for the number of hidden layers and number of neurons per hidden layer. In the experiments we used 3 hidden layers with 100 neurons per layer. We trained the models for 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 epochs. In this way we wanted to find out if the models are underfitted or overfitted.

WekaDeeplearning4j package [20] was used in this study that utilizes the Deeplearning4j Java library. We used the default setting for the rest of the parameters.

III. Results and Discusion

To evaluate the efficacy of the approach used in this study, we employed data from the PDB database [2] regarding the confirmation of the protein backbone presented by the 3D coordinates data of the Cα atoms. As for the class labels, we utilized knowledge from the SCOP database [3]. It is worth noting that, in this study, we considered only the domain level, thereby establishing a correspondence between the domains of the analyzed protein chains and the classes to which they would be assigned.

The dataset used in this study comprises 6145 protein chains, evenly distributed across 150 SCOP domains, which serve as the potential classes for classification. To evaluate the proposed method, this dataset is split into training and testing sets where 90% of the samples are training data and 10% are test data, ensuring that the uniform distribution of the classes is maintained in both subsets. The resulting training set consists of 5531 protein chains, while the remaining 614 protein chains are used for testing the accuracy of the classification models.

In this section, we will present the results regarding the classification accuracy achieved with the classification models. Given that the test set is balanced, classification accuracy is suitable to measure the predictive performance of the models.

The results given in Table 1 present the classification accuracy of the models. The results revealed that by using the original form of the protein ray-based descriptor, as the number of epochs is increased, the accuracy also increases slightly, but when training for more than 80 epochs, the classification accuracy decreases that indicates that the model is overfitted. The best model is obtained when training the network for 80 epochs. However, with the other three variants, the same pattern is not shown. With the remaining three variants, the best model is obtained when training for 80, 100 and 20 epochs respectively. The best models obtained with the second and third variant achieved the same value for classification accuracy. The fourth variant has slightly lower accuracy (around 1% lower) due to the fact that only the sign of the change (difference) is considered. Although the original version of the protein ray-based descriptor outperforms the other models for almost 0.5% and 1.5%, it encapsulates more info, although the descriptor is longer just for one element. When comparing the second, third and fourth variant, we can note that using more info (magnitude and sign of the change with the second variant), the model is tuned for less epochs (80 in this case for the best model), while when the sign is not considered (with the third variant) a little bit more training is needed (the best model is obtained for 100 epochs). The fourth variant holds least info (only the sign of the change), but it still can provide accurate retrieval comparable to the other variants. Since the dataset is simpler in this case, there is a need for lower number of epochs. The best model in this case (with the fourth variant) is obtained for 20 epochs, and by training further, the model is overfitted, and more significant drop in the accuracy is observed.

In our earlier research [15], we conducted an analysis in which various approaches for comparing proteins were assessed in respect with DALI and CE methods. The results of this analysis demonstrated that, despite its simplicity and speed, the protein ray-based descriptor yields precise predictions of similar proteins and is competitive with time-intensive state-of-the-art methods.

In our study, we conducted an analysis to compare the performance of the models created in this research with those from our previous study [16], which utilized different classification methods and the protein ray-based descriptor. The results of this comparison are presented in Table 2. Our analysis revealed that the best model obtained in this study outperforms the majority of the models obtained in [16], with the exception of knn. However, it is worth noting that the testing time with knn is higher. If larger dataset is used, the difference in testing

TABLE I. The Results Obtained Using Different Number of Epochs

Number of epochs	Euclidean distance	Diff	Abs diff	Sing diff
10	97.07	95.77	95.60	94.63
20	97.39	96.91	94.79	**96.25**
30	97.23	97.07	95.77	95.44
40	97.23	96.91	96.42	95.77
50	97.23	96.42	95.77	95.60
60	97.23	96.74	96.91	94.79
70	97.56	95.60	96.42	94.79
80	**97.72**	**97.23**	96.58	96.09
90	97.56	96.74	96.42	95.77
100	96.91	97.07	**97.23**	94.95

time would be even more significant, as knn is an instance-based learning classifier where model is not created, rather it makes decision based on the similarity between the query and the training samples.

IV. CONCLUSION

In this research paper, we presented a two-steps approach that could be used to classify novel protein structures utilizing the data regarding their tertiary structure. Firstly, we extract the protein ray-based descriptors for the training protein chains. Besides the standard form of the protein ray-based descriptor, we also used three other variants where instead of the Euclidean distances between the interpolation points and center of mass, the change of the Euclidean distance was considered. Particularly, we analyzed the difference between the Euclidean distances for two consecutive interpolation points, as well as the absolute value and the sign of this difference. In this way, we examined what is the best variant to represent the confirmation of the protein backbone in the 3D space. After extracting the feature vectors, then the second step utilizes a deep learning architecture to create classification models. We utilized fully connected neural network models with 3 hidden layers, each containing 100 neurons, as these settings showed best performance in our former study. The evaluation is based on knowledge obtained from the SCOP database. As evaluation measure, the classification accuracy was used as it is appropriate when using balanced dataset. As standard of truth, knowledge about the belonging to SCOP domains from the SCOP hierarchy was used.

In this study we performed experiments by training the model using different setting for the number of epochs. The best model is obtained with the first variant (the original form of the protein ray-based descriptor) when training the network for 80 epochs. The accuracy increased by increasing the number of epochs up to 80, while when training further the model is overfitted. The other three variants showed other patterns, and they lead to the best model for 80, 100 and 20 epochs respectively. The accuracy with the fourth variant is slightly lower (around 1% lower) since it considers only whether the backbone goes towards the surface or center of mass (presented with the sign of the change). The first variant (the original version of the descriptor) slightly outperforms the others since it considers more data. It is interesting to note that this variant is longer just for one element compared to the others. When comparing the three other variants we can make conclusion that using both the magnitude and sign of the change (the second variant), the model is fine-tuned earlier (for less epochs) compared with the case when the sign is not considered (third variant). The fourth variant considers info only for the sign of the change but is comparable to the other variants. With this variant, the data are simpler, so less training is needed, leading to best performance when training the model for 20 epochs. If the model is trained further, the model becomes overfitted. Although this variant has lowest accuracy (less than 1.5% lower compared to the first variant), it is comparable to the others, and especially it is worth due to the faster training of the network. This would be more important if larger dataset is used for training where more time would be needed for training, so for less epochs an appropriate model would be obtained using this variant.

TABLE II. THE RESULTS OBTAINED BY UTILIZING DIFFERENT CLASSIFICATION METHODS

Classification model	**Classification accuracy**
This study	97.720
C4.5	92.997
Naïve Bayes	94.625
Bayes Net	96.417
knn	98.534
SVM	97.557

As regards potential improvements, there exist various possible directions that could be taken. Aside from the protein ray-based descriptor, other feature vectors could also be employed. In relation to the neural network model, other parameters such as activation function and optimization algorithm could be assessed. Also, the other settings could be examined in order to create more accurate model. Additionally, beyond the fully connected neural network, other deep learning architectures could be explored. It is also intended to apply deep learning architectures that exploits the fuzzy logic. Besides deep learning architectures, also other approaches based on other classification methods could be employed, taking into consideration algorithms that are established on classical sets and fuzzy sets.

ACKNOWLEDGMENT

This work was partially financed by the Faculty of computer science and engineering at the “Ss. Cyril and Methodius University in Skopje”, Skopje, Macedonia.

REFERENCES

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The protein data bank,” Nucleic Acids Res., vol. 28, no. 1, pp. 235–242, January 2000.

[2] RCSB Protein Data Bank, http://www.rcsb.org, 2019.

[3] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, “Scop: a structural classification of proteins database for the investigation of sequences and structures,” J. Mol. Biol., vol. 247, no. 4, pp. 536–540, April 1995.

[4] C. A. Orengo, A. D. Michie, D. T. Jones, M. B. Swindells, and J. M. Thornton, “CATH – a hierarchic classification of protein domain structures,” Structure, vol. 5, no. 8, pp. 1093–1108, August 1997.

[5] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” J. Mol. Biol., vol. 48, no. 3, pp. 443–453, March 1970.

[6] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” J. Mol. Biol., vol. 215, no. 3, pp. 403–410, October 1990.

[7] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,” Nucleic Acids Res., vol. 25, no. 17, pp. 3389–3402, September 1997.

[8] I. N. Shindyalov and P. E. Bourne, “Protein structure alignment by incremental combinatorial extension (CE) of the optimal path,” Protein Eng., vol. 11, no. 9, pp. 739–747, September 1998.

[9] A. R. Ortiz, C. E. Strauss, and O. Olmea, “Mammoth: an automated method for model comparison,” Protein Sci., vol. 11, no. 11, pp. 2606–2621, November 2002.

[10] L. Holm and C. Sander, “Protein structure comparison by alignment of distance matrices,” J. Mol. Biol., vol. 233, no. 1, pp. 123–138, September 1993.

[11] S. Cheek, Y. Qi, S. S. Krishna, L. N. Kinch, and N. V. Grishin, “SCOPmap: automated assignment of protein structures to evolutionary superfamilies,” BMC Bioinformatics, vol. 5, pp. 197–221, December 2004.

[12] C. H. Tung and J. M. Yang, “fastSCOP: a fast web server for recognizing protein structural domains and SCOP superfamilies,” Nucleic Acids Res., vol. 35, W438–W443, July 2007.

[13] K. Marsolo, S. Parthasarathy, and C. Ding, “A multi-level approach to SCOP fold recognition,” IEEE Symposium on Bioinformatics and Bioeng., pp. 57–64, October 2005.

[14] P. H. Chi, Efficient protein tertiary structure retrievals and classifications using content based comparison algorithms, PhD thesis, University of Missouri-Columbia, 2007.

[15] G. Mirceva, I. Cingovska, Z. Dimov, and D. Davcev, “Efficient approaches for retrieving protein tertiary structures,” IEEE/ACM Trans. Comput. Biol. Bioinform., vol. 9, no. 4, pp. 1166–1179, July/August 2012.

[16] G. Mirceva and A. Kulakov, “Protein classification by using four approaches for extraction of the protein ray-based descriptor,” CIIT 2020, Macedonia, 2020.

[17] D. V. Vranic, 3D Model Retrieval, Ph.D. Thesis, University of Leipzig, 2004.

[18] G. Mirceva, M. Mirchev, and D. Davcev, “Hidden Markov Models for classifying protein secondary and tertiary structures,” Journal of Convergence, vol. 1, no. 1, pp. 57–64, 2010.

[19] G. Mirceva, A. Naumoski, and A. Kulakov, “Classification of Protein Structures Using Deep Learning Models,” 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO 2022), pp. 991-996, Opatija, Croatia, 2022.

[20] S. Lang, F. Bravo-Marquez, C. Beckham, M. Hall, and E. Frank, “WekaDeeplearning4j: a Deep Learning Package for Weka based on DeepLearning4j,” Knowledge-Based Systems, vol. 178, no. 15, pp. 48-50, August 2019.