

LSTM-GRU Based Deep Learning Model with Word2Vec for Transcription Factors in Primates

Ali Burak Oncul


Abstract— The study of the structures of proteins and the relationships of amino acids remains a challenging problem in biology. Although some bioinformatics-based studies provide partial solutions, some major problems remain. At the beginning of these problems are the logic of the sequence of amino acids and the diversity of proteins. Although these variations are biologically detectable, these experiments are costly and time-consuming. Considering that there are many unclassified sequences in the world, it is inevitable that a faster solution must be found. For this reason, we propose a deep learning model to classify transcription factor proteins of primates. Our model has a hybrid structure that uses Recurrent Neural Network (RNN) based Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks with Word2Vec preprocessing step. Our model has 97.96% test accuracy, 97.55% precision, 95.26% recall, 96.22% f1-score. Our model was also tested with 5-fold cross-validation and reached 97.42% accuracy result. In the prepared model, LSTM was used in layers with fewer units, and GRU was used in layers with more units, and it was aimed to make the model a model that can be trained and run as quickly as possible. With the added dropout layers, the overfitting problem of the model is prevented.

Index Terms—Protein classification, Hybrid deep learning, Word2Vec, LSTM, GRU

I. INTRODUCTION

THE HEREDITARY material of living things is Deoxyribonucleic Acid (DNA). DNA sequences consist of adenine (A), guanine (G), cytosine (C), and thymine (T) nucleotides and, with their various sequences, form the structures necessary for the creation and survival of living things with all their units [1], [2]. One of these structures is amino acids. There are over 300 amino acids in nature. 20 of these amino acids are mostly found in mammals and plants. These amino acids produced by DNA are arranged in different ways to form proteins. Proteins are involved in almost all vital tasks in living things, such as growth, measures under stress, and communication of cells. [3]. Many amino acid sequences have been discovered in light of scientific studies and advances in science. The discovered sequences belong to many different realms, species, and families.

ALİ BURAK ÖNCÜL, is with Department of Computer Engineering Department of Kastamonu University, Kastamonu, Turkey, (e-mail: boncul@kastamonu.edu.tr).

 <https://orcid.org/0000-0001-9612-1787>

Manuscript received October 18, 2022; accepted Nov 13, 2022.
DOI: [10.17694/bajece.1191009](https://doi.org/10.17694/bajece.1191009)

Each unit that is different has many different structural features. These different features determine the types and groups of these amino acid sequences, namely proteins.



Fig.1. Example of TF protein (bHLH) [4].

Transcription is the first step of gene expression from DNA to RNA transcript production in a gene. This process, together with the continuation steps, carries out the production of the protein. Transcription factors (TFs) are genomic regions that can bind to certain DNA sequences and fragments, directing gene expression in different ways in cells and, thus, organisms. TFs are proteins that manage the transfer of genetic information from DNA to mRNA. Each TF has a structure specific to the sequences [5]–[7]. Animal TF proteins are proteins that have essential roles in regulating many vital functions of cells, such as their development, communication, response under stress, and cell cycle. These proteins, like other proteins, have been identified and sequenced by biological experiments, and new sequences are being discovered every day [8]. Figure 1 shows a TF protein from basic helix-loop-helix TF family.

Conventional biological research methods can classify these many different proteins. However, these classification processes consist of both costly and relatively long-term experiments. In addition, the possibility of error by the experimenter is among the handicaps of these traditional methods [9]. This situation has pushed researchers to search for new analysis methods. In this field, especially in studies conducted jointly with computer scientists, these data were firstly the subject of statistical models [10]. Examples of these models are the studies prepared with the Hidden Markov Model (HMM) [11], which is widely used [12], and the Basic Local Alignment Search Tool (BLAST) [13] study. Acting on the principles of statistical science, these studies act and make predictions according to the probabilities of each amino acid in the positions in the protein sequence. In this way, the classes and structures of the arrays are determined. However, the models developed by these studies work better with annotated, i.e., additional labeled data such as sequence or the primat

name. For this reason, the success rate will be relatively low in plain sequences [14], [15].

There have also been studies in which artificial intelligence applications such as machine learning and deep learning, which are used in different fields and studies, are used in the classification of proteins in the field of bioinformatics. Examples of machine learning algorithms used for this purpose are Naive Bayes (NB) Classifier, Gradient Descent Algorithm (GD), K-Nearest Neighbor (KNN) algorithm, and Support Vector Machine (SVM). Recursive neural networks (RNN) and convolutional neural networks (CNN) based deep learning applications, which have been brought to the literature with the developments in computer science and the latest innovations in artificial intelligence, are also used for different kinds of classification problems in this field [16]. In addition to these studies, studies have been carried out on similar data with various deep learning models, together with the developments in the field [17].

II. RELATED WORKS

Looking at the literature, different bioinformatic studies have been developed in addition to biological studies to analyze various protein sequences. These studies draw their strength from the intersection of biological studies and computer science. Although these may be statistically based, in recent studies, the emphasis has been on artificial neural networks, machine learning, and especially deep learning studies [18]. These studies have been included in the literature with different data sets and models. In a study dealing with gene ontology and protein function analysis, PSI-BLAST and position-specific scoring matrix (PSSM) were used, along with the alignment results of PSI-BLAST and position information of PSSM, protein functions, and annotations were determined in new genomes [19]. A KNN-based OET-KNN model was developed in a study that controls protein sequences to become enzymes and determines their sub-functional classes. This model, with its variations, has achieved success on a scale of 86%-98% [20]. In another study, which detects the enzymatic functions of proteins, the ECPred model was developed, and a combination of Pepstats-SVM, SPMaP, and BLAST-KNN models contributed to the literature [21]. The SVM structure was used

in yet another gene ontology-based model, and a new approach was brought to the field [22]. These studies are mid-term studies, and since some of them require some additional experimentally acquired information besides lean protein sequences, current models that only work with protein sequences and do not require any additional information were needed [15].

The model working with Keras embeddings, ProtVec vector representation, and SVM using the Swiss-Prot dataset from studies conducted with arrays only has a success rate of 93% [23]. Again with the same dataset, Keras embeddings achieved success on a scale of 81.2% to 91.24% with a vector representation tool like ProtVec, model variations prepared with LSTM and CNN [16]. Another study used in enzyme research, using a pre-trained model with the Swiss-Prot dataset, achieved 97% success with the support of CNN and LSTM networks [15]. In this study, the sequences were divided into 3-character parts and processed just like the sentences and words of a text [23]. In another study of transport proteins, an accuracy rate of 85.8% was achieved with PSSMs, CNN, and GRU models [24]. In a study prepared with several different artificial neural network models such as KNN, Naive Bayes, and SVM, classification was made on the mouse protein dataset with Down syndrome [25]. In the study on urease activity in full-fat soybean production, a model was prepared with CNN-LSTM networks, and 96.57% and 90.29% successes were obtained according to the number of classes [26]. In a study prepared for protein homology detection, preprocessing with single-hot coding and model preparation with bidirectional LSTM were performed, and the model achieved 97% success [27]. In another study, the Pfam seed dataset was used, and code dictionary and LSTM were used. In accordance with the nature of the code dictionary structure, a number was assigned for each amino acid in the sequences, and analyzes were made on the digitized sequences [27], [28]. The 3-layer LSTM model used in a transfer learning-based study has brought 85% success [29]. In another protein classifier, the power of residual blocks of the CNN-based ResNet architecture was utilized, and a success rate of 93.7% was achieved [30]. The studies given above have demonstrated the importance of artificial intelligence in studies such as the classification of proteins, detection of binding sites, and detection of proteins from genes [31]–[33].

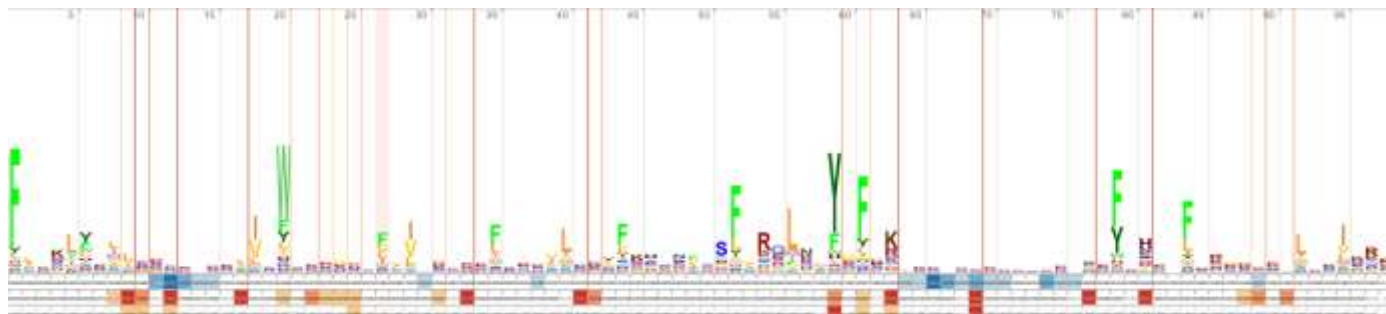


Fig. 2. HSF TF protein family motif structure [34].

III. MATERIALS AND METHODS

Transcription factor proteins in primates, including humans and various monkey species, were obtained from fragmented and

scattered data from the Animal Transcription Factor Database (AnimalTFDB), open to researchers. AnimalTFDB, with its current version, is a database containing TFs and their

cofactors, various annotations and binding domains, and subsections of these TFs in 183 animal genomes. Various analysis and alignment tools are also included in AnimalTFDB. It also includes additional information such as gene expression, post-translational modification and mutation information, and paralog and ortholog information [35].

A. Representation of Protein Sequences

Various proteins are formed by the sequential arrangement of amino acids according to the structural information [36]. This difference in structural information creates the unique structures of different proteins and, thus, different 3D structures [3]. Although each protein formed is different, they form different protein families together with the proteins with which they are structurally common. This structural commonality is called motifs in sequences. Sequences with identical motifs are proteins of the same family [37]. Figure 2 shows an example of a motif belonging to the Heat Shock Factor (HSF) TF protein family.

The letters representing amino acids in the motif in Figure 2 represent the probability of having the amino acid at the letter's position—the larger the representation, the more likely that amino acids will be found at the location. In addition to this information, when the motif is examined, it is seen that there may be many different amino acids in each position of the motif. For this reason, it is almost impossible to know which family each protein belongs to by examining the sequence character representations by human manually. For this reason, biological experiments or various artificial intelligence applications are some of the solutions. Considering that biological experiments are also a long time and costly, artificial intelligence applications stand out as one of the most effective solutions. Figure 3 shows a raw sample sequence of Homo sapiens, downloaded from AnimalTFDB [35].

```
>ENSP00000306549.3 ENSG00000169953 HSFY2 HSF
MAHVSSETQDVSPKDELTAESTRSPICEHTFPGSDLRSMIEEHAFQVLSQGSLLSESP
SYTVCVSEPKDDDFLSLNFPRKLNKIVESDQFKSISWDENGTCIVINEELFKKEILETK
APYRIFQTDIAIKSFVRQLNLVYFSKIQQNFQSAFLATFLSEKESVLSKLFYFYNPNF
KRGYPQLLVYKRRIGVKNASPISTLFNEDFNKKHFRAGANMENHNSALAAEASEESLFS
ASKNLMPLTRESSVRQIIANSSVIPRSGFPPSPSTSVGPSEQIATDQHAILNLQTTIH
MHSSTYMQARGHIVNFITTTTSQYHIISPLQNGYGLTVEPSAVPTRYPLVSVNEAPYR
NMLPAGNPWLQMPITADRSAAPHSRLALQPSPLDKYHPNYN
```

Fig. 3. A raw sample sequence of Homo sapiens from AnimalTFDB [35].

In the representation in Figure 2, the character ">" denotes the beginning of each new sequence. The first line contains the sequence Protein ID, Ensembl ID, gene symbol, and protein family information, respectively. From the second line to the next ">" character, there is the protein sequence. These files for each species are stored separately in AnimalTFDB in FASTA format.

These files in FASTA format, downloaded separately for all primates from AnimalTFDB, are created with the prepared Python scripts and replication operations, with a single line containing only protein families and sequences in a single line, with each protein family - protein sequence pair represented on a single line, containing all primates. combined in a tab-delimited text (.txt) file. As a result, a data set consisting of 72 different TF protein families belonging to 24 primate species

and containing 36242 sequences was created. Figure 4 shows a sample part of this dataset.

```
family sequence
Hsfreobox MDTSRPGAFVLSSAPLAALHMAENKSTLFPYALGQPAQKAPALGDLQALPLGTPHGISDILGRPYTA
AGDGLLGLPRLNQLASSAGVYFQPAARVARDYPLAELPQHPPIFWDVYQGAQWRDPLACAPAGDVLDDKDKKHSRPT
FSGQIFALKTEFTQTKYLAGPERARLAYSLQMTESQVKNYQRRRTKMKRHAEMASAKKKQSDAEKLVKGGSDAEDD
EYNPLDPSNDEKISRLKXKHKPSNLALVSPCGGAGDAL
zf-CZH2 MEIKLLPARGTLQGGGGSLPAGGGRVHRSDPFFAGQVPTRRLLLRGPDQGGPGRREEAFTA5R6GPGSL
APRPDQGGDDFFLVLLDPVGGVETAGSQAAGPVLREEAAGPQLQGGSSANPAGRPALGPRCLSAVSTPAPISAPF
PAAAFAGTVTIHQDQLLRFENGVLTLATPPHAMDPAAPAGDQDLVAPQAGFPAAAGSDCPELPPDLLAESAEPAF
PAPEEAEEDPAALGPRGRLGSAPOVVLVLCPEAGCQSTFAKHKQLKVVLLTHSSQQGRPFKCPLOGCOMFTTYSYKLRH
LQSHDKLRPFQCPAEGCGKSFITVYNLKAHMKGHEQENSFKCEVCEESFPTQAKLSAHQRSHFERPQCAFSGCKKFTIT
V5ALFSHNRHFREQLF5CSFPGCSKQYDKACRLX3HLRSHTEGRRFLCDFGGCWNFTSMKLLRHKRKHDDRRFTCPV
EGCGKSFTRAEHLKQHSITHLGTRKPFVCPVEGCCARFASRSLYHSHKHLQDQDVTWKSRCPISTCNKLFKSHMKNTHMAK
RHKVQDQLLAQLEAANSITPSSSELTSQGGNDLSEAEIVSLFSDVDPGTSAAVLDLALVNSGILTIDVASVSSLTAGNLANN
MNSVGGAVDPPALMATSDPPQLDLSLFFGTAAATGQDQSLDMVESSVTVGPLGSLGSLAVKNSPEPQALTPSSKLTVD
DALTPSSLTENSVELLTPTKAEVNHVPSDFGREGETQGFNAGNHSQSKTDLVTVTGSCSFLY
```

Fig. 4. A sample part from the prepared data set.

B. Data Preprocessing Techniques

The structures in the form of "MDTSRPGAFVLSSAPLA...", which is the representation of proteins in letters in character representation, is a meaningful structure for researchers in biology and bioinformatics. However, computers can only evaluate these sequences as words or sentences, and it is not possible for them to derive a direct meaning. Just as preprocessing is done on images, such as reducing size, changing color values, and extracting images locally, various operations, such as trimming sentences and texts, clipping suffixes, and adjusting sentences to equal length, are performed in natural language processing studies. Classification of protein sequences, which is a sub-branch of natural language processing studies, also requires a similar set of preprocessing steps [37].

In this study, a series of preprocessing steps, such as adjusting the protein sequences to a fixed length, shortening the long sequences, extending the short sequences by filling from the end, and preparing the embeddings from the sequences, were applied to the prepared data set.

Protein sequences differ from the data used in other natural language processing studies. Normally, sentences in texts consist of words, while protein sequences, each of which can be evaluated as a sentence, are in one piece. For this reason, previous studies have been done in the literature on the representation of each amino acid of the sequence with a number [28], [38]. However, suppose protein sequences are preprocessed at the amino acid level. In that case, achieving the highest desired success will be difficult since the relationships between these amino acids, that is, the relationship of each amino acid with the previous and next amino acid, cannot be revealed.

The probability that amino acids or amino acid groups come before or after each other, that is, their proximity to their location, is essential for capturing motifs and successful protein classification, just as knowing the similarities of words in natural language processing studies [39]. For this reason, it is necessary to know the affinity between these words (k-mers) by separating the protein sequences, which are one piece, into small amino acid groups, that is, words (k-mers). In addition, a preprocessing process that will provide high success is carried out with vector assignments and sequence digitization to be made according to this proximity information. When looking at the literature, the lengths of these amino acid groups are usually chosen in the range of 3 to 6 characters [40], [41].

After the sequences are divided into k-mers, it is necessary to analyze the closeness of these k-mers with each other. To avoid any k-mers combinations on all sequences and reveal different affinities, each sequence should be divided into k-mers 3 times by shifting one character from the first character 3 times. In this way, many k-mers are obtained, just as if 3 different sequences were processed [23]. In addition, the relationship between amino acids is confirmed 3 times. An example sequence-k-mer conversion is given in Figure 5.

EXAMPLE SEQUENCE

MDTSRPGAFVLSSAPLAAALHNMAEMKT...

WORDS (K-MERS)

MDT, SRP, GAF, VLS, SAP, LAA, LHN, MAE, ...
DTS, RPG, AFV, LSS, APL, AAL, HNM, AEM, ...
TSR, PGA, FVL, SSA, PLA, ALH, NMA, EMK, ...

Fig. 5. An example sequence-k-mer conversion.

If sequences are split into words with more characters (e.g., 5-mer, etc.) instead of 3-mers, there will be fewer word combinations as well as splits in the vector representations of motifs as fewer words will remain in each motif. These divisions will cause the motifs to be partially lost and the model's success to decrease.

In order to analyze the proximity of words in sequences divided into k-mers and digitize the sequences, vector assignments to these k-mers should be done with the Word2Vec model. Word2Vec is an unsupervised neural network model that represents words in vector space. According to the prepared vocabulary, the words in the sequences are given points according to their similarity, and their probability of coming before or after each other is determined. Word2Vec has two different architectures, Continuous Bag-of-Words (CBOW) and Continuous Skip-Gram; although both of these architectures do the same job, there is a fundamental difference between them. CBOW determines the word in the center according to the previous and next word as much as the specified window size. Skip-Gram, on the other hand, predicts the previous and following words by the window size according to the central word [42].

In the tests performed in this study, 6 was chosen for the CBOW architecture and window size for representation. The vector size in which each k-mer will be represented in the model has been determined as the default value of 300. Figure 6 shows the distribution of k-mer affinities in the vector space at window size 6 of the Word2Vec model.

Just as the resolution of all images is determined as the same in deep learning models working with images, the same length of sequences in models working with protein sequences in text structure is a factor that increases success. However, choosing a shorter than required length while bringing the sequences to a fixed length may cause partial or complete loss of protein motifs and, thus, classification errors. The selection made longer than necessary will cause a waste of resources and time, as well as classification errors, since the processing will be done with strings filled with too many characters. For this reason, it is necessary to analyze the data set well and choose the right

length. Figure 7 shows the distribution of sequences in the data set by length.

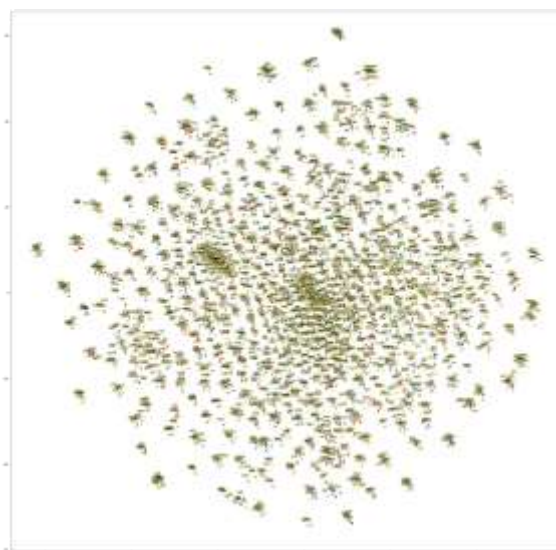


Fig. 6. Proximity representation of the Word2Vec model with 6 window sizes.

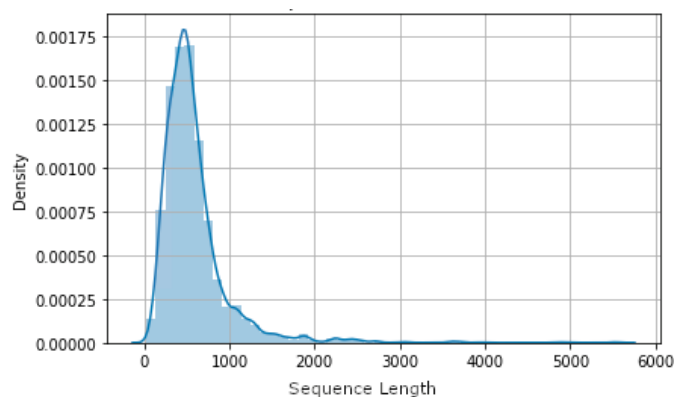


Fig. 7. Distribution of sequences in the data set according to their lengths.

Another purpose of separating sequences into words is to get shorter sequences from longer sequences. Looking at Figure 5, it can be seen that the majority of sequences are 400 to 600 in length. In this case, choosing the length of the arrays within this range will give the best results. Considering that the sequences are converted to 3-mers and digitized with vectors, it can be seen that synchronizing the sequences in their new form with 200 characters in length means equalizing the raw sequences in 600 characters. Sequences divided into K-mers are shortened to 200 words, long sequences are shortened from the end, and short sequences are extended to 200 words by adding the required number of 0s at the end.

The digitized and equal-length sequences are now ready to be given to the input embedding layer of the prepared deep learning model. In addition, the proximity information generated by training the Word2Vec model is also given to the "weights" parameter of the embedding layer so that the model starts training with these closeness values instead of random weights, which is aimed to perform a more successful training in a shorter time.

C. Long Short-Term Memory and Gated Recurrent Unit

LSTM is a new generation network developed against the problem that RNN forgets long-term information and dependencies [43]. As a solution to the continuous increase in gradients in RNN, a forget gate was added to each unit of the network, so that the network can decide which information to forget and which to keep [43], [44]. This structure uses tanh and sigmoid activation functions [45]. Similarly, GRU is a network designed to keep long and short-term dependencies in its memory. Although the internal structure of the units is very similar to the LSTM units, LSTM has input, output, and forget gates, while GRU has input and output gates. In GRU, the forget function is done with a forget key [46]. With all these functions, LSTM and GRU can decide which long- and short-term dependencies are remembered and which are forgotten by controlling the flow of information [47]. Although the GRU has almost the same level of success as the LSTM, it has been observed that the GRU mostly completes the training process faster than the LSTM because the GRU has 2 gates instead of 3 in the experiments.

D. Performance Metrics

The success of the models prepared in classification problems and other studies should be determined by various criteria. One of the most commonly used tools for determining success is the confusion matrix [48], and the accuracy, sensitivity, specificity, f-score, and average-precision values are calculated from this matrix [49]. In addition, these values will be used in drawing the Receiver Operating Characteristic (ROC) Curve [50]. While developing the models, the data sets are divided into parts in certain proportions and training, validation and testing steps are performed. Formulas for performance metrics are given in Equation 1-6.

$$\text{Accuracy (Acc)} = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$\text{Sensitivity (Se)} = \frac{TP}{TP + FN} \quad (2)$$

$$\text{Specificity (Sp)} = \frac{TN}{TN + FP} \quad (3)$$

$$\text{Precision (Pre)} = \frac{TP}{TP + FP} \quad (4)$$

$$F - \text{score} = \frac{2 * TP}{2 * TP + FP + FN} \quad (5)$$

$$\text{Average - Precision (AP)} = \frac{TP}{TP + TN} \quad (6)$$

While developing the models, the data sets are divided into parts in certain proportions, and training, validation, and testing steps are performed. In K-fold cross-validation, the data set is divided into k pieces, and k-1 pieces are used for training and 1 for testing at each step. In this way, one of the most accurate evaluations is performed as the models are trained and tested with different data many times [51], [52].

IV. RESULTS

All training and testing processes of the proposed model, the design details given in Table I, were carried out in Google Colab Pro with version 3.8 of the Python programming language. The primates TF protein dataset used in the training and testing of the model consists of 36243 protein sequences. Of these sequences, 70%, i.e., 25370, were used for training, 15% (5437) for validation, and 15% (5436) for testing.

Samples from all 72 classes are also available in all train, validation, and test datasets. This partitioning is done completely randomly via Python's Sklearn library. In order to achieve the highest success in training the model, an early stop function was added [38], and the fault tolerance was determined as 3 epochs. In order to determine the amount of data to be processed in unit time, the value of 256 was given to the batch_size parameter, taking into account the sequence lengths.

Two different models were studied for the effect of model layers and the number of units in each layer on model success and running time. The structures of these models are given in Table I.

TABLE I
BASIC STRUCTURES OF THE IMPLEMENTED MODELS

Model No	Model Structure
M1	Bidirectional LSTM (128), Bidirectional LSTM (256), Bidirectional GRU (256)
M2	Bidirectional LSTM (128), Bidirectional LSTM (128), Bidirectional LSTM (128), Bidirectional GRU (128)

The accuracy, precision, recall, f-score, and train time results of these two implemented models are given in Table II.

TABLE II
TEST RESULTS IMPLEMENTED MODELS

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
M1	96.85	97.77	93.67	95.47
M2	97.96	97.55	95.26	96.22

When Table II is examined, it can be concluded that the model numbered M2 is a more successful classification. This result also indicates that models with more tiers and fewer units per tier perform better than models with fewer but more units per tier. The 5-fold cross-validation results of these two models are given in Table III.

TABLE III
5-FOLD CROSS VALIDATION RESULTS OF IMPLEMENTED MODELS

Model No	5-Fold Cross Validation Accuracy Results (%)
M1	96.67
M2	97.42

Table III also supports the result in Table II, showing that multilayer models with fewer units are more successful in this study. For this reason, in the continuation of the article, the model numbered M2 was determined as the proposed model. The reason for choosing 5-fold in this study is that the data set size is not very large.

In the proposed model M2, 3 layers of bidirectional LSTM and 1 layer of bidirectional GRU were used after the embedding layer, where sequences represented by vectors using Word2Vec and these vector affinity values were given as initial weights. 128 units were used in LSTM layers. A 0.3 dropout [53] layer has been added to prevent overfitting. The network of the next 128-unit bidirectional layer is determined as a bidirectional GRU for the model to complete a faster training process. To prevent overfitting, a dropout [53] layer of 0.3 was added, and then the flatten function was used. At the output of this layer, first, a dense layer with a value of 256, then a dropout [53] layer of 0.4 to prevent overfitting, and the model was finalized with

a dense layer of the class number to classification. The details of the model are shown in Table IV.

TABLE IV
DETAILED STRUCTURE OF THE PROPOSED MODEL

Layer (type)	Output Shape	Param #
Embedding	(None, 200, 300)	2540100
Bidirectional LSTM (128)	(None, 200, 256)	439296
Bidirectional LSTM (128)	(None, 200, 256)	394240
Bidirectional LSTM (128)	(None, 200, 256)	394240
Dropout (0.3)	(None, 200, 256)	0
Bidirectional GRU (128)	(None, 200, 256)	296448
Dropout (0.3)	(None, 200, 256)	0
Flatten	(None, 51200)	0
Dense (256)	(None, 256)	13107456
Dropout (0.4)	(None, 256)	0
Dense (72) (Classification)	(None, 72)	18504

The default learning rate value of 0.01 was used in the LSTM and GRU layers of the model. ReLU [54] was used as the activation function in the first dense layer, and Softmax [55] was used in the second because it was the classification layer. Since the data set is multi-class, Categorical Crossentropy [56] was used as the loss function, and Adam [57] was used as the optimizer.

The train and validation graphs for accuracy and loss of the model M1 are given in Figure 8, and the train and validation accuracy and loss graphs of the proposed model M2 are given in Figure 9.

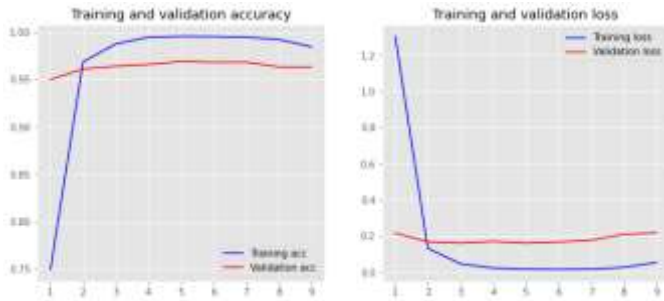


Fig. 8. Accuracy and loss graphs for train and validation of the model M1.

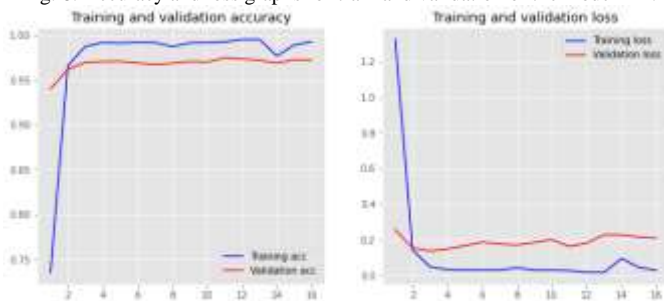


Fig. 9. Accuracy and loss graphs for train and validation of the proposed model M2.

The test results of the proposed model, the multi-layered structure of the model, which is prepared with layers with fewer units, increases the model's success and greatly shortens the training time. Figure 10 shows the ROC graph of the model M1. Figure 11 shows the ROC graph of the proposed model M2.

Some extension of Receiver operating characteristic to multi-class

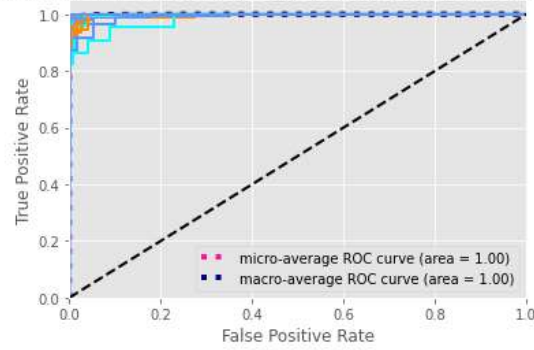


Fig. 10. ROC curve graph of the model M1.

Some extension of Receiver operating characteristic to multi-class

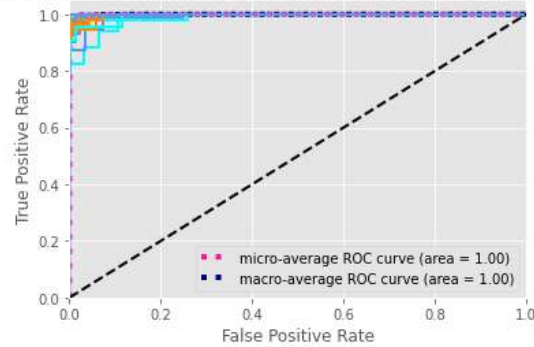


Fig. 11. ROC curve graph of the proposed model M2.

All the curves of the ROC graph in Figure 11 are gathered in the upper left corner, and their values are close to 1.0. This graph shows that the proposed model classifies 72 different classes with high success.

V. DISCUSSIONS

Along with the developments in computer science, computer science-supported studies have started to be carried out in bioinformatics. Statistical-based studies such as HMM can be among the early-mid-term examples of bioinformatics. Then, artificial neural networks and machine learning studies were used in bioinformatics. Today, various deep learning-based studies are used in bioinformatics and various biological and genetic data. In this study, transcription factor proteins in primates were classified by preparing a hybrid model combining the light and fast structure of RNN-based LSTM and GRU networks with the success of the Word2Vec model in preprocessing and similarity analysis. The data set was automatically compiled from public bioinformatics data and prepared originally. In this way, it was ensured that a deficiency in the literature was completed and an important and new contribution to the literature was made. In this process, a proximity analysis was performed by first dividing the sequences into small protein groups, namely k-mers. In this way, the model, which started its education ahead of similar models, achieved a more successful result in less epochs. Then, 3 LSTM layers and 1 GRU layer with fewer units per layer used were also used with lighter weight and more layers, and this design also increased the model's success while reducing the training time. In addition to all these contributions, the prepared

primates TF protein dataset was prepared and stored with the newly trained Word2Vec model and model weights for use in future studies.

Although there is no direct study on the classification of TF proteins in primates when the literature is scanned, the comparison with studies using similar data for the evaluation of the study will highlight the success of this study. Table V shows the comparison of the proposed model with some studies in the literature.

TABLE V
COMPARISON OF THE PROPOSED MODEL WITH VARIOUS STUDIES
IN THE LITERATURE

Authors	Dataset	Accuracy (%)
Le et al, 2019 [24]	Vesicular transport proteins	85.8
Belzen et al., 2019 [30]	CAFA3	93.7
Bileschi et al., 2022 [28]	Pfam Seed	95.8
Proposed model (M2)	Primates TF Proteins (original)	97.96

VI. CONCLUSION

Within the scope of this study, the classification of primates TF proteins, which has no precedent in the literature, was carried out. The model completed the classification process with an accuracy of 97.96% and achieved a high success rate in the literature. However, although plant TF proteins have been studied before in the literature [38], there are still deficiencies in the literature regarding the classification of TF proteins of other animals and organisms besides primates. In future studies, more successful classification of the TF proteins of primates classified in this study and the development of preprocessing steps will be provided, and it will be possible to study on the classification of TF proteins in other kingdoms.

REFERENCES

- [1] J. J. Shu, "A new integrated symmetrical table for genetic codes," *Biosystems*, vol. 151, pp. 21–26, Jan. 2017, doi: 10.1016/J.BIOSYSTEMS.2016.11.004.
- [2] J. D. WATSON and F. H. C. CRICK, "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid," *Nature*, vol. 171, no. 4356, pp. 737–738, Apr. 1953, doi: 10.1038/171737a0.
- [3] D. R. Ferrier, "Protein Yapısı ve İşlevi," in *Lippincott Biyokimya: Görsel Anlatımlı Çalışma Kitapları*, B. A. Jameson, Ed. İstanbul: Nobel Tıp Kitapevleri, 2019, pp. 1–68.
- [4] Pfam, "Family: HLH (PF00010)." <http://pfam.xfam.org/family/pf00010> (accessed Feb. 02, 2019).
- [5] T. Kaplan and M. D. Biggin, "Quantitative Models of the Mechanisms that Control Genome-Wide Patterns of Animal Transcription Factor Binding," *Methods Cell Biol.*, vol. 110, pp. 263–283, Jan. 2012, doi: 10.1016/B978-0-12-388403-9.00011-4.
- [6] D. S. Latchman, "Transcription factors: an overview Function of transcription factors," *Int. J. Exp. Path.*, vol. 74, pp. 417–422, 1993.
- [7] M. Karin, "Too many transcription factors: positive and negative interactions," *New Biol.*, vol. 2, no. 2, pp. 126–131, 1990.
- [8] D. S. Latchman, "Transcription factors: An overview," *Int J Biochem Cell Biol.*, vol. 29, no. 12, pp. 1305–1312, Dec. 1997, doi: 10.1016/S1357-2725(97)00085-X.
- [9] D. Petrey and B. Honig, "Is protein classification necessary? Toward alternative approaches to function annotation," *Curr Opin Struct Biol.*, vol. 19, no. 3, pp. 363–368, Jun. 2009, doi: 10.1016/J.SBI.2009.02.001.
- [10] P. Baldi and S. Brunak, *Bioinformatics, Second Edition: The Machine Learning Approach*. Cambridge: MIT Press, 2001.
- [11] S. R. Eddy, "Hidden Markov models," *Curr Opin Struct Biol.*, vol. 6, no. 3, pp. 361–365, Jun. 1996, doi: 10.1016/S0959-440X(96)80056-X.
- [12] M. M. Gromiha, "Protein Sequence Analysis," *Protein Bioinformatics*, pp. 29–62, Jan. 2010, doi: 10.1016/B978-8-1312-2297-3.50002-3.
- [13] 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, Oct. 1990, doi: 10.1016/S0022-2836(05)80360-2.
- [14] M. N. Price *et al.*, "Mutant phenotypes for thousands of bacterial genes of unknown function," *Nature*, vol. 557, no. 7706, p. 503–509, May 2018, doi: 10.1038/s41586-018-0124-0.
- [15] N. Strodthoff, P. Wagner, M. Wenzel, and W. Samek, "UDSMProt: universal deep sequence models for protein classification," *Bioinformatics*, vol. 36, no. 8, pp. 2401–2409, Apr. 2020, doi: 10.1093/bioinformatics/btaa003.
- [16] K. S. Naveenkumar, B. R. Mohammed Harun, R. Vinayakumar, and K. P. Soman, "Protein Family Classification using Deep Learning," *bioRxiv*, p. 414128, Jan. 2018, doi: 10.1101/414128.
- [17] X. Du, Y. Cai, S. Wang, and L. Zhang, "Overview of deep learning," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 159–164. doi: 10.1109/YAC.2016.7804882.
- [18] M. Huerta, F. Haseltine, Y. Liu, G. Downing, and B. Seto, "NIH working definition of bioinformatics and computational biology," Jul. 2000.
- [19] Q. Gong, W. Ning, and W. Tian, "GoFDR: A sequence alignment based method for predicting protein functions," *Methods*, vol. 93, pp. 3–14, Jan. 2016, doi: 10.1016/J.YMETH.2015.08.009.
- [20] H. bin Shen and K. C. Chou, "EzyPred: A top-down approach for predicting enzyme functional classes and subclasses," *Biochem Biophys Res Commun.*, vol. 364, no. 1, pp. 53–59, Dec. 2007, doi: 10.1016/J.BBRC.2007.09.098.
- [21] A. Dalkiran, A. S. Rifaioglu, M. J. Martin, R. Cetin-Atalay, V. Atalay, and T. Doğan, "ECPred: a tool for the prediction of the enzymatic functions of protein sequences based on the EC nomenclature," *BMC Bioinformatics*, vol. 19, no. 1, p. 334, 2018, doi: 10.1186/s12859-018-2368-y.
- [22] D. Cozzetto, F. Minneci, H. Curren, and D. T. Jones, "FFPred 3: feature-based function prediction for all Gene Ontology domains," *Sci Rep*, vol. 6, no. 1, p. 31865, 2016, doi: 10.1038/srep31865.
- [23] E. Asgari and M. R. K. Mofrad, "Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics," *PLoS One*, vol. 10, no. 11, Nov. 2015.
- [24] N. Q. K. Le, E. K. Y. Yapp, N. Nagasundaram, M. C. H. Chua, and H. Y. Yeh, "Computational identification of vesicular transport proteins from sequences using deep gated recurrent units architecture," *Comput Struct Biotechnol J.*, vol. 17, pp. 1245–1254, Jan. 2019, doi: 10.1016/J.CSBJ.2019.09.005.
- [25] F. G. Furat and T. Ibriki, "Classification of Down Syndrome of Mice Protein Dataset on MongoDB Database," *Balkan Journal of Electrical and Computer Engineering*, pp. 44–49, Apr. 2018, doi: 10.17694/bajece.419553.
- [26] İ. ÖZER, "Classification of Urease Activity in Full-Fat Soybean Production by Extrusion Using Machine Learning Algorithms," *Balkan Journal of Electrical and Computer Engineering*, Aug. 2021, doi: 10.17694/bajece.941007.
- [27] S. Li, J. Chen, and B. Liu, "Protein remote homology detection based on bidirectional long short-term memory," *BMC Bioinformatics*, vol. 18, no. 1, p. 443, 2017, doi: 10.1186/s12859-017-1842-2.
- [28] M. L. Bileschi *et al.*, "Using deep learning to annotate the protein universe," *Nat Biotechnol.*, vol. 40, no. 6, pp. 932–937, Jun. 2022, doi: 10.1038/s41587-021-01179-w.
- [29] R. Rao *et al.*, "Evaluating Protein Transfer Learning with TAPE," *Adv Neural Inf Process Syst.*, vol. 32, pp. 9689–9701, Dec. 2019, [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/33390682>
- [30] J. Upmeyer zu Belzen *et al.*, "Leveraging implicit knowledge in neural networks for functional dissection and engineering of proteins," *Nat Mach Intell.*, vol. 1, no. 5, pp. 225–235, 2019, doi: 10.1038/s42256-019-0049-9.
- [31] M. Torrisi, G. Pollastri, and Q. Le, "Deep learning methods in protein structure prediction," *Comput Struct Biotechnol J.*, vol. 18, pp. 1301–1310, Jan. 2020, doi: 10.1016/j.csbj.2019.12.011.
- [32] S. Lim *et al.*, "A review on compound-protein interaction prediction methods: Data, format, representation and model," *Comput Struct Biotechnol J.*, vol. 19, pp. 1541–1556, Jan. 2021, doi: 10.1016/J.CSBJ.2021.03.004.
- [33] C. Gustafsson, J. Minshull, S. Govindarajan, J. Ness, A. Villalobos, and M. Welch, "Engineering genes for predictable protein expression," *Protein Expr Purif.*, vol. 83, no. 1, pp. 37–46, May 2012, doi: 10.1016/J.JEP.2012.02.013.

- [34] Pfam, “HSF-type DNA-binding PF00447,” <https://www.ebi.ac.uk/interpro/entry/pfam/PF00447/logo/> (accessed Sep. 11, 2022).
- [35] H. Hu, Y.-R. Miao, L.-H. Jia, Q.-Y. Yu, Q. Zhang, and A.-Y. Guo, “AnimalTFDB 3.0: a comprehensive resource for annotation and prediction of animal transcription factors,” *Nucleic Acids Res*, vol. 47, no. D1, pp. D33–D38, Jan. 2019, doi: 10.1093/nar/gky822.
- [36] IUPAC-IUB Comm. on Biochem. Nomenclature, “A one-letter notation for amino acid sequences. Tentative rules,” *Biochemistry*, vol. 7, no. 8, pp. 2703–2705, Aug. 1968, doi: 10.1021/bi00848a001.
- [37] D. Ofer, N. Brandes, and M. Linial, “The language of proteins: NLP, machine learning & protein sequences,” *Comput Struct Biotechnol J*, vol. 19, pp. 1750–1758, Jan. 2021, doi: 10.1016/J.CSBJ.2021.03.022.
- [38] A. B. Oncul, Y. Celik, N. M. Unel, and M. C. Baloglu, “Bhlhdb: A next generation database of basic helix loop helix transcription factors based on deep learning model,” *J Bioinform Comput Biol*, Jun. 2022, doi: 10.1142/S0219720022500147.
- [39] B. Ay Karakuş, M. Talo, İ. R. Hallaç, and G. Aydin, “Evaluating deep learning models for sentiment classification,” *Concurr Comput*, vol. 30, no. 21, pp. 1–14, Nov. 2018, doi: 10.1002/cpe.4783.
- [40] J. K. Vries, X. Liu, and I. Bahar, “The relationship between N-gram patterns and protein secondary structure,” *Proteins: Structure, Function, and Bioinformatics*, vol. 68, no. 4, pp. 830–838, May 2007, doi: 10.1002/prot.21480.
- [41] J. K. Vries and X. Liu, “Subfamily specific conservation profiles for proteins based on n-gram patterns,” *BMC Bioinformatics*, vol. 9, no. 1, p. 72, Dec. 2008, doi: 10.1186/1471-2105-9-72.
- [42] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Jan. 2013.
- [43] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A Search Space Odyssey,” *IEEE Trans Neural Netw Learn Syst*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.
- [44] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [45] G. van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artif Intell Rev*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.
- [46] Y. Gao and D. Glowacka, “Deep Gate Recurrent Neural Network,” in *Proceedings of The 8th Asian Conference on Machine Learning*, Jul. 2016, vol. 63, pp. 350–365. [Online]. Available: <https://proceedings.mlr.press/v63/gao30.html>
- [47] A. Şeker, B. Diri, and H. H. Balık, “Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme,” *Gazi Mühendislik Bilimleri Dergisi*, vol. 3, no. 3, pp. 47–64, Nov. 2017.
- [48] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. doi: 10.1007/978-0-387-30164-8.
- [49] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” *Pattern Recognit*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/J.PATCOG.2019.02.023.
- [50] B. Ozenne, F. Subtil, and D. Maucort-Boulch, “The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases,” *J Clin Epidemiol*, vol. 68, no. 8, pp. 855–859, Aug. 2015, doi: 10.1016/J.JCLINEPI.2015.02.010.
- [51] A. Rohani, M. Taki, and M. Abdollahpour, “A novel soft computing model (Gaussian process regression with K-fold cross validation) for daily and monthly solar radiation forecasting (Part: I),” *Renew Energy*, vol. 115, pp. 411–422, Jan. 2018, doi: 10.1016/j.renene.2017.08.061.
- [52] Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu, and J. Hu, “Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation,” *Comput Mater Sci*, vol. 171, p. 109203, Jan. 2020, doi: 10.1016/j.commatsci.2019.109203.
- [53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014, [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>
- [54] L. Parisi, D. Neagu, R. Ma, and F. Campean, “Quantum ReLU activation for Convolutional Neural Networks to improve diagnosis of Parkinson’s disease and COVID-19,” *Expert Syst Appl*, vol. 187, p. 115892, Jan. 2022, doi: 10.1016/j.eswa.2021.115892.
- [55] A. Basturk, M. E. Yuksei, H. Badem, and A. Caliskan, “Deep neural network based diagnosis system for melanoma skin cancer,” in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, May 2017, pp. 1–4. doi: 10.1109/SIU.2017.7960563.
- [56] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.
- [57] E. YAZAN and M. F. Talu, “Comparison of the stochastic gradient descent based optimization techniques,” in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sep. 2017, pp. 1–5. doi: 10.1109/IDAP.2017.8090299.

BIOGRAPHIES



ALİ BURAK ÖNCÜL Amasya, in 1991. He received the B.S. in 2013 and M.S. degrees in 2016 in computer engineering from Ondokuz Mayıs University, Samsun, and the Ph.D. degree in computer engineering from Karabük University, Karabük, in 2022.

Since 2015, he has been a Research Assistant with the Computer Engineering Department, Kastamonu University. His areas of study are GNU/Linux, operating systems, deep learning, bioinformatics, protein structure and classifications.