Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 370 – 386
© TÜBİTAK
doi:10.3906/elk-2003-116

A novel Fibonacci hash method for protein family identification by using recurrent neural networks

Talha Burak ALAKUŞ[1,*], **İbrahim TÜRKOĞLU**[2]

[1]Department of Software Engineering, Faculty of Engineering, Kırklareli University, Kırklareli, Turkey
[2]Department of Software Engineering, Faculty of Technology, Fırat University, Elazığ, Turkey

Received: 20.03.2020 • **Accepted/Published Online:** 10.09.2020 • **Final Version:** 27.01.2021

Abstract: Identification and classification of protein families are one of the most significant problem in bioinformatics and protein studies. It is essential to specify the family of a protein since proteins are highly used in smart drug therapies, protein functions, and, in some cases, phylogenetic trees. Some sequencing techniques provide researchers to identify the biological similarities of protein families and functions. Yet, determining these families with sequencing applications requires huge amount of time. Thus, a computer and artificial intelligence based classification system is needed to save time and avoid complexity in protein classification process. In order to designate the protein families with computer-aided systems, protein sequences need to be converted to the numerical representations. In this paper, we provide a novel protein mapping method based on Fibonacci numbers and hashing table (FIBHASH). Each amino acid code is assigned to the Fibonacci numbers based on integer representations respectively. Later, these amino acid codes are inserted a hashing table with the size of 20 to be classified with recurrent neural networks. To determine the performance of the proposed mapping method, we used accuracy, f1-score, recall, precision, and AUC evaluation criteria. In addition, the results of evaluation metrics with other protein mapping techniques including EIIP, hydrophobicity, CPNR, Atchley factors, BLOSUM62, PAM250, binary one-hot encoding, and randomly encoded representations are compared. The proposed method showed a promising result with an accuracy of 92.77%, and 0.98 AUC score.

Key words: Protein family classification, recurrent neural networks, protein mapping, artificial intelligence

1. Introduction

Specification of protein residues to distinguish the protein families is an important research area for the bioinformatics studies like bio-molecular recognition, bioinformatics, smart drug therapy, etc. During the specification process, protein families are defined as a group, which share the similar functions. Protein family includes number of proteins, which displays the analogue structures. However, determining a protein family based on the structural information is one of the main problems in bioinformatics studies since it is difficult to obtain uncharacterized proteins [1, 2]. Deprivation of knowledge of functions about protein sequences based on structure information led to the researches to study primary sequences of proteins to identify the protein families [3–5]. Also, using a 3D structure of protein to determine the protein functions is onerous, not time efficient, and requires some complex methods like X-ray and NMR spectroscopy [6]. This caused the studies only use primary sequences of the proteins. Feature engineering is needed to analyze the primary sequences of proteins to obtain continuous and discrete features. After feature engineering, deep learning and machine

[*]Correspondence: talhaburakalakus@klu.edu.tr

learning algorithms can be applied to the extracted features to perform protein family classification. Yet, in order to apply these methods, sequences are needed to be converted to numerical representations since there is no such a method to perform artificial intelligence with raw protein sequences [7, 8].

In the literature, there are limited methods for converting protein sequences into numbers. In general, BLOSUM62 (**BLO**cks **SU**bstitution **M**atrix), PAM25 (point accepted mutation), hydrophobicity, EIIP (electron-ion interaction potential) are applied and the performance of family classification is highly dependent on the conversion method [7, 9]. Recently, deep learning models are actively used in bioinformatics studies and show promising results. CNNs (convolutional neural networks) are generally applied for image classification yet it has also been used for processing sequence data, such as protein function classification [10], protein family identification [11], and RNA/DNA binding sites [12]. Yet, recent studies indicate that recurrent neural networks work better than CNNs and have better deep learning in classifying time series [13–15]. Compared to the CNNs, recurrent neural networks may possibly improve the performance of classification of protein families by forgeting gate structure, remembering past and future information, and having a good prediction performance in given time lags of unknown duration. Motivated by this fact and the limitation of protein conversion methods, we propose a novel protein mapping method for protein family classification based on recurrent neural networks. In this paper, we apply recurrent neural networks based architecture on primary protein sequences to discriminate protein families. Firstly, each amino acid was converted to numerical representations with CPNR (complex prime number representation), EIIP, hydrophobicity, binary one-hot, Atchley factors, PAM250, BLOSUM62, randomly encoded by FIBHASH (the proposed conversion method) and were represented as a vector. Later, we fed these mapped protein sequences as an input for both recurrent neural networks. In conclusion, we compared the accuracy, recall, f1-score, precision, and AUC scores of the proposed conversion method with various mapped methods to determine the performance of the proposed method. Main contributions of this paper can be summarized as follows: We propose a novel protein mapping method to encode amino acid sequences to the numerical representations, and we only apply the primary amino acid sequences for prediction of protein families.

The rest of the part of this paper are organized as follows. In Section 2, we discuss the related works about protein family classification studies. Section 3 provides the data used in this work and background details of deep learning architecture of recurrent neural networks. Conversion of amino acid sequences to the numbers with the proposed method is also provided in this section. Section 4, depicts the application results of the paper. In Section 5, we discuss the results and provide potential usage areas of the proposed method.

2. Related work

In this section, we provide various studies about protein family classification with deep learning models. We apply deep learning in this paper rather than machine learning since machine learning methods require obtaining features manually, makes the process more complicated and they are not time-efficient. Thus, we perform deep learning and only focus on the deep learning models in this section. In the study of [16], authors provided deep learning models for protein family classification. In the study, four different deep learning approaches were applied and cross-validation scores were evaluated. Protein sequences were converted to the numerical representations with n-gram and Keras embedding methods. Data were collected from Swiss-Prot database and a total of 84,753 protein sequences were used. Authors applied DNN (deep neural network), RNN (recurrent neural network), LSTM (long-short term memory), and CNN (convolutional neural network). Best classification accuracy was observed from LSTM Keras embedding with 91.24%. Authors in [17] proposed neural networks to classify the protein families. Data were collected from UniProt database and totally 550,960 protein sequences were analyzed from 10,345 different protein families. Sequences were encoded to the numbers with GloVe (global vector) representations. In the study, GRU (gated recurrent neural networks), LSTM, biLSTM (Bidirectional LSTM), and CNN were applied. The performance of the neural networks was determined with F1 score and the best score was obtained from GRU with 94.84%. Also, in the study, authors applied one of the machine learning SVM (support vector machine) and compared the results with deep learning models. All of the deep learning models outnumbered the SVM. It showed that using a deep learning model is more effective than machine learning model. In the study of [6], authors applied RNN, LSTM, and GRU deep learning models to classify the protein families. Authors considered primary protein sequences and protein family information about 40,433 proteins was gathered from Swiss-Prot database with 30 different protein families. Like in the study of [16], protein sequences were converted to the numbers with n-gram and Keras embedding methods. Best performance was obtained from LSTM deep learning model with 78.4% accuracy. In addition to these, deep learning modes were compared with some of the machine learning algorithms such as Naive Bayes, kNN (k Nearest Neighbor), decision tree, SVM, and logistic regression. The overall performance of deep learning models was better than the traditional machine learning models. Authors in [18] provided the GCN (Graph Convolutional Network) to classify the protein families. In the study, data were obtained from HPRD (Human Protein Interaction Dataset) and totally 22,725 protein sequences were considered. They applied only GCN however they changed the layer of GCN to find the best classification performance with different number of samples. The performance of the proposed method was evaluated with accuracy score and the best result was collected from 2-layer GCN with 74.33% accuracy. Also, the classification performance was observed with t-SNE (t-Distributed Stochastic Neighbor Embedding) visualization method. In the study of [19], authors developed a protein family classification model based on second-order RNNs. In the aforementioned paper, data were obtained from Swiss-Prot database and totally 873 globin sequences were aforehanded. Authors applied second order RNN with 3 different number of epochs to determine the best classification result. Performance was determined with specificity, sensitivity, and accuracy and the best results were obtained from number of 33 epochs with 95.2% accuracy.

3. Data and methods

3.1. Protein data

We obtained protein family information of about 97,576 protein sequences in UniProt database, which consists of 60 different families. It is a freely accessible database, which includes protein sequences and their functional information. The data set consists of proteins from many bioinformatics projects including a large amount of information about the biological functions of proteins and their interactions and families. We only used the reviewed protein sequences in this work. Table 1 specifies the protein families and their number of sequences.

3.2. Protein encoding modules

Firstly, in this study, we need to encode protein sequences to numerical representations. To do that, 9 different protein mapping methods are applied. These methods can be defined as follows:

- **CPNR:** This encoding method was developed by the authors in [20] to determine the protein function comparison. In this method, amino acid codes were divided into the number of codons and each amino acid code was assigned to a prime number. Amino acids and their prime number representations can be

Table 1. Protein families used in this work.

Protein family	No. of seqs.	Protein family	No. of seqs.
Class-II aminoacyl-tRNA synthetase	7335	Universal ribosomal protein uS3	993
ABC transporter	5707	Krueppel C2H2-type zinc-finger protein	990
Class-I aminoacyl-tRNA synthetase	5656	LDH	981
G-protein coupled receptor 1	5245	MDH	981
Class I-like SAM-binding methyltransferase	4572	mitochondrial release factor	960
Protein kinase	4480	Enolase	915
TRAFAC class translation factor GTPase	4179	Universal ribosomal protein uL1	900
radical sam	2480	TRAFAC class OBG-HflX-like GTPase	889
Prokaryotic	2408	FKBP-type PPIase	888
TrmE-Era-EngA-EngB-Septin-like GTPase	2293	MnmG	888
Cytochrome b	2242	NAD-dependent DNA ligase	876
Major facilitator	2191	Universal ribosomal protein uL3	867
Cyclohydrolase	2190	Universal ribosomal protein uL4	848
Metallo-dependent hydrolases	1859	DNA mismatch repair MutS	843
Transferase hexapeptide Repeat	1772	Tetrahydrofolate dehydrogenase	842
Cytochrome P450	1500	SHMT	836
DEAD box helicase	1460	RecA	833
MurCDEF family	1405	Adenylosuccinate synthetase	829
Heat shock protein 70	1378	Tetrahydrofolate dehydrogenasecyclohydrolase	828
EPSP synthase	1218	EF-Ts	826
AB hydrolase	1206	SecA	818
Universal ribosomal protein uS4	1150	RNA polymerase alpha chain	802
GHMP kinase	1143	IPP transferase	801
Chaperonin (HSP60)	1125	Glycosyltransferase 28	787
Universal ribosomal protein uS2	1094	Polyribonucleotide nucleotidyltransferase	783
Class-III pyridoxal-phosphate-dependent	1081	Fmt	765
reductases (SDR)	1070	RuvB	760
Short-chain dehydrogenases	1070	ATCase	705
Universal ribosomal protein uL2	1055	methyltransferase superfamily	519
Methylthiotransferase	1053	OTCase	406

found in [20]. Let's define protein sequence $S(n) = [MAKQDYY]$. The numerical representation of this sequences with CPNR is calculated as $[1\ \ 37\ \ 67\ \ 29\ \ 59\ \ 5\ \ 5\ \ ]$.

- **EIIP:** This method was proposed to predict protein-protein and protein-DNA interactions [21]. The method converted the genomic sequences into the signals and these converted signals were converted again to obtain power spectrum with Fourier transform. Power spectrum values were assigned to each amino acids. In [21], researchers can see the EIIP representation of each amino acid. For instance, $S(n) = [MAKQDYY]$ represents the protein sequence. The numerical representation of this sequence is calculated as $[0.0823\ \ 0.0373\ \ 0.0371\ \ 0.0761\ \ 0.1263\ \ 0.0516\ \ 0.0516\ \ ]$.
- **Hydrophobicity:** This method was developed based on the hydrophilic and hydrophobic tendencies of

polypeptide chains in protein sequences [22]. It is generally used in protein-protein interaction and protein function classification studies. Hydrophobicity values of each amino acid codes can be seen in [22]. Let's define protein sequence $S(n) = [MAKQDYY]$. The numerical representation of this sequences with hydrophobicity is calculated as $[1.9 \;\; 1.8 \;\; -3.9 \;\; -3.5 \;\; -3.5 \;\; -1.3 \;\; -1.3 \;\;]$.

- **Atchley Factors:** This encoding method was developed by the authors in [23] to resolve the protein sequence metric problem. It was derived from the large and interpretable components of protein variation and reflects the physiochemical properties of amino acid sequences. Atchley factor values of each amino acid codes were provided in [23]. Let's define protein sequence $S(n) = [MAKQDYY]$. The numerical representation of this sequence with Atchley factors is calculated as $[-0.663 \;\; -0.591 \;\; 1.831 \;\; 0.931 \;\; 1.050 \;\; 0.260 \;\; 0.260 \;\;]$.

- **PAM250:** This matrix was developed to score aligned protein sequences to designate the similarity between those sequences [24]. The numbers were assigned to each amino acid code by determining the PAM (accepted point mutations). Researchers can obtain PAM250 scores of each amino acid code in [24]. Let's define protein sequence as $S(n) = [MAKQDYY]$. The numerical representation of this sequence with PAM250 is calculated as $[6 \;\; 2 \;\; 5 \;\; 4 \;\; 4 \;\; 10 \;\; 10 \;\;]$.

- **BLOSUM62:** This method was derived from protein blocks to determine the measure of the protein sequences similarity [25]. The values found in BLOSUM62 were obtained by subtracting the values in the PAM160 matrix and the scores can be obtained in the study [25]. Let's define protein sequence $S(n) = [MAKQDYY]$. The numerical representation of this sequences with BLOSUM62 is calculated as $[5 \;\; 4 \;\; 5 \;\; 5 \;\; 6 \;\; 7 \;\; 7 \;\;]$.

- **Binary one-hot encoding:** In this conversion method, each amino acid is specified by a twenty dimensional binary vector. Twenty amino acids are sorted in an alphabetical order, and the i^{th} amino acid is designated with by twenty bits. For instance, [A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y] represent the amino acids. The binary one-hot value for D is 00100000000000000000, for F is 00001000000000000000. Let's define protein sequence as $S(n) = [ACD]$. The numerical representation of these sequences with binary one-hot is calculated as $[1000000000000000000 \;\; 01000000000000000000 \;\; 00100000000000000000 \;\;]$.

- **Randomly encoded method**: In this conversion method, we generated random numbers between 1, and 100 and assigned them to each amino acid code. Table 2 shows the random encoding scores for each amino in this study. Let's define protein sequence $S(n) = [MAKQDYY]$. The numerical representation of this sequences with randomly encoded method is calculated as $[69 \;\; 79 \;\; 96 \;\; 40 \;\; 30 \;\; 50 \;\; 50 \;\;]$.

3.3. The proposed FIBHASH method

In this paper, we propose a novel protein mapping method called FIBHASH. In this method, we apply both Fibonacci numbers and hashing table. Fibonacci numbers are a sequence in which each term is the sum of two numbers preceding it. It can be represented as $F(n) = F_{n-1} + F_{n-2}$, where $F_1 = 1$, and $F_2 = 1$ for all $n > 3$. In the representation, n specifies the nth Fibonacci number. In the nature, Fibonacci sequences can be observed clearly. We can observe the Fibonacci numbers in flowers, organs of human body, physics,

Table 2. Amino acid encoding table with randomly encoded representation.

Amino acids	Abbreviation	Quantization	Amino acids	Abbreviation	Quantization
Methionine	M	69	Glutamine	Q	40
Tryptophan	W	78	Serine	S	88
Phenylalanine	F	32	Alanine	A	79
Tyrosine	Y	50	Asparagine	N	66
Proline	P	13	Glycine	G	54
Cysteine	C	70	Arginine	R	42
Threonine	T	37	Isoleucine	I	96
Histidine	H	1	Aspartic	D	30
Valine	V	74	Glutamic	E	58
Leucine	L	46	Lysine	K	96

etc. [26]. Fibonacci numbers are so common in nature, so we can find this similarity in microcosmos of life, in proteins. If we evaluate the life forms in nature, some of them is admissible to be compared with the sequence but it is not widely accepted in biology. Some supporters of golden ratio and its relation to Fibonacci sequence, makes remodeling of organisms' structures flawlessly fit to it. On other hands, Fibonacci numbers is a mathematical model in science, describing or identifying something in life with a collection of ordered numbers in an accurate way. In addition to these, it was demonstrated that the human genome includes fractal behavior, which indicates Fibonacci series and the golden proportion [27]. In the study of [28], author provided that the gene-coding regions of DNA sequences were highly related to the Fibonacci numbers. Further, in one study [29], an evidence was presented that the Fibonacci numbers map codons to amino acids. In the study of [30], author provided the Fibonacci sequences to determine the genetic code of amino acid sequences. Firstly, protein sequences were sorted by their atomic numbers including carbon, nitrogen, oxygen, and sulfur. Later, protein sequences were mapped with Fibonacci chain based on two elements: S (small), and L (large) atoms. In the study, Fibonacci sequences were started from zero as in this study. Inspired by these works, and observations, we apply Fibonacci numbers to map protein sequences as numerical representations. Proteins are the essential molecules, because they perform various of functions to sustain life. They are responsible for transportation of molecules from one cell to other cells, DNA replication, acceleration of metabolic reactions, and many other important functions. There are 20 amino acids, which are accepted, reviewed, and used in studies, thus we calculate Fibonacci numbers from 1 to 20. After calculation, we assign each Fibonacci number to each amino acid in alphabetical order. Table 3, shows the Fibonacci representations of each amino acid.

Because of the size of the decimal places of Fibonacci numbers after the $12th$ index, we map these representations to the hash table. When we apply the Fibonacci numbers without using hash table, the calculation process is complex and it is not time-efficient. Thus, it is needed to map these Fibonacci numbers on a hash table. Hash table is a data structure and it stores the data in a correlated manner. In a hash table, data is stored in an array format, where each data has its own unique index value. Accession of the data is very fast since the insertion and search operations are quick irrespective of the size of the data [31, 32]. Values are added to the hash table with a hashing technique. This technique converts the range of key values to the range of indexes of an array by using modulo operator. Consider an example of hash table of size 5, and these values $(15, 3, 16, 32, 4)$ are to be stored. The calculation is given in Table 4.

Table 3. Amino acid encoding table with Fibonacci numbers representation.

Amino acids	Abbreviation	Quantization	Amino acids	Abbreviation	Quantization
Methionine	M	89	Glutamine	Q	377
Tryptophan	W	4181	Serine	S	987
Phenylalanine	F	5	Alanine	A	1
Tyrosine	Y	6765	Asparagine	N	144
Proline	P	233	Glycine	G	8
Cysteine	C	1	Arginine	R	600
Threonine	T	1597	Isoleucine	I	21
Histidine	H	13	Aspartic	D	2
Valine	V	2584	Glutamic	E	3
Leucine	L	55	Lysine	K	34

Table 4. Example of a hashing operation.

Value	Hash	Index
15	15 % 5	0
3	3 % 5	3
16	16 % 5	1
32	32 % 5	2
4	4 % 5	4

However, in some cases, values are needed to be stored in the same index. This is called collision. To solve the collisions, many methods were proposed such as linear probing, quadratic probing, double hashing [31]. In linear probing, next empty location in the array is searched by looking into the next cell until an empty cell is found. In the proposed conversion method, collision is occurred and to solve the collusion, we performed linear probing. We developed a hash table of size 20 (because of the number of amino acids), and inserted the mapped amino acids into this hash table. Collisions are solved by linear probing and index values are considered to encode the amino acid sequences. Table 5 infers the FIBHASH encoding results of amino acids. Let's define protein sequence $S(n) = [MAKQDYY]$. The numerical representation of this sequences with FIBHASH is calculated as $[9\ \ 1\ \ 14\ \ 17\ \ 2\ \ 19\ \ 19\ \ ]$. After the encoding processes, we classified the protein families by applying RNN, LSTM, BRNN (Bidirectional RNN), and BLSTM (Bidirectional LSTM).

3.4. Deep learning models

In this paper, we use four deep learning models: RNN, LSTM, BRNN, and BLSTM. RNN is a deep learning algorithm and takes time series as an input to perform classification and prediction. RNN is a kind of feed-forward neural network, which has an internal memory. In general, it applies the similar functions for every types of input, while the output depends on the past calculation. After the calculation process, it is sent back to the recurrent network. The detailed information about RNN can be found in [33]. LSTM is a modified version of RNN. In RNN, there is a vanishing gradient problem. The main reason of developing LSTM networks is to resolve this issue. It is effective in classification and prediction of time series [33, 34]. Simply put, BRNN was

Table 5. Amino acid encoding table with the proposed FIBHASH method.

Amino acids	Abbreviation	Quantization	Hash	Collision (if any)	Index
Alanine	A	1	1 % 20	-	1
Cysteine	C	1	1 % 20	+1	2
Aspartic	D	2	2 % 20	+1	3
Glutamic	E	3	3 % 20	+1	4
Phenylalanine	F	5	5 % 20	-	5
Glycine	G	8	8 % 20	-	8
Histidine	H	13	13 % 20	-	13
Isoleucine	I	21	21 % 20	+5	6
Lysine	K	34	34 % 20	-	14
Leucine	L	55	55 % 20	-	15
Methionine	M	89	89 % 20	-	9
Asparagine	N	144	144 % 20	+3	7
Proline	P	233	233 % 20	+3	16
Glutamine	Q	377	377 % 20	-	17
Arginine	R	600	600 % 20	-	0
Serine	S	987	987 % 20	+3	10
Threonine	T	1597	1597 % 20	+1	18
Valine	V	2584	2584 % 20	+7	11
Tryptophan	W	4181	4181 % 20	+11	12
Tyrosine	Y	6765	6765 % 20	+14	19

developed by combining two RNN structures. In BRNN, the input sequence is given in its normal order for one network, and in reverse time order for another network. Generally, the output of the two networks is combined by summation [35]. BLSTM networks are the extension of LSTMs, which can improve the model performance on sequence classification problems [36–38]. In this paper, we both used RNN, LSTM, BRNN, and BLSTM to classify the protein families. We compared the performance of all protein mapping methods regarding the classification process. To evaluate the success of the deep learning models, we calculated the accuracy, precision, recall, f1-scores, and AUC scores of each protein mapping method for each recurrent neural networks. Workflow of the proposed method is given Figure 1.

4. Application results

In this section, we provide the application results of each classifiers. The evaluation criteria of all recurrent neural networks were defined as accuracy, precision, recall, f1-score, and AUC scores. The calculation formula of these metrics can be seen in [32, 39, 40]. Classification of protein families was performed with four different application scenarios. In the first scenario, we applied RNN; in the second we carried out LSTM; in the third scenario, we considered BRNN; and in the last scenario, we implemented BLSTM.

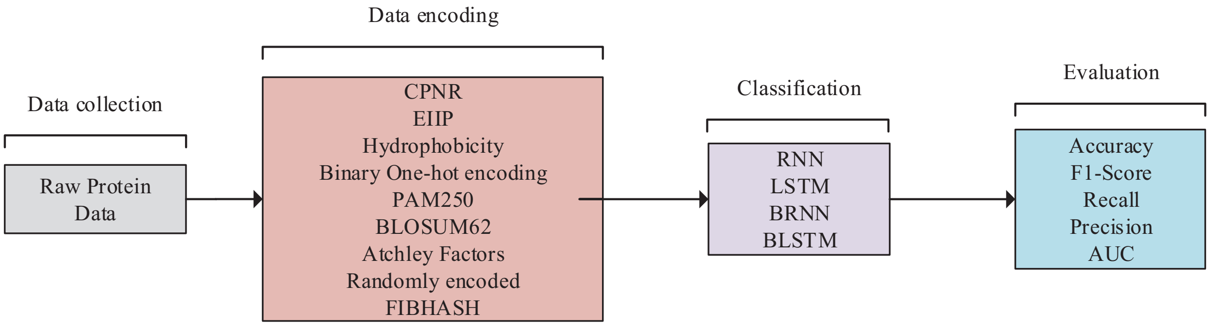

Figure 1. Workflow of the classification process.

4.1. The performance comparison by using the RNN

In the first scenario, we performed RNN to classify the protein families. In the application, we set the ratio of the training data set and testing dataset as 80%, and 20%, respectively. The parameters of the developed RNN model were determined with trial and error approach with number of 200 epochs and can be summarized as follows:

- In input layer, encoded protein sequences in the size of 97,576 × 1 were applied.
- In the RNN layer, we applied number of 256 RNN units with ReLU activation function.
- Later, fully connected layers of 1024 and 512 neurons were used, respectively.
- In the last layer, Softmax function was applied and classification was performed.
- In order to determine the loss of the model, categorical crossentropy was calculated. As an optimizer, Adam optimizer with default parameters was considered.

The evaluation results of RNN classifier for both protein encoding modules are given in Table 6. According to results given in the Table 6, both protein encoding modules did not perform well for the protein families' classification. Yet the best accuracy, and AUC scores were obtained from the proposed FIBHASH method with 0.57, and 0.67, respectively. The AUC score of 1.00 means that the method is effective to identify the protein families. In bioinformatics and medical studies, AUC score is really important [41–43]. If the AUC score of a classifier is greater than 70%, that classifier is considered fair [44]. In this case, the RNN classifier did not perform well with the methods and parameters mentioned. This performance of the RNN classifier is not surprising, since it suffers greatly from vanishing gradients.

4.2. The performance comparison by using the LSTM

In the second scenario, we developed LSTM to classify the protein families. As in RNN application, in this scenario, training and test data were determined as 80%, and 20%, respectively. All of the parameters of LSTM model were determined with trial and error approach with the number of 200 epochs and can be stated as follows:

- In input layer, with the size of 97,576 x 1 encoded protein sequences were applied.
- In the LSTM layer, we applied number of 256 LSTM units with ReLU activation function.

Table 6. Evaluation results of RNN classification for both protein encoding modules.

Encoding module	Accuracy	F1-Score	Precision	Recall	AUC
Atchley factors	0.5515	0.4020	0.6970	0.2825	0.65
Binary One-hot encoding	0.5519	0.4610	0.6809	0.3153	0.64
BLOSUM62	0.4602	0.1998	0.5527	0.1220	53
CPNR	0.4976	0.3762	0.6175	0.2706	0.61
EIIP	0.3446	0.2108	0.2523	0.1811	0.52
Hydrophobicity	0.4897	0.3789	0.5779	0.2819	0.58
PAM250	0.5055	0.4474	0.3797	0.5446	0.55
Randomly encoded	0.4600	0.4166	0.5252	0.3453	0.52
The proposed method (FIBHASH)	**0.5700**	**0.4438**	**0.6536**	**0.3360**	**0.67**

- Later, fully connected layers of 1024, and 512 neurons were used, respectively.
- In the last layer Softmax function was applied and classification was performed.
- In order to determine the loss of the model, categorical crossentropy was calculated. As an optimizer, Adam optimizer with default parameters was considered.

The evaluation results of LSTM classifier for both protein encoding modules are given in Table 7.

Table 7. Evaluation results of LSTM classification for both protein encoding modules.

Encoding module	Accuracy	F1-Score	Precision	Recall	AUC
Atchley factors	0.6454	0.6033	0.6572	0.5576	0.68
Binary One-hot encoding	0.6005	0.5354	0.6685	0.4466	0.74
BLOSUM62	0.6692	0.6040	0.7077	0.5269	0.77
CPNR	0.7393	0.7165	0.7884	0.6567	0.86
EIIP	0.5116	0.4392	0.5992	0.3467	0.70
Hydrophobicity	0.4253	0.6399	0.6874	0.5987	0.50
PAM250	0.6454	0.5979	0.6883	0.5285	0.73
Randomly encoded	0.5963	0.6863	0.6966	0.6764	0.71
The proposed method (FIBHASH)	**0.6947**	**0.6785**	**0.7455**	**0.6227**	**0.83**

As can be seen from Table 7, the results of the LSTM model are better than the RNN model. Almost all AUC values of all methods were above 70%. Only the proposed method and CPNR's AUC scores were higher than 80%. This shows that these methods are successful in the classification process since the AUC scores lies between 70%–80%. In addition, it has been observed that the proposed method is at least as effective as others, and even produced better results than some various other methods.

4.3. The performance comparison by using the BRNN

In the third scenario, we developed BRNN to discriminate the protein families. In this scenario, as in other previous scenarios, the data were split according to the 80-20 train-test split approach. In addition, the parameters of the BRNN model were considered with trial and error approach with the number of 200 epochs, and can be summarized as follows:

- In input layer, with the size of 97,576 x 1 encoded protein sequences were applied.
- In the bidirectional layer, the number of 256 RNN units with ReLU activation function were provided.
- In the second bidirectional layer, like in first bidirectional layer, the number of 256 RNN units were applied.
- Then, flatten, batch normalization, and dropout (0.25) operations were done, respectively.
- Later, fully connected layers of 1024, and 512 neurons were used, respectively.
- In the last layer Softmax function was applied and classification was performed.
- In order to determine the loss of the model, categorical crossentropy was calculated. As an optimizer, Adam optimizer with default parameters was considered.

The evaluation results of BRNN classifier for both protein encoding modules are given in Table 8.

Table 8. Evaluation results of BRNN classification for both protein encoding modules.

Encoding module	Accuracy	F1-Score	Precision	Recall	AUC
Atchley factors	0.7959	0.8031	0.8229	0.7844	0.85
Binary One-hot encoding	0.6419	0.6377	0.6945	0.5895	0.73
BLOSUM62	0.7197	0.7065	0.7892	0.6395	0.81
CPNR	0.7905	0.7753	0.8339	0.7245	0.83
EIIP	0.7388	0.7298	0.7902	0.6781	0.74
Hydrophobicity	0.7954	0.7853	0.8282	0.7467	0.88
PAM250	0.7403	0.7274	0.7745	0.6858	0.79
Randomly encoded	0.6923	0.7035	0.7895	0.6345	0.72
The proposed method (FIBHASH)	**0.7847**	**0.7917**	**0.8193**	**0.7660**	**0.90**

According to the evaluation results in Table 8, all of the protein encoding method performed well and had higher AUC scores than 70%. Yet only the proposed method has reached the excellent testing performance with 0.90 AUC score. The proposed method, which has the fourth best classification accuracy, showed at least as good classification performance as others. Although, LSTM results were good, we did not reach an effective AUC values. Thus, we needed third and fourth alternative scenarios. In RNN and LSTM models, input values were processed only based on the past information. This causes the information to be processed in a limited way [35]. Yet, in BRNN and BLSTM, inputs are processed in both forward and reverse time order which allows BRNN and BLSTM to look at future context as well [35, 36]. For these reasons, we thought that the BRNN and BLSTM models would give better results, and we performed the classification using these methods in the third and fourth scenarios.

4.4. The performance comparison by using the BLSTM

In the last scenario, we designed BLSTM to classify the protein families. In this scenario, the data were split as training and testing data according to the 80-20 approach. All of the parameters of the BLSTM were determined with trial and error approach with the number of 200 epochs. The parameters of BLSTM model can be expressed as follows:

- In input layer, with the size of 97,576 x 1 encoded protein sequences were applied.
- In the bidirectional layer, the number of 256 LSTM units with ReLU activation function were provided.
- In the second bidirectional layer, we provided the number of 256 LSTM units.
- Then, flatten, batch normalization, and dropout (0.25) operations were carried out, respectively.
- Later, fully connected layers of 1024 and 512 neurons were developed, respectively.
- In the last layer, Softmax function was applied and classification was achieved.
- In order to determine the loss of the model, categorical crossentropy was calculated. As an optimizer, Adam optimizer with default parameters was considered.

In Figure 2, we provided the general architecture of BLSTM model.

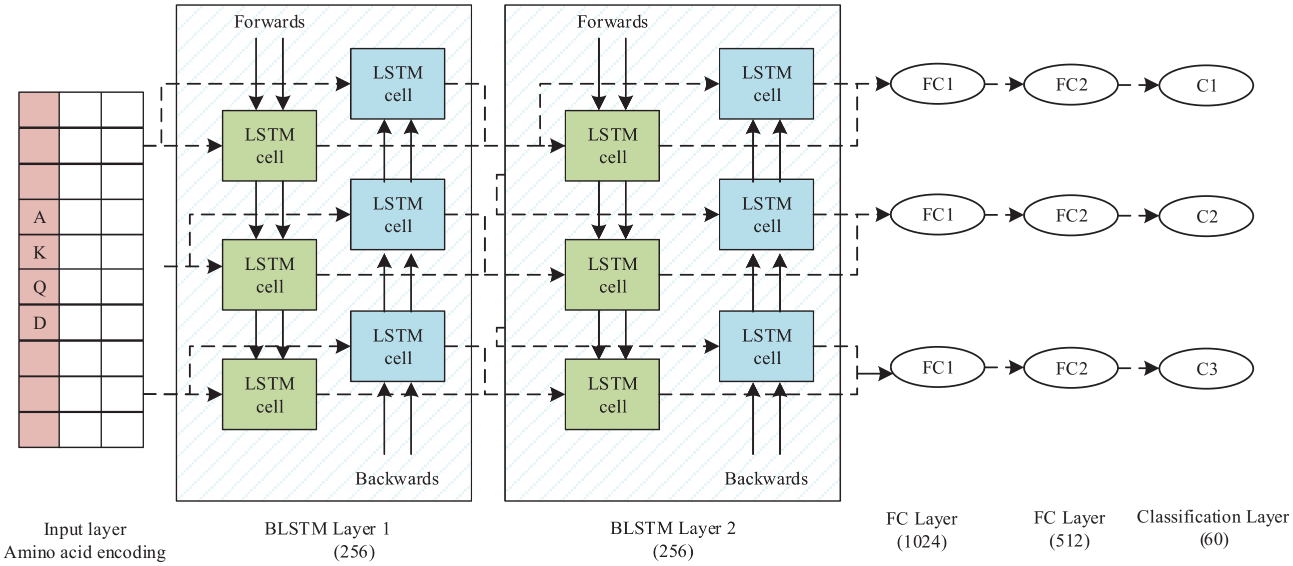

Figure 2. The architecture of the BLSTM model for protein family classification.

The evaluation results of BLSTM model for both protein encoding modules are given in Table 9.

According to Table 9, all of the AUC scores are higher than 0.80, which means that all of the protein mapping method performed well with the BLSTM deep learning model. The proposed method provided the best classification accuracy of 92.77%, and excellent AUC score of 0.98. According to these results, the best classification process was performed with the proposed method. In compliance with the accuracy, and AUC results obtained from 4 different deep learning models, it may be indicated that the proposed method is as effective as the other methods. Since the best classification results obtained from the BLSTM, we provided the AUC scores of each mapping methods with BLSTM in Figure 3.

When the results are examined, it can be said that some mapping methods are successful in the classification process. Yet, considering all the results, EIIP and hydrophobicity method are not successful enough.

Table 9. Evaluation results of BLSTM classification for both protein encoding modules.

Encoding module	Accuracy	F1-Score	Precision	Recall	AUC
Atchley factors	0.7823	0.7322	0.8879	0.6231	0.84
Binary One-hot encoding	0.8244	0.7782	0.9041	0.6832	0.85
BLOSUM62	0.8563	0.8233	0.9262	0.7410	0.96
CPNR	0.8850	0.8506	0.9423	0.7753	0.96
EIIP	0.7876	0.7555	0.8751	0.6648	0.86
Hydrophobicity	0.8232	0.7484	0.9622	0.6124	0.91
PAM250	0.8446	0.8331	0.9676	0.7315	0.90
Randomly encoded	0.8542	0.7989	0.8925	0.7232	0.90
The proposed method (FIBHASH)	**0.9277**	**0.8831**	**0.9788**	**0.8046**	**0.98**

The reason for this may be that these methods have become degenerated [20, 39], since different amino acids can be mapped to same numbers (Q, and N in hydrophobicity, I, and L in EIIP). Further, the performance of the random encoded method depends on the number generated. If other random numbers were assigned to each code, the performance of random encoded method may be better or worse. The BLOSUM62 and PAM250 matrices are mostly used for the alignment process in the literature [40]. Yet, in this work, proteins were mapped with these methods, and in general, the results were better than some methods. The best results have not been achieved, and the main reason may be that these methods, including Atchley factors are more effective in prediction of secondary structure of proteins [9]. The performance of binary one-hot seems fair. The main reason for this may be due to dimension of this method since too large dimensions may lead to poor performance. If one-hot (6-bit) or binary 5-bit were used, the application results could be different.

5. Conclusion

In this paper, we constructed a novel protein encoding method to predict protein families with recurrent neural networks, including RNN, LSTM, BRNN, and BLSTM. Compared to the other encoding methods, our approach uses Fibonacci numbers and hash tables to address the index values of amino acid codes. Fibonacci numbers can be found in nature as well as in human genomes. The human genome provides fractal behavior, which is related to the Fibonacci numbers. Further, Fibonacci numbers can be observed to determine the genetic code of amino acid sequences. Based on these information, we provided a novel protein mapping method via Fibonacci numbers. As the digit value increased, Fibonacci numbers increased, thus we had to decrease the dimension of these values. Therefore, we applied hash table to reduce the size of numbers and combined these two approaches to provide mapping method. In addition, we developed four kinds of recurrent neural network models in four different scenarios to elicit the generation and selection of features manually. In our paper, protein sequences were encoded with several mapping methods including Atchley factors, binary one-hot encoding, CPNR, EIIP, BLOSUM62, PAM250, hydrophobicity, and randomly encoded. After, each encoded sequences were used as an input for both recurrent neural networks. The performance of both encoding modules and classification models were evaluated with accuracy, precision, recall, f1-score, and AUC scores. The experimental results proved that our method is effective as state-of-art methods. With RNN classification, the best results were obtained from the proposed FIBHASH method with an accuracy of 57%, and an AUC score of 0.67. Yet AUC score was lower than 0.70, means that the RNN classifier did not perform well for both protein mapping methods. In the second

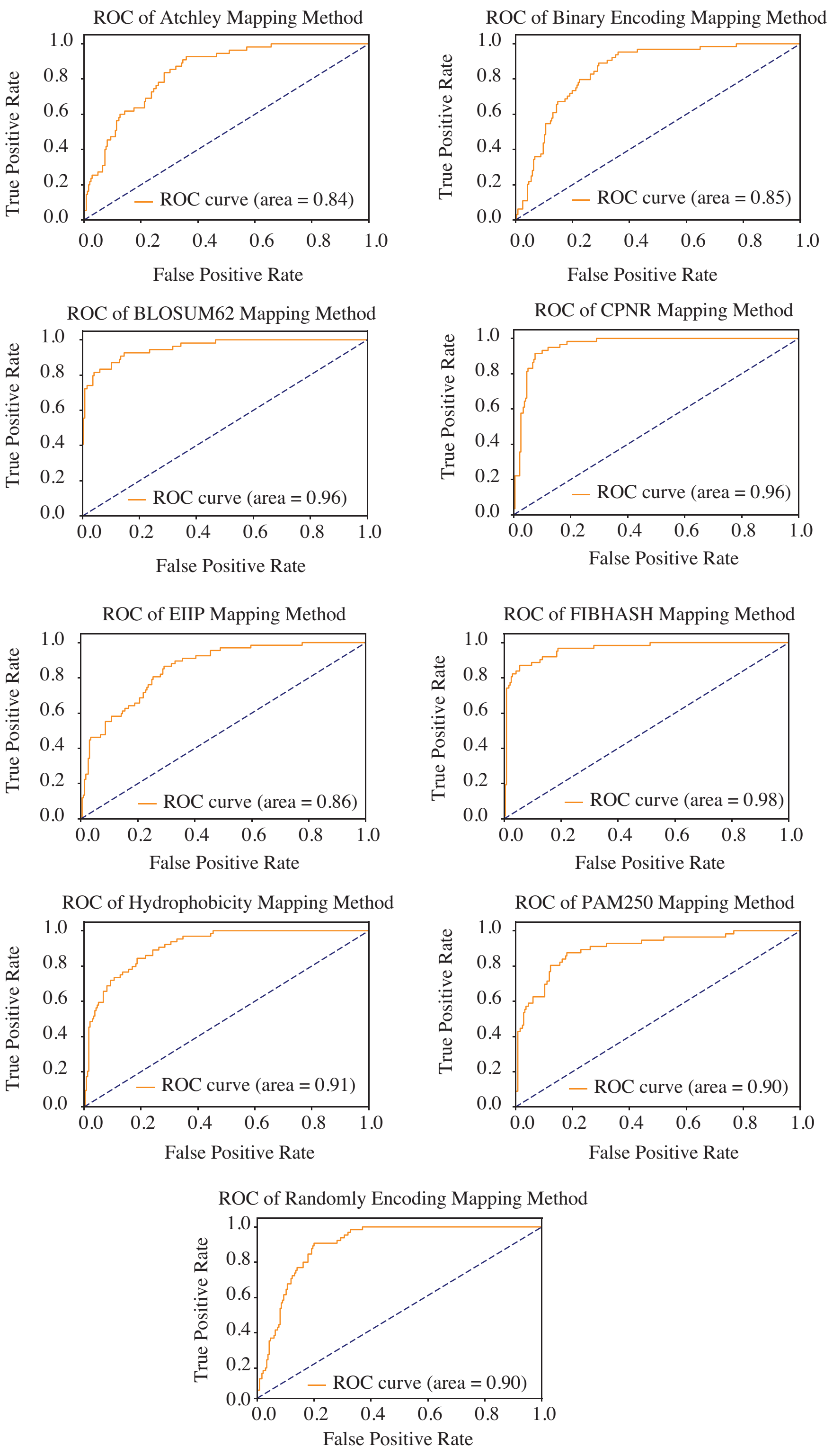

Figure 3. AUC scores of each protein mapping methods.

scenario, we developed LSTM model to determine the protein families. The accuracy and AUC scores were increased with this model, and we obtained 69.47% accuracy, and 0.83 AUC score with the proposed method. These results are acceptable since the AUC score is higher than 0.70. Yet, the best classification accuracy and AUC score were obtained with CPNR with an accuracy of 73.93%, and 0.86 AUC score. Although the proposed method did not perform best, we can say that it is at least as effective as CPNR. Due to some disadvantages of RNN and LSTM models, we used BRNN, and BLSTM in the third and fourth scenario, respectively. In the third scenario, the proposed method reached the best AUC score (0.90) indicating that the proposed method has an excellent testing performance. In addition, all AUC scores of all protein mapping methods were higher than 0.70, therefore we considered that the BRNN was successful. Among other scenarios, we achieved the best evaluation criteria in the fourth scenario for all mapping methods. In this scenario, nearly all of the protein mapping methods reached the excellent AUC scores. Yet, the best performance was obtained from the proposed FIBHASH method with an accuracy of 92.77%, and AUC score of 0.98. At the end of the study, it was observed that the proposed method is at least as effective as other methods and even more successful in some cases. Also, the proposed method may be applied effectively;

- In determining the protein functions,
- In drug therapy and drug development studies,
- In identification and classification of protein families,
- In phylogenetic analysis studies,
- In determining viral-host protein interactions.

In the future studies, the proposed method will be used and tested on mentioned different protein studies.

References

[1] Nguyen N, Nute M, Mirarab S, Warnow Tandy. HIPPI: Highly accurate protein family classification with ensembles of HHMs. BMC Genomics 2016; 17(765): 89-100. doi: 10.1186/s12864-016-3097-0

[2] Dawson N, Sillitoe I, Marsden RL, Orengo CA. The classification of protein domains. Methods in Molecular Biology 2017; 1525: 137-164. doi: 10.1007/978-1-4939-6622-6_7

[3] Enright AJ, Van Dongen S, Ouzonis CA. An efficient algorithm for large-scale detection of protein families. Nucleic Acid Research 2022; 30(7): 1575-1584. doi: 10.1093/nar/30.7.1575

[4] Bork P, Dandekar T, Diaz-Lazcoz Y, Eisenhaber F, Huynen M et al. Predicting function: from genes to genomes and back. Journal of Molecular Biolog 1998; 283(4): 707-725. doi: 10.1006/jmbi.1998.2144

[5] Remmert M, Biegert A, Hauser A, Söding J. HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nature Methods 2012; 9(2): 173-175. doi: 10.1038/nmeth.1818

[6] Vazhayil A, Vinayakumar R, Soman KP. DeepProteomics: Protein Family Classification Using Shallow and Deep Networks. Cold Spring Harbor, NY, USA: Cold Spring Harbor Laboratory, 2018.

[7] Zamani M, Kremer SC. Amino acid encoding schemes for machine learning methods. In: 2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW); Atlanta, GA, USA; 2011. pp. 327-333.

[8] Yin C, Yau SST. A coevolution analysis for identifying protein-protein interactions by Fourier transform. PLOS ONE 2017; 12(4): e0174862. doi: 10.1371/journal.pone.0174862

[9] Jing X, Dong Q, Hong D, Lu R. Amino acid encoding methods for protein sequences: A comprehensive review and assessment. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2019; Early access. doi: 10.1109/TCBB.2019.2911677

[10] Zacharaki EI. Prediction of protein function using a deep convolutional neural network ensemble. PeerJ Computer Science 2017; 3(e124): 1-17. doi: 10.7717/peerj-cs.124

[11] Zhang D, Rabuka MR. Protein family classsification from Scratch: A CNN based deep learning approach. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2020; early acess. doi: 10.1109/TCBB.2020.2966633

[12] Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nature Biotechnology 2015; 33(8): 831-838. doi: 10.1038/nbt.3300

[13] Hüsken M, Stagge P. Recurrent neural networks for time series classification. Neurocomputing 2003; 50: 223-235. doi: 10.1016/S0925-2312(01)00706-8

[14] Jin X, Yu X, Wang X, Bai Y, Su T et al. Prediction for time series with CNN and LSTM. In: 11th International Conference on Modelling, Identification and Control (ICMIC2019); Tianjin, China; 2019. pp. 631-641.

[15] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 1997; 45(11): 2673-2681. doi: 10.1109/78.650093

[16] Naveenkumar KS, Harun M, Babu R, Vinayakumar R, Soman KP. Protein family classification using deep learning. bioRxiv 2018; 414128: doi: 10.1101/414128

[17] Lee TK. Protein family classification with neural networks. MSc., University of Stanford, California, USA, 2016.

[18] Zhang D, Kabuka MR. Protein family classification with .ulti-layer graph convolutional networks. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM); Madrid, Spain; 2018. pp. 2390-2393.

[19] Chen J, Chaudhari NS. Protein family classification using second-order recurrent neural networks. Genome Informatics 2003; 14: 520-521.

[20] Chen D, Wang J, Yan M, Bao FS. A complex numerical representation of amino acids for protein function comparison. Journal of Computational Biology 2016; 23(8): 669-677. doi: 10.1089/cmb.2015.0178

[21] Veljkovic N, Glisic S, Prljic J, Perovic V, Botta M et al. Discovery of new therapeutic targets by the informational spectrum method. Current Protein and Peptide Science 2008; 9(5): 493-506. doi: 10.2174/138920308785915245

[22] Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. Journal of Molecular Biology 1982; 157(1): 105-132. doi: 10.1016/0022-2836(82)90515-0

[23] Atchley WR, Zhao J, Fernandes AD, Drüke T. T. Solving the protein sequence metric problem. PNAS 2005; 102(18); 6395-6400. doi: 10.1073/pnas.0408677102

[24] Dayhoff MO, Schwartz RM, Orcutt BC. A model of evolutionary change in proteins. National Biomedical Research Foundation 1978; 5(3): 345-352.

[25] Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences of the United States of America 1992; 89(22): 10915-10919. doi: 10.1073/pnas.89.22.10915

[26] Sinha S. The Fibonacci numbers and its amazing applications. International Journal of Engineering Science Invention 2017; 6(9): 7-14.

[27] Persaud D, O'Leary JP. Fibonacci series, golden proportions, and the human biology. Austin Journal of Surgery 2015; 2(5): 1066.

[28] Perez JC. Chaos, DNA and neuro-computers: A golden link. Speculations in Science and Technology 1991; 14(4): 336-347.

[29] Perez JC. Codon populations in single-stranded whole human genome DNA are fractal and fine-tuned by the Golden Ratio 1.618. Interdisciplinary Sciences: Computational Life Sciences 2010; 2(3): 228-240. doi: 10.1007/s12539-010-0022-0

[30] Negadi T. A mathematical model for the genetic code(s) based on Fibonacci numbers and their q-analogues. NeuroQuantology: An Interdisciplinary Journal of Neuroscience and Quantum Physics 2015; 13(3): 259-272. doi: 10.14704/nq.2015.13.3.850

[31] Weiss MA. Data Structures & Algorithm Analysis in C++. USA: Pearson, 2013.

[32] Nimbe P, Frimpong SO, Opoku M. An efficent way strategy for collision resolution in hash tables. International Journal of Computer Applications 2014; 99(10): 35-41. doi: 10.5120/17411-7990

[33] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long-short term memory (LSTM) network. Physica D: Nonlinear Phenomena 2020; 404. 132306. doi: 10.1016/j.physd.2019.132306

[34] Hochreiter S, Schmidhuber J. Long-short term memory. Neural Computation 1997; 9(8): 1735-1780. doi 10.1162/neco.1997.9.8.1735

[35] Cai R, Zhang X, Wang H.Bidirectional recurrent convolutional neural network for relation classification. In: 2016 54th Annual Meeting of the Association for Computational Linguistics; Berlin, Germany; 2016. pp. 756-765.

[36] Liu G, Guo J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 2019; 337: 325-338. doi: 10.1016/j.neucom.2019.01.078

[37] Basaldella M, Antolli E, Serra G, Tasso C. Bidirectional LSTM recurrent neural network for keyphrase extraction. In: 2018 14th Italian Research Conference on Digital Libraries; Udine, Italy; 2018. pp. 180-187.

[38] Graves A, Jaitly N, Mohamed A. Hybrid speech recognition with deep bidirectional LSTM. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding; Olomouc, Czech Republic; 2013. pp. 273-278.

[39] Skutkova H, Maderankova D, Sedlar K, Jugas R, Vitek M. A degeneration-reducing criterion for optimal digital mapping of genetic codes. Computational and Structural Biology 2019; 17: 406-141. doi: 10.1016/j.csbj.2019.03.007

[40] Durbin R. Biological Sequence Analysis: Probabilistic Models of Proteins and Nuclear Acids. Cambridge, UK: Cambridge University Press, 1998.

[41] Kamarudin AN, Cox T, Kolamunnage-Dona R. Time-dependent ROC curve analysis in medical research: Current methods and applications. BMC 2017; 17(1): 53. doi: 10.1186/s12874-017-0332-6

[42] Safari S, Baraloo A, Elfil M, Negida A. Evidance based emergency medicine; part 5 receiver operation curve and area under the curve. Emergency 2016; 4(2): 111-113.

[43] Zhao XG, Dai W, Li Y, Tian L. AUC-based biomarker ensemble with an application on gene scores prediction low bone mineral density. Bioinformatics 2011; 27(21): 3050-3055. doi: 10.1093/bioinformatics/btr516

[44] Wington RS, Connor JL, Centor RM. Transportability of a decision rule for the diagnosis of streptococcal pharyngitis. Archives of International Medicine 1986; 146(1): 81-83.