biomolecules

MDPI

Article

An Efficient Lightweight Hybrid Model with Attention Mechanism for Enhancer Sequence Recognition

Suliman Aladhadh [1,*], Saleh A. Almatroodi [2], Shabana Habib [1], Abdulatif Alabdulatif [3], Saeed Ullah Khattak [4] and Muhammad Islam [5]

[1] Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[2] Department of Medical Laboratories, College of Applied Medical Sciences, Qassim University, Buraydah 51452, Saudi Arabia
[3] Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[4] Centre of Biotechnology and Microbiology, University of Peshawar, Peshawar 25120, Pakistan
[5] Department of Electrical Engineering, College of Engineering and Information Technology, Onaizah Colleges, Onaizah 56447, Saudi Arabia
* Correspondence: s.aladhadh@qu.edu.sa

Abstract: Enhancers are sequences with short motifs that exhibit high positional variability and free scattering properties. Identification of these noncoding DNA fragments and their strength are extremely important because they play a key role in controlling gene regulation on a cellular basis. The identification of enhancers is more complex than that of other factors in the genome because they are freely scattered, and their location varies widely. In recent years, bioinformatics tools have enabled significant improvement in identifying this biological difficulty. Cell line-specific screening is not possible using these existing computational methods based solely on DNA sequences. DNA segment chromatin accessibility may provide useful information about its potential function in regulation, thereby identifying regulatory elements based on its chromatin accessibility. In chromatin, the entanglement structure allows positions far apart in the sequence to encounter each other, regardless of their proximity to the gene to be acted upon. Thus, identifying enhancers and assessing their strength is difficult and time-consuming. The goal of our work was to overcome these limitations by presenting a convolutional neural network (CNN) with attention-gated recurrent units (AttGRU) based on Deep Learning. It used a CNN and one-hot coding to build models, primarily to identify enhancers and secondarily to classify their strength. To test the performance of the proposed model, parallels were drawn between enhancer-CNNAttGRU and existing state-of-the-art methods to enable comparisons. The proposed model performed the best for predicting stage one and stage two enhancer sequences, as well as their strengths, in a cross-species analysis, achieving best accuracy values of 87.39% and 84.46%, respectively. Overall, the results showed that the proposed model provided comparable results to state-of-the-art models, highlighting its usefulness.

Keywords: deep learning; enhancer sequence; convolution neural network; sequential learning models; temporal attention mechanism

check for updates

Citation: Aladhadh, S.; Almatroodi, S.A.; Habib, S.; Alabdulatif, A.; Khattak, S.U.; Islam, M. An Efficient Lightweight Hybrid Model with Attention Mechanism for Enhancer Sequence Recognition. *Biomolecules* **2023**, *13*, 70. https://doi.org/10.3390/biom13010070

Academic Editor: William Cho

Received: 4 November 2022
Revised: 22 December 2022
Accepted: 26 December 2022
Published: 29 December 2022

1. Introduction

Transcriptomics describes enhancers [1] as DNA segments that target the control of gene expression for the production of proteins (activators) and RNA. These proteins are comprised of transcription factors, which are proteins that are involved in the transcription process in which DNA is transformed into RNA and vice versa [2]. It is possible for these to be located as far as 1 MBp away from the gene or even at different locations on the chromosome [3]. Identifying enhancers can be difficult, since they are present in perceptible genomic sections due to their vigorous nature. Over the past few years, a number of useful methods have been developed that have helped to overcome challenges associated with identifying enhancers at the genomic level. As a result of recent research, the presence of enhancers in vertebrates and mammals has been confirmed [4]. For example, genes upregulated by inflammatory bowel disease (IBD) contain highly enriched concentrations of IBD-associated SNPs, and the transcription factors bound to the promoters and enhancers of these genes are very similar to those binding to the genes of IBD [5].

There is still a significant amount of work to be done in order to identify enhancers and correlate them with human biology and disease on a global scale. Enhancers play a significant role in determining the function of many genes throughout the human genome. Furthermore, both prokaryotes and eukaryotes feature enhancers as part of their DNA. There is a certain sequence in DNA called a promoter; it is a slightly different piece of DNA where gene transcription begins [3]. The promoter is typically found at the beginning of a gene, but an enhancer is usually found at the end of a gene, or even on a chromosome that has no genes on it. It is extremely challenging to identify new enhancers when there is such a disparity in location between these various enhancers. In certain contemporary studies of alterations, it was demonstrated that enhancers are a large family of functional elements. These may be divided into several subgroups, whose targets undergo different types of biological activities, and regulatory effects based on their target mutations [6]. There is further evidence that genetic variation in enhancers is linked to an increased risk of diseases in humans, such as inflammatory bowel disease and a variety of cancers.

Recently, significant computational work was conducted in order to identify regulator enhancers using computational algorithms. This was done as part of efforts to save time and money; experimentation is time-consuming, expensive and not always effective. The outgrowth of biological data has become a major concern for computational researchers, as they now have high-profile computing assets, as well as sophisticated strategies with which to deal with it. Several computational prediction models to rapidly recognize enhancers in genes were developed in recent years as a result of the improvement of machine learning. These include Enhancer-LSTMAtt [7], CSI-ANN [8], EnhancerFinder [9], Chrome, GKM-SVM [10], DEEP [11], GenSVM [12], RFECS [13], EnhancerDBN [14], and BiRen [15]. Despite this, these methods merely act as a classification tool for enhancers that have been identified. Meanwhile, there are many different types of enhancers, including strong and weak enhancers, pent-up enhancers, and inactive enhancers. Enhancers are broad-ranging and comprise numerous subgroups of enhancers. There is a primary predictive model that relies solely on sequence data as a starting point for distinguishing enhancers and their quality as part of a prediction tool that can assist predictors in identifying enhancers and their quality. This model is known as iEnhancer-2L [3]. In order to identify enhancers and their strengths, and categorize them accordingly based on their strength, several accurate predictors have been proposed. These include iEnhancer-EL [16] and iEnhancer-2L [3], which can identify enhancers and their strengths, as well as classify them accordingly as strong or weak enhancers. According to Liu et al., iEnhancer-2L [3] proposed a methodology which employed a support vector machine (SVM) learning algorithm to incorporate operational changes. The PseKenny-Nucleotide Composition (PseKNC) is a secondary nucleotide composition scheme that was used in addition to pseudo-k-tuple nucleotide compositions as the sequence-encoding scheme in iEnhancer-2L. In 2018, a new improved version of iEnhancer-2L called iEnhancer-EL [16] was presented as a replacement for iEnhancer-2L [3]. The first stage of this model required a set of six classifiers, while the second stage required an ensemble of ten classifiers, which made it a complex model to implement. Several elementary classifiers [16] composed of three different feature categories—the PseKNC, the k-mers, and the subsequence profiles—were used to construct the crucial classifiers. These classifiers were assembled using many SVM-based elementary classifiers [16]. As much as machine learning-based methods like those mentioned above are capable of delivering good results, it has been shown that deep learning models produce better results without requiring a manual feature extraction operation. Furthermore, machine learning techniques for genomics analysis still require input features that are hand-designed by an individual and extracted from predetermined sequences of input data in order to be able to make decisions [17,18]. It should be noted, however, that convolutional neural networks (CNNs) are able to quickly extract substantial features from a variety of stages. The iEnhancer-EL method [16] is currently considered one of the best methods for classifying enhancers and their strength, but it is likely to inspire even better models, such as ones that use new encoding methods and learning algorithms. In order to increase the accuracy of predictions of enhancer strength, we proposed to use a two-stage framework, instead of relying on a single deep learning prediction framework, in the first stage aimed at classifying enhancers. This framework was dubbed iEnhancer-Deep. Despite the fact that the aforementioned experimental methods may be useful to some extent, at the time of this writing, there did not appear to be any unified standard for the identification of enhancers in biology. Furthermore, current empirical approaches are labor-intensive, time-consuming, and impractical, rendering them ineffective for application to all cell types at various stages of the cell cycle at the same time. The model used in this paper offered the use of One-Hot encoding as well as the use of CNNs to encode sequences. There was also an effort to determine whether the input sequences from the proposed model met the quality and strength criteria to be classified as enhancers during testing. They were sent to the secondary stage to determine this. In the case that the proposed sequence did not meet all of these criteria, the sequence was referred to as a non-enhancer sequence. The analysis in this study was based on Chou's five-step rule, which has been extensively used in recent studies [17–20]. As required, the following procedures were followed: (i) for the preparation and testing of the indicator, the large benchmark dataset must be assembled and analyzed; (ii) enhance the significance of genomic sequences by emphasizing a meaningful pattern during the extraction or determination; (iii) develop a classifier capable of identifying these sequences in an effective and accurate manner; (iv) perform various cross-validation techniques on the data; and (v) assemble a web server. To summarize, we presented a learning-based model that enabled accurate identification of enhancer sequences, their strength, and their evolutionary properties. Our article's novel contributions can be summarized as follows:

In the current state of enhancer sequence classification, many methods rely on manual features extraction to be effective. When it comes to analyzing pre-miRNAs, there are two major approaches; one focuses on their spatial structure, while the other focuses on their sequential structure. Both are ineffective. We developed hybrid architectures that combined the encoding and representation power of convolutional neural networks (CNN) with the ability to handle large DNA sequences and the ability to accurately identify enhancers based on DNA sequence alone—making them ideal for handling DNA sequences.

There are different ways in which nucleotides can be represented within a sequence of nucleotides, such as the assignment of labels and the encoding of those labels. We were able to convert these nucleotide positions into a numerical description with the help of efficient coding methods, which we could then use to illustrate the nucleotide positions within the sequence.

The framework was validated using several benchmark enhancer sequence datasets in order to achieve state-of-the-art results for accurate classification of enhancer sequences and their strengths and prove the validity of the framework.

This manuscript was organized in the following manner: Section 2 explains the materials and methods used to identify enhancer sequences in this manuscript. In Section 3, we discussed the implementation of the proposed model, the experimental results, and the evaluation of its performance. We then discussed the conclusions of our current research in Section 4.

2. Materials and Methods

This section was intended to provide a brief overview of the underlying architecture of the proposed model, as illustrated in Figure 1. First, we described the basic structure of the proposed model in order to facilitate a better understanding. Next, we discussed the details of the feature extraction method in our model, which represents the encoding technique and the backbone CNN as a feature vector. In the last step, we used attentional bidirectional GRUs to map long-range dependencies on arbitrary DNA sequence lengths and form fixed-length feature representations. Table 1 presents all the abbreviation used in this work with their description.

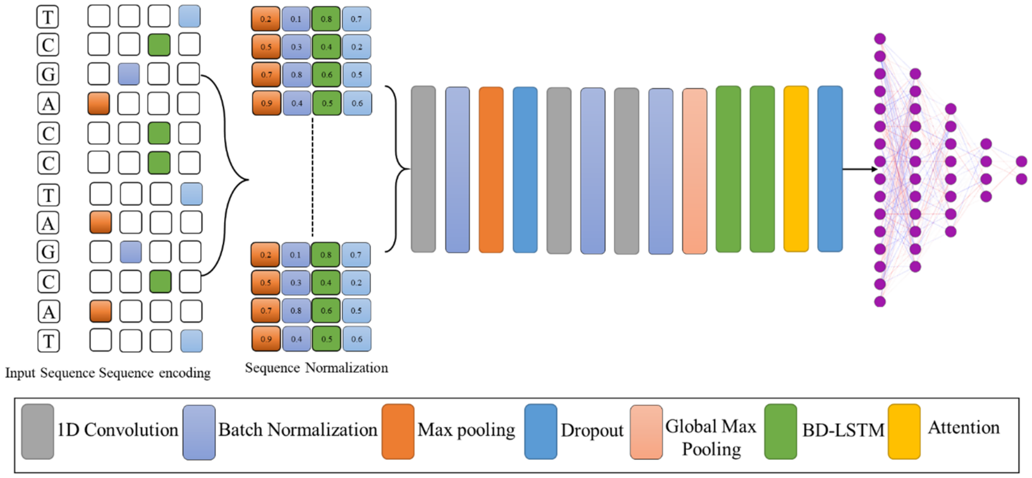


Figure 1. The proposed model for enhancer sequence classification and strength identification.

Table 1. The list of notations and their descriptions used in this research work.

Notation	Description
$\mathcal{D}$	Dataset
$\mathcal{D}^+$	Enhancer sequences
$\mathcal{D}^-$	Non-enhancer sequence
$\mathscr{r}_t$	Reset gate
$\mathscr{z}_t$	Update gate
$\mathscr{h}_t$	Hidden state
$\mathscr{a}_t$	Attention weights
$\mathbb{ACC}$	Accuracy
$\mathbb{SN}$	Sensitivity
$\mathbb{MCC}$	Matthews' Correlation Coefficient
$\mathbb{SP}$	Specificity
$\mathbb{TP}$	True Positive
$\mathbb{TN}$	True Negative
$\mathbb{FP}$	False Positive
$\mathbb{FN}$	False Negative

2.1. Dataset

This study utilized a dataset obtained from Liu et al. in their study [3]. Furthermore, this dataset was used in the development of iEnhancer-EL [16], iEnhancer-2L [3], and EnhancerPred [21]. There were nine different cell lines in this dataset. These were used for the extraction of enhancers, which were separated from short 200 bp clips of the same length and extracted as DNA groupings out of the DNA. A total of nine different cell lines were used in this study, including H1ES, K562, GM12878, HUVEC, HSMM, NHLF, NHEK, HepG2, and HMEC. We used CD-HIT software program [22] to prevent pairwise sequences with more than 20% of features in common that were present in each sequence from crossing. In the benchmark dataset, there were 1484 enhancers, of which 742 were strong enhancers, while the remaining 742 were weak enhancers—which is an increase from the baseline dataset. Thus, based on the information provided above, the benchmark dataset can be defined as follows:

$$\mathcal{D} = \mathcal{D}^{+} \cup \mathcal{D}^{-} \tag{1}$$

$$\mathcal{D}^{+} = \mathcal{D}^{+}_{Strong} \cup \mathcal{D}^{+}_{weak} \tag{2}$$

There are positive and negative sequences in the dataset, where $\mathcal{D}$ represents the overall number of sequences in the dataset. Set theory illustrates the concept of union through the use of the symbol $\cup$. An enhancer subset of $\mathcal{D}^{+}$ contained 1484 enhancer sequences, while non-enhancer subsets of $\mathcal{D}^{-}$ contained 1484 non-enhancer sequences. There were 1484 enhancers in the original set, divided into two parts: the strong enhancers, which constituted 752, and the weak enhancers, which comprised 742 enhancers. There are both $\mathcal{D}^{+}_{Strong}$ (strong enhancers) and $\mathcal{D}^{+}_{weak}$ (weak enhancers) enhancers within the nine tissues emphasized above; however, there was substantial variation between tissues for $\mathcal{D}^{+}$ weak (weak enhancers). As a result, human embryonic stem cells were used to develop weak enhancers to account for this. The training set provided us with the opportunity to build two different models for two different problems, much like other studies have done. The first model was used to identify enhancers in stage 1, while the second model was used to classify enhancers in stage 2. The training set was divided randomly into ten folds, utilizing stratified sampling for both layers, with a randomized distribution between each fold. We used each fold individually as a validation set, then used the remaining four folds as the training set and as a basis for the construction of a CNN model using the ten folds.

2.2. Sequence Encoding and Proposed Model

The proposed method involved taking DNA segments with a size of 200 bp and converting them into a number sequence in which $\mathcal{N}$ represents the character of the unknown nucleotide. The convolution module and the Gate Recurrent Units (GRU) with attention mechanism were used to process the sequences of numbers that were entered. There were two main modules in the LSTM: a convolution module that used mainly 1D CNNs, and an attention module that used mostly feed-forward LSTMs and bi-LSTMs [23,24]. After concatenating the outputs of the two modules, the outputs were incorporated into the fully connected layer of the system. The final layer of the network followed after the fully connected layer. This latter contained two neuronal structures that represented the probabilities of belonging to enhancer layers. With a threshold of 0.5, it was predicted that a positive input would generate an output above 0.5, whereas a negative input would cause output above 0.5 to indicate the opposite.

2.3. Features Extraction

Convolutional neural network architecture is one of the most common structures employed to build deep neural networks [24,25] in different domains, including fire detection [26,27]. It is mainly known for its ability to locate and capture local hidden structures by means of convolutional kernels, or filters, within the network. Convolution kernels map input feature maps into feature maps based on inputs, which in turn are further convoluted. A stride is a horizontal interval between adjacent patches that can overlap, and it is measured by the distance between adjacent patches. Each patch in the same input shares a convolutional kernel set of parameters which can be learned over time. As a result of the input padding being required in some cases, the size of the input can always be maintained without changing. In order to increase the nonlinear capability of the CNN, the activation function of the feature map can be used to increase its nonlinear capacity. The activation function can be composed of the following functions: ReLU, sigmoid, tanh, weakly ReLU, and ELU. In CNNs, the pooling function is a nonlinear down-sampling procedure whose purpose is to reduce the dimensionality of representations. This serves to speed up computations by reducing their dimension. A further advantage of the pooling technique is that it is capable of avoiding or reducing the problem of over-fitting.

2.4. Multi-Layer Bi Direction Gated Recurrent Units

In the area of recurrent neural networks (RNNs), long-short term memory (LSTM) [28] is also known as a recurrent neural network (RNN) [29]. RNNs are particularly suitable for time series questions because of the way their architecture works: they share weights at every single time step in a series. Various applications of RNNs have been made in various fields, including anomaly detection [23], continuous B-cell epitope prediction [30,31], sentiment analysis [32,33], action recognition [34], and time series data analysis [35,36]. Generally, RNNs are prone to causing gradient vanishing or exploding when they are applied to long sequences of data; this is one of their major shortcomings. Consequently, RNNs were only capable of analyzing short sequences of data [37]. As a result of LSTM [28], gate mechanisms were used to control information conveying, which included selective additions and deletions of information that had already been accumulated. Although the LSTM was effective in capturing the relationship between words that were in the front and those in the back, it was unable to characterize the relationship between the words in the back and those in the front. In order to tackle this issue effectively, Bi-LSTM solutions are used [25,38]. By utilizing GRU [39], Cho proposed a new model that accounted for recurrences on different time scales, using loop blocks as an adaptive means of simplifying LSTM models without reducing their effectiveness. GRU models require the previous word vector results to be processed in order to perform the actual computation of the current word vector for a sequence $\mathcal{S} = [\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \ldots\ldots \mathcal{S}_N]$. Figure 2 illustrates the GRU model that was developed. In contrast to the LSTM model, the GRU model did not contain any storage units, which was an important difference from the LSTM model. As a result of these calculations, the following result was obtained:

$$r_t = \sigma(\mathcal{W}_r \cdot [h_{t-1} * \mathcal{X}_t]) \tag{3}$$

$$z_t = \sigma(\mathcal{W}_z \cdot [h_{t-1} * \mathcal{X}_t]) \tag{4}$$

$$h^{\sim}{}_t = tanh(\mathcal{W}_h \cdot [r_t \ * h_{t-1} * \mathcal{X}_t]) \tag{5}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * h^{\sim}{}_t \tag{6}$$

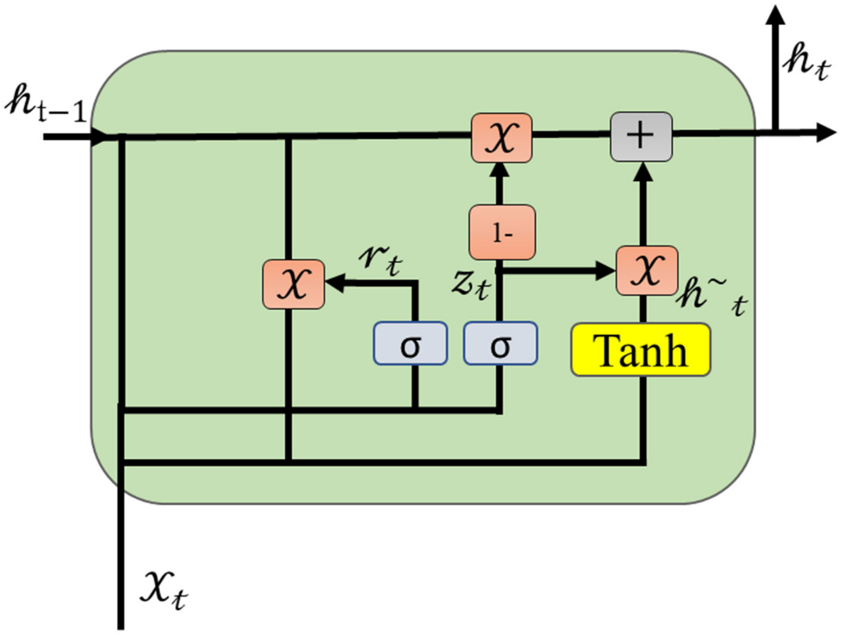


Figure 2. Illustration for the internal mechanism of GRU.

In the equations above, $*$ is a symbol that represents the multiplication of the elements corresponding to the matrices. During the reset gate r_t, all the previous activations of the units of the same layer are taken and updated to determine the number of units being updated. Depending on its activation, the number of units that are updated from the update gate z_t is determined. Finally, the activation unit for a GRU is generated by combining the past activation unit with the current candidate unit.

2.5. Attention Mechanism with GRU

The field of deep learning has increasingly focused on studying attention mechanisms. The mechanisms of attention are essentially a way of allocating weights. The scheme of distributing weights is very similar to what one does when one watches an object and places a different focus on different parts of the object. It is well known that there are several attention schemes, such as feed-forward attention [40] and self-attention [41]. In order to address the medium-term dependency related to the GRU, a feed-forward attention was used to compensate for the GRU's deficiencies in this area. We assumed that $\hbar_{\mathrm{t}}$ is the hidden state in the GRU at time step. In order to generate the context vector, the feed-forward attention method was used, as follows:

$$\mathbb{C} = \sum_{\mathrm{t}=1}^{\mathbb{T}} a_{\mathrm{t}}\, \hbar_{\mathrm{t}} \tag{7}$$

$$a_{\mathrm{t}} \; = exp(e_{\mathrm{t}}) / \sum_{\mathrm{t}=1}^{\mathbb{T}} exp(e_{\mathrm{t}})' \tag{8}$$

$$e_{\mathrm{t}} = \beta(\hbar_{\mathrm{t}}) \tag{9}$$

In the above equations, a_{t} presents the attention weight of the model hidden state, while $\hbar_{\mathrm{t}}$ and β are the learning parameters of the proposed model.

3. Results and Discussion

We experimentally evaluated the proposed model using various enhancer sequence datasets in order to test its performance. The evaluation metrics we used are in general use in state-of-the-art schemes as ways to assess their effectiveness. In our experiments, the results clearly proved the success of our proposed method, providing a greater degree of precision in identifying enhancer sequences compared to existing methods.

3.1. Training Detail, Cross Validation and Evaluation Metrics

To implement the Enhancer-AttGRU, we used Python and TensorFlow (version 2.0) as deep learning tools. In order to assess the validity of our method, we conducted tenfold cross-validation and independent testing on a notebook computer with 32G RAM and six CPUs, each of which featured a speed of 2.60 GHz. In the training process, each epoch took about 25 s, while prediction of each sample required just 2 s using the trained Enhancer-AttGRU. Table 2 shows the number of parameters, the shape of the output, and the number of layers in the enhancement LSTM algorithm for each layer. $\mathrm{D_S}$, $\mathrm{N_S}$, and $\mathrm{F_s}$ represent total numbers of dataset samples, number of steps, and features space, respectively.

Table 2. Detailed information on input shape and number of parameters in proposed Enhancer-AttGRU model.

Layer	Output Shape	Param
Input layer	[(D_S, N_S, F_s)]	0
conv1d (Conv1D))	(None, 298, 27)	459
max_pooling1d (MaxPooling1D)	(None, 99, 27)	0
dropout (Dropout)	(None, 99, 27)	0
conv1d_1 (Conv1D)	(None, 99, 27)	770
bidirectional (Bidirectional GRU)	(None, 256)	110,592
Attention (None, 64)	(None, 256)	0
Dropout_1 (Dropout)	(None, 256)	0
Dense (Dense)	(None, 128)	32,896
Dense_1 (Dense)	(None, 64)	8256
Dense_2 (Dense)	(None, 64)	4160
Dense_3 (Dense)	(None, 16)	1040
Dense_4 (Dense)	(None, 2)	34

We used a tenfold cross-validation approach and independent testing in order to test the predictive performance of our presented method. In the n-fold cross-validation method, the training dataset was divided into n parts, some of which had the same size, others which had only an approximate equal size. The parts of the dataset were used to train and test the model were n−1 parts. The process was repeated n times in order to achieve the desired result. For the independent test, the training datasets were used in training the model, while the independent datasets were used in testing it. Since this was a binary classification problem, we evaluated the performance of the model using common metrics such as sensitivity ($\mathbb{SN}$), specificity ($\mathbb{SP}$), accuracy ($\mathbb{ACC}$), and Matthews' correlation coefficient ($\mathbb{MCC}$), all of which were defined as follows:

$$\mathbb{SN} = {}^{\mathbb{TP}}/_{\mathbb{TP}+\mathbb{FN}} \tag{10}$$

$$\mathbb{SP} = {}^{\mathbb{TN}}/_{\mathbb{FP}+\mathbb{TN}} \tag{11}$$

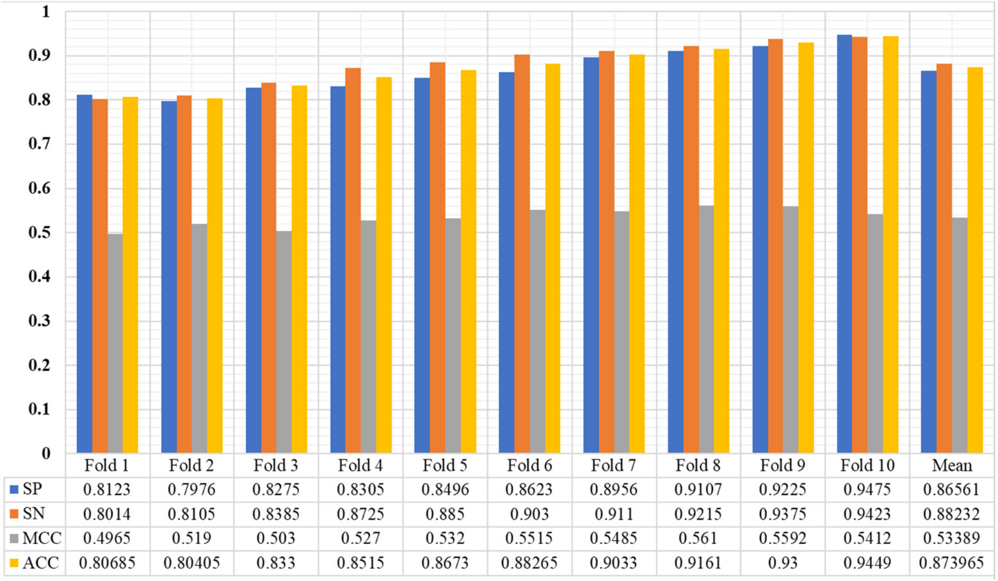
$$\mathbb{ACC} = {}^{\mathbb{TP}+\mathbb{TN}}/_{\mathbb{TP}+\mathbb{FN}+\mathbb{FP}+\mathbb{TN}} \tag{12}$$

$$\mathbb{MCC} = {}^{\mathbb{TP}\times\mathbb{TN}-\mathbb{FP}\times\mathbb{FN}}/_{\sqrt{(\mathbb{TP}+\mathbb{FN})(\mathbb{TP}+\mathbb{FP})(\mathbb{TN}+\mathbb{FN})(\mathbb{TN}+\mathbb{FP})}} \tag{13}$$

In the above equations, $\mathbb{TP}$ represents the number of true positive samples, $\mathbb{FN}$ represents the number of false negative samples, $\mathbb{FP}$ represents the number of false positive samples, and $\mathbb{TN}$ represents the number of true negative samples. It is generally accepted that $\mathbb{SN}$, $\mathbb{SP}$, and $\mathbb{ACC}$ belong to the 0–1 range. In most cases, the $\mathbb{SN}$, $\mathbb{SP}$, and $\mathbb{ACC}$ fall between 0 and 1 and the $\mathbb{MCC}$ is between −1 and 1. In general, higher values of $\mathbb{SN}$, $\mathbb{SP}$, $\mathbb{ACC}$, and $\mathbb{MCC}$ indicate greater efficiency.

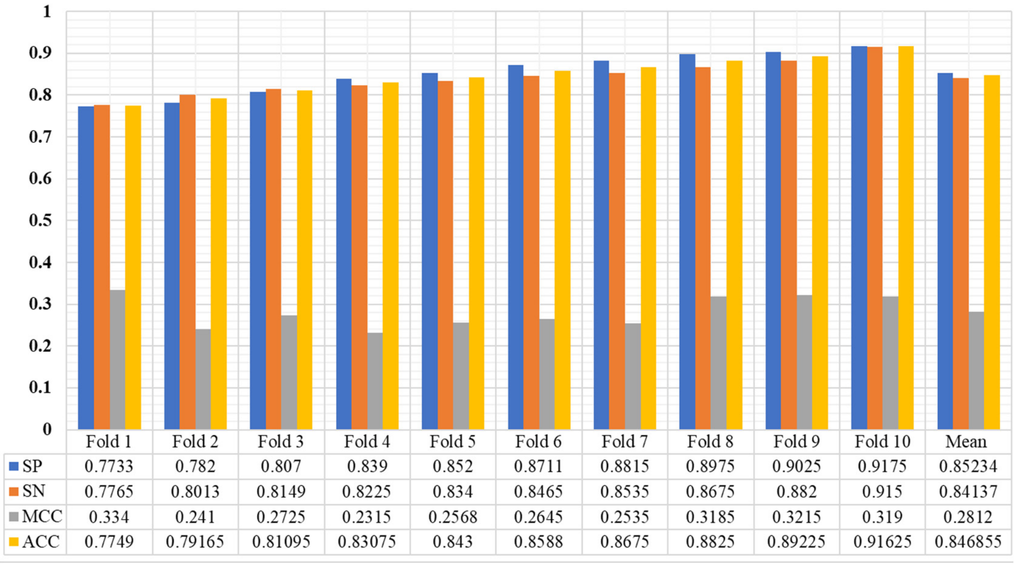
3.2. Results

The Enhancer-AttGRU was tested to determine whether it could distinguish between enhancers and non-enhancers, as well as between strong enhancers and weak enhancers, based on its ability to identify both enhancers and non-enhancers. During the first stage of distinguishing between enhancers and non-enhancers, all of the enhancers, including weak enhancers, were considered positive samples. In the first stage, all of the enhancers were positive samples. In the second stage of the process, the strong enhancers were classified as positive, and the weak enhancers were classified as negative, a process of discriminating strong from weak enhancers. We performed a tenfold cross-validation on the dataset $\mathcal{D}$ in order to evaluate the performance. In Figure 3, we presented the experimental results of the proposed model. During the first stage of the experiment, we were able to achieve an average accuracy of 0.8739%, and during the second stage, 0.8468%. As a result, the proposed model achieved $\mathbb{SN}$ 0.8823, $\mathbb{SP}$ 0.8656, and $\mathbb{MCC}$ 0.5339 in the first stage. While in the second stage the proposed model accomplished $\mathbb{SN}$ 0.8413, $\mathbb{SP}$ 0.8523, $\mathbb{ACC}$ 0.8468, and $\mathbb{MCC}$ 0.2804, as shown in Figure 4. It was evident that the first stage of the analysis produced much better predictive performance results than the second stage, which suggested that it was more difficult to determine whether strong enhancers were stronger or weaker than it was to determine whether non-enhancers were stronger.



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean
■ SP	0.8123	0.7976	0.8275	0.8305	0.8496	0.8623	0.8956	0.9107	0.9225	0.9475	0.86561
■ SN	0.8014	0.8105	0.8385	0.8725	0.885	0.903	0.911	0.9215	0.9375	0.9423	0.88232
■ MCC	0.4965	0.519	0.503	0.527	0.532	0.5515	0.5485	0.561	0.5592	0.5412	0.53389
■ ACC	0.80685	0.80405	0.833	0.8515	0.8673	0.88265	0.9033	0.9161	0.93	0.9449	0.873965

Figure 3. The performance of the proposed model using benchmark enhancers sequence datasets.



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean
■ SP	0.7733	0.782	0.807	0.839	0.852	0.8711	0.8815	0.8975	0.9025	0.9175	0.85234
■ SN	0.7765	0.8013	0.8149	0.8225	0.834	0.8465	0.8535	0.8675	0.882	0.915	0.84137
■ MCC	0.334	0.241	0.2725	0.2315	0.2568	0.2645	0.2535	0.3185	0.3215	0.319	0.2812
■ ACC	0.7749	0.79165	0.81095	0.83075	0.843	0.8588	0.8675	0.8825	0.89225	0.91625	0.846855

Figure 4. The performance of the proposed model using benchmark enhancers sequence dataset (Stage 2).

3.3. Comparison of the Proposed Model with Existing Techniques

This study evaluated the proposed model against state-of-the-art classification models, such as EnhancerPred [21], iEnhancer-RF [42], iEnhancer-PsedeKNC [43], DeployEnhance [44], iEnhancer-EL [16], iEnhancer-RD [45], Enhancer-LSTMAtt [7], iEnhancer-XG [46], iEnhancer-2L [3], iEnhancer-5Step [47], iEnhancerDSNet [48], and iEnhancer-CNN [49]. The proposed model was compared to all of these prediction methodologies in order to make a more accurate comparison. As in aforementioned studies, both of these studies used the same benchmark dataset to do their functional evaluations and analyze the data collected during the course of these studies. The comparison results showed that our model was significantly more accurate than other models. The performance of the state-of-the-art predictors in the first stage is depicted in Figure 5. A comparison of the prediction performances in the second stage is illustrated in Figure 6. In each class of metrics, the most noteworthy values are highlighted in the table below. Moreover, as illustrated in Figure 5, the performances of enhancer identification in the first stage of the algorithm were improved by 12.69%, 13.39%, 9.89%, 11.21%, 11.21%, 10.61%, 10.84%, 10.84%, 10.84%, 8.59%, and 13.39%, respectively, by resolving the enhancers in $\mathbb{SN}$, $\mathbb{ACC}$, and $\mathbb{MCC}$, respectively. Similarly, the second stage also showed significant improvements, of 23.68%, 29.68%, 09.68%, 24.18%, 22.15%, 21.27%, 25.72%, 20.73%, 14.18%, and 23.068%, respectively, as illustrated in Figure 6, for $\mathbb{SN}$, $\mathbb{SP}$, $\mathbb{ACC}$, and $\mathbb{MCC}$, respectively. Based on a detailed comparison between the performance of the proposed model and the performance of the existing model in our study, it was found that the proposed model achieved significant improvements in model execution. This was based on measuring the performance of the model using the performance assessment metrics. Both stages 1 and 2 demonstrated significant improvements in perceived parameters as a result of the proposed predictor, and this improvement was observed in both phases. We were able to conclude from the considerable increases in $\mathbb{MCC}$ values that our study provided a substantial increase in the stability of the predictor and a greater level of performance overall than that seen in state-of-the-art methodologies, which had smaller $\mathbb{MCC}$ values on average. As a result of this advancement, binary classification problems can now be verified to be consistent. A comparison between the $\mathbb{MCC}$ value and the $\mathbb{ACC}$ value indicated that the $\mathbb{MCC}$ value demonstrated a greater level of insight, in that it considered the extent of each of the four parameters ($\mathbb{TN}$, $\mathbb{TP}$, $\mathbb{FP}$, and $\mathbb{FN}$) of the confusion matrix. This demonstrated a balanced assessment during model evaluation [50]. The results of iEnhancer-Deep showed that it was capable of producing results similar to other, previously proposed strategies.

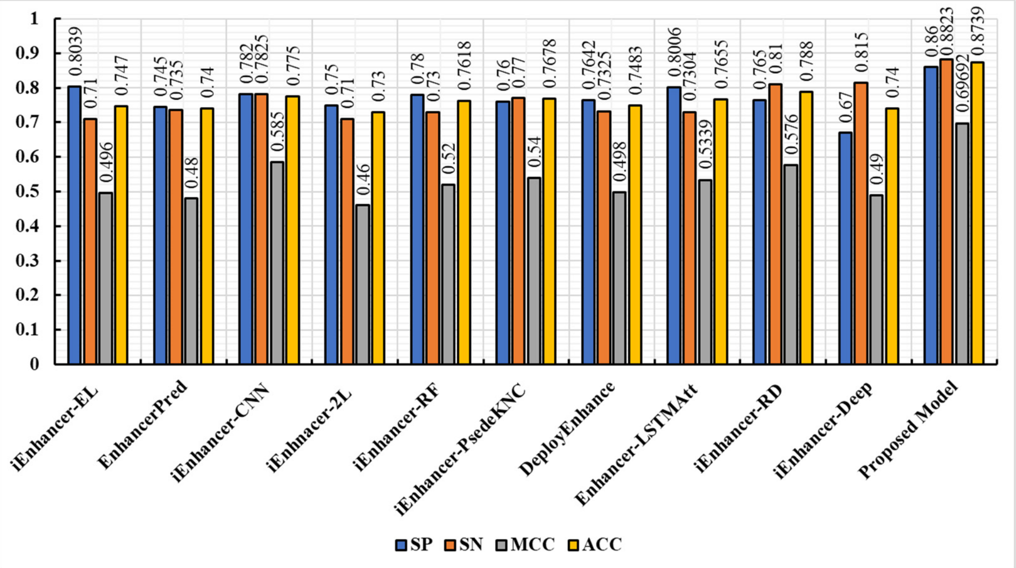


Figure 5. Performance comparison of the proposed model with baseline techniques (Stage 1).

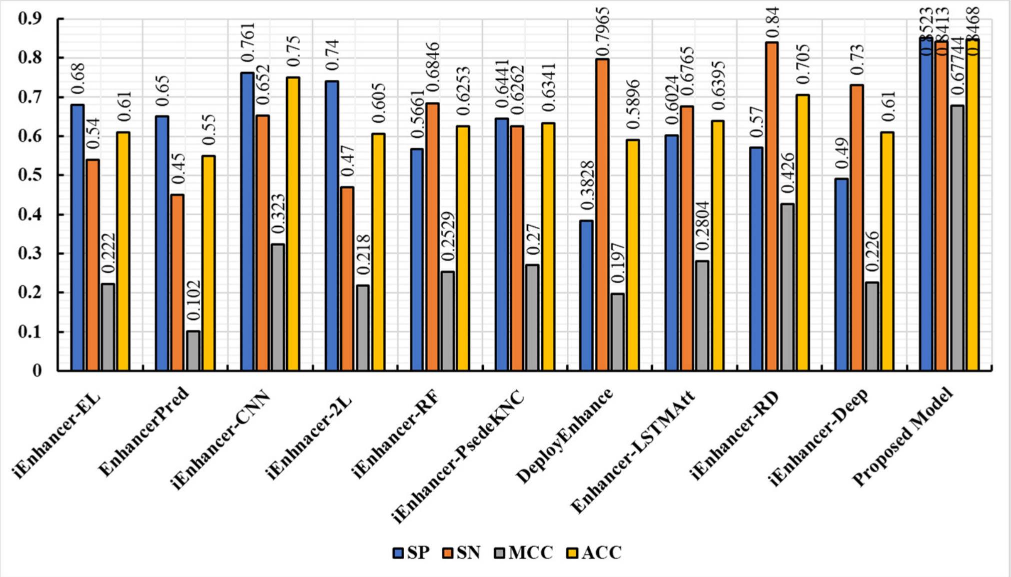


Figure 6. Performance comparison of the proposed model with existing techniques (stages 2).

3.4. Experimental Result on Independent Dataset

The proposed model was further tested using the independent dataset that was also presented in [16]. In this dataset, there were 100 weak enhancers, 100 strong enhancers, and 200 non-enhancers. Based on the results of the first stage comparison, Figure 7 shows the proposed model's results against other state-of-the-art models. Figure 8 reveals the results of the second stage comparison. The results of the proposed model showed that, in terms of sensitivity, accuracy, and specificity, the proposed model performed better in both stages. In addition, the model was also capable of predicting the true enhancer sites and the strength of the enhancers, indicating its robustness. Generally, there was more confidence in the prediction time when iEnhancer-CNN was used in combination with the results of the proposed model.

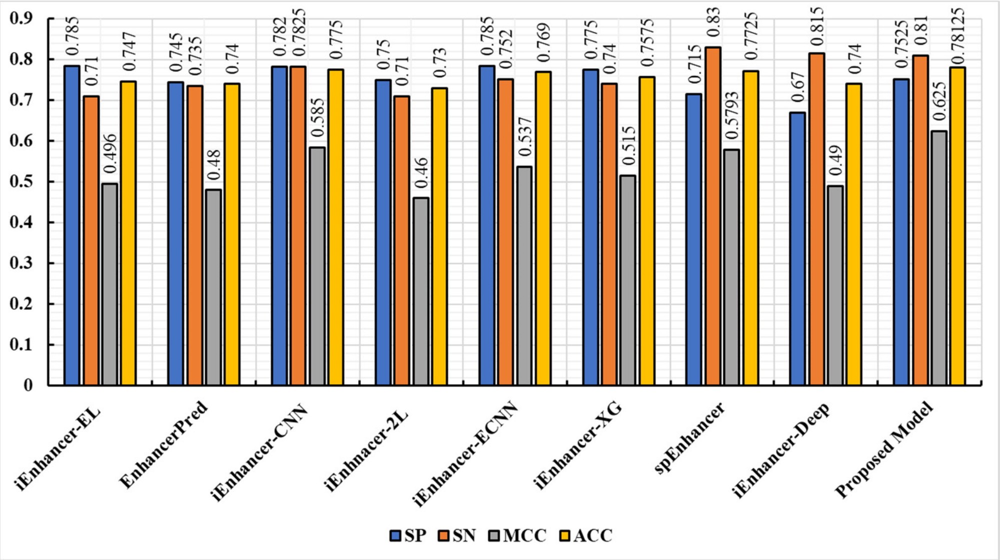


Figure 7. The performance of state-of-the-art predictors and the proposed model in the independent test in the first stage.

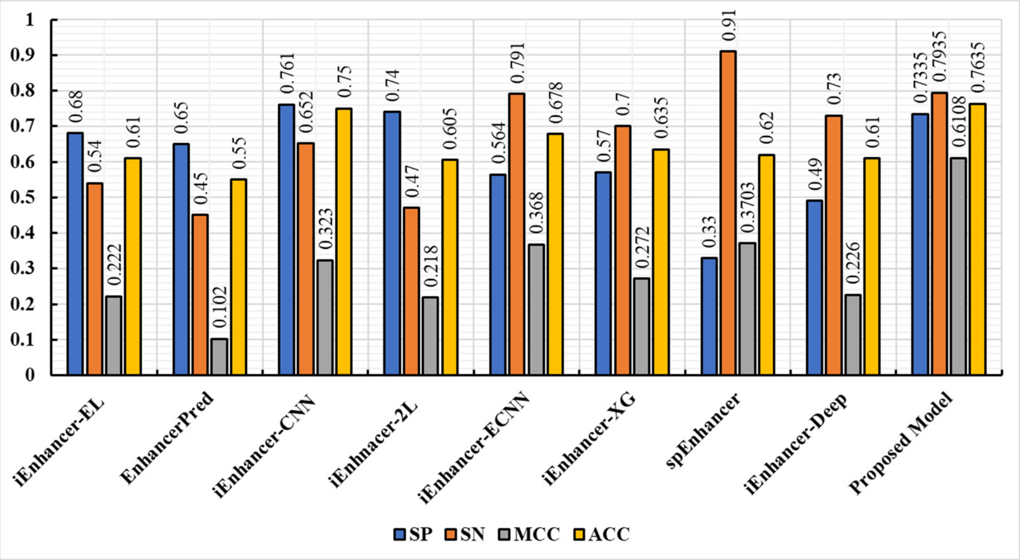


Figure 8. Comparison of the proposed model's performance against state-of-the-art predictors in the second stage.

3.5. Discussion

We studied the effects of a number of non-enhancers on the methods in order to establish their effectiveness. Our sampling and mutation strategy was used to generate new non-enhancers. The sampling and mutation strategy was used because there were no non-enhancers available to generate before sampling. Three distinct sets of non-enhancers, including mutated non-enhancers and non-mutated non-enhancers, were formulated, along with three new training sets. As a result of the independent test that we conducted, our objective was to check the performance of the proposed method (trained from the new training sets) in order to assess its efficiency. There were up to 158,207 parameters that could be trained in the Enhancer-AttGRU. In the case of deep learning, the more parameters that can be trained, the greater the risk of overfitting. In order to reduce overfitting of the model, we used dropouts. Enhancer-AttGRU is a deep learning and end-to-end method that requires no feature design whatsoever, and is based on deep learning. Therefore, it avoids the use of artificial interference and complex methods of extracting or selecting features. In this regard, Enhancer-AttGRU is easier to implement than the feature-based methods. In cross-validation tests, the majority of feature-based methods did quite well, but they fared badly in independent tests, suggesting that they were not very generalizable. In this study, the performance of the Enhancer-AttGRU model was compared with that of nine state-of-the-art models. There are a number of deep learning-based techniques that use either CNNs or LSTMs for enhancer recognition, or combinations of these two techniques. It is important to keep in mind that these techniques are computationally intensive and have limited recognition capabilities.

4. Conclusions and Future Directions

There is a major role to be played by enhancers in regulating transcription for target genes. These must be identified in order to uncover their role. There is one fundamental issue which we have to deal with, and that is the difference between enhancers and non-enhancers. Initially, this classification was done using biological experiments, but, given the amount of time, money, and effort that is required to classify enhancers in this manner, it was not possible to perform this classification so early on. Therefore, we used a computational approach based on deep learning for the purpose of quickly distinguishing enhancers from others. The proposed model performed two tasks, namely: identifying enhancers and estimating the strength of these enhancers. The experimental results revealed that the proposed model outperformed state-of-the-art models. In contrast with state-of-the-art strategies, a comprehensive comparison with the proposed model suggested that the method was more than stable, it was also a highly effective and efficient method for identifying enhancers. In the future, we will explore sequence coding schemes, feature extraction methods, and data augmentation methods in order to further improve the predictive ability of the model, in terms of accuracy.

Author Contributions: Conceptualization, S.A., S.H. and S.U.K.; methodology, S.A.A. and M.I.; software, S.H. and M.I.; validation, S.U.K., S.A.A. and M.I.; formal analysis, S.U.K. and S.H.; investigation, S.A. and A.A.; resources, S.H., S.U.K. and M.I.; data curation, A.A.; writing—original draft preparation, S.A.; writing—review and editing, S.U.K., S.H. and M.I.; visualization, A.A.; supervision, M.I. and S.U.K.; project administration, S.H.; funding acquisition, M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://github.com/Saeedkhattak194/Enhancer-AttGRU, accessed on 4 November 2022.

Acknowledgments: The authors would like to thank the research Chair of Prince Faisal for Artificial Intelligence (CPFAI) for supporting this research work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pennacchio, L.A.; Bickmore, W.; Dean, A.; Nobrega, M.A.; Bejerano, G. Enhancers: Five essential questions. *Nat. Rev. Genet.* **2013**, *14*, 288–295. [CrossRef] [PubMed]
2. Plank, J.L.; Dean, A. Enhancer function: Mechanistic and genome-wide insights come together. *Mol. Cell* **2014**, *55*, 5–14. [CrossRef] [PubMed]
3. Liu, B.; Fang, L.; Long, R.; Lan, X.; Chou, K.-C. iEnhancer-2L: A two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. *Bioinformatics* **2016**, *32*, 362–369. [CrossRef] [PubMed]
4. Bejerano, G.; Pheasant, M.; Makunin, I.; Stephen, S.; Kent, W.J.; Mattick, J.S.; Haussler, D. Ultraconserved elements in the human genome. *Science* **2004**, *304*, 1321–1325. [CrossRef] [PubMed]
5. Boyd, M.; Thodberg, M.; Vitezic, M.; Bornholdt, J.; Vitting-Seerup, K.; Chen, Y.; Coskun, M.; Li, Y.; Lo, B.Z.S.; Klausen, P.; et al. Characterization of the enhancer and promoter landscape of inflammatory bowel disease from human colon biopsies. *Nat. Commun.* **2018**, *9*, 1661. [CrossRef] [PubMed]
6. Shlyueva, D.; Stampfel, G.; Stark, A. Transcriptional enhancers: From properties to genome-wide predictions. *Nat. Rev. Genet.* **2014**, *15*, 272–286. [CrossRef]
7. Alsanea, M.; Habib, S.; Khan, N.F.; Alsharekh, M.F.; Islam, M.; Khan, S. A Deep-Learning Model for Real-Time Red Palm Weevil Detection and Localization. *J. Imaging* **2022**, *8*, 170. [CrossRef]
8. Zuhaib, M.; Shaikh, F.A.; Tanweer, W.; Alnajim, A.M.; Alyahya, S.; Khan, S.; Usman, M.; Islam, M.; Hasan, M.K. Faults Feature Extraction Using Discrete Wavelet Transform and Artificial Neural Network for Induction Motor Availability Monitoring—Internet of Things Enabled Environment. *Energies* **2022**, *15*, 7888. [CrossRef]
9. Albattah, W.; Kaka Khel, M.H.; Habib, S.; Islam, M.; Khan, S.; Abdul Kadir, K. Hajj Crowd Management Using CNN-Based Approach. *Comput. Mater. Contin.* **2020**, *66*, 2183–2197. [CrossRef]
10. Ghandi, M.; Lee, D.; Mohammad-Noori, M.; Beer, M.A. Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput. Biol.* **2014**, *10*, e1003711. [CrossRef]
11. Kleftogiannis, D.; Kalnis, P.; Bajic, V.B. DEEP: A general computational framework for predicting enhancers. *Nucleic Acids Res.* **2015**, *43*, e6. [CrossRef] [PubMed]
12. Fernandez, M.; Miranda-Saavedra, D. Genome-wide enhancer prediction from epigenetic signatures using genetic algorithm-optimized support vector machines. *Nucleic Acids Res.* **2012**, *40*, e77. [CrossRef] [PubMed]
13. Rajagopal, N.; Xie, W.; Li, Y.; Wagner, U.; Wang, W.; Stamatoyannopoulos, J.; Ernst, J.; Kellis, M.; Ren, B. RFECS: A random-forest based algorithm for enhancer identification from chromatin state. *PLoS Comput. Biol.* **2013**, *9*, e1002968. [CrossRef]
14. Bu, H.; Gan, Y.; Wang, Y.; Zhou, S.; Guan, J. A new method for enhancer prediction based on deep belief network. *BMC Bioinform.* **2017**, *18*, 418. [CrossRef]
15. Yang, B.; Liu, F.; Ren, C.; Ouyang, Z.; Xie, Z.; Bo, X.; Shu, W. BiRen: Predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics* **2017**, *33*, 1930–1936. [CrossRef]

16. Liu, B.; Li, K.; Huang, D.S.; Chou, K.C. iEnhancer-EL: Identifying enhancers and their strength with ensemble learning approach. *Bioinformatics* **2018**, *34*, 3835–3842. [CrossRef]
17. Ullah, W.; Muhammad, K.; Haq, I.U.; Ullah, A.; Khattak, S.U.; Sajjad, M. Splicing sites prediction of human genome using machine learning techniques. *Multimed. Tools Appl.* **2021**, *80*, 30439–30460. [CrossRef]
18. Ahmad, F.; Ikram, S.; Ahmad, J.; Ullah, W.; Hassan, F.; Khattak, S.U.; Rehman, I.U. GASPIDs Versus Non-GASPIDs-Differentiation Based on Machine Learning Approach. *Curr. Bioinform.* **2020**, *15*, 1056–1064. [CrossRef]
19. Habib, S.; Alsanea, M.; Aloraini, M.; Al-Rawashdeh, H.S.; Islam, M.; Khan, S. An Efficient and Effective Deep Learning-Based Model for Real-Time Face Mask Detection. *Sensors* **2022**, *22*, 2602. [CrossRef]
20. Ali, S.D.; Alam, W.; Tayara, H.; Chong, K. Identification of functional piRNAs using a convolutional neural network. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *9*, 8491–8496. [CrossRef]
21. Jia, C.; He, W. EnhancerPred: A predictor for discovering enhancers based on the combination and selection of multiple features. *Sci. Rep.* **2016**, *6*, 38741. [CrossRef] [PubMed]
22. Fu, L.; Niu, B.; Zhu, Z.; Wu, S.; Li, W. CD-HIT: Accelerated for clustering the next-generation sequencing data. *Bioinformatics* **2012**, *28*, 3150–3152. [CrossRef]
23. Ullah, W.; Ullah, A.; Haq, I.U.; Muhammad, K.; Sajjad, M.; Baik, S.W. CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks. *Multimed. Tools Appl.* **2021**, *80*, 16979–16995. [CrossRef]
24. Ullah, W.; Ullah, A.; Hussain, T.; Khan, Z.A.; Baik, S.W. An efficient anomaly recognition framework using an attention residual LSTM in surveillance videos. *Sensors* **2021**, *21*, 2811. [CrossRef] [PubMed]
25. Ullah, W.; Ullah, A.; Hussain, T.; Muhammad, K.; Heidari, A.A.; Del Ser, J.; Baik, S.W.; De Albuquerque, V.H.C. Artificial Intelligence of Things-assisted two-stream neural network for anomaly detection in surveillance Big Video Data. *Future Gener. Comput. Syst.* **2022**, *129*, 286–297. [CrossRef]
26. Khan, Z.A.; Hussain, T.; Ullah, F.U.M.; Gupta, S.K.; Lee, M.Y.; Baik, S.W. Randomly Initialized CNN with Densely Connected Stacked Autoencoder for Efficient Fire Detection. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105403. [CrossRef]
27. Yar, H.; Hussain, T.; Agarwal, M.; Khan, Z.A.; Gupta, S.K.; Baik, S.W. Optimized Dual Fire Attention Network and Medium-Scale Fire Classification Benchmark. *IEEE Trans. Image Process.* **2022**, *31*, 6331–6343. [CrossRef] [PubMed]
28. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
29. Giles, C.L.; Kuhn, G.M.; Williams, R.J. Dynamic recurrent neural networks: Theory and applications. *IEEE Trans. Neural Netw.* **1994**, *5*, 153–156. [CrossRef]
30. Saha, S.; Raghava, G.P.S. Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. *Proteins Struct. Funct. Bioinform.* **2006**, *65*, 40–48. [CrossRef]
31. Ullah, W.; Ullah, A.; Malik, K.M.; Saudagar, A.K.J.; Khan, M.B.; Hasanat, M.H.A.; AlTameem, A.; AlKhathami, M. Multi-Stage Temporal Convolution Network for COVID-19 Variant Classification. *Diagnostics* **2022**, *12*, 2736. [CrossRef] [PubMed]
32. Arras, L.; Montavon, G.; Müller, K.-R.; Samek, W. Explaining recurrent neural network predictions in sentiment analysis. *arXiv* **2017**, arXiv:1706.07206.
33. Ullah, W.; Hussain, T.; Khan, Z.A.; Haroon, U.; Baik, S.W. Intelligent dual stream CNN and echo state network for anomaly detection. *Knowl.-Based Syst.* **2022**, *253*, 109456. [CrossRef]
34. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1110–1118.
35. Khan, Z.A.; Ullah, A.; Haq, I.U.; Hamdy, M.; Mauro, G.M.; Muhammad, K.; Hijji, M.; Baik, S.W. Efficient short-term electricity load forecasting for effective energy management. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102337. [CrossRef]
36. Khan, Z.A.; Hussain, T.; Haq, I.U.; Ullah, F.U.M.; Baik, S.W. Towards efficient and effective renewable energy prediction via deep learning. *Energy Rep.* **2022**, *8*, 10230–10243. [CrossRef]
37. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
38. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The performance of LSTM and BiLSTM in forecasting time series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 3285–3292.
39. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
40. Raffel, C.; Ellis, D.P. Feed-forward networks with attention can solve some long-term memory problems. *arXiv* **2015**, arXiv:1512.08756.
41. Habib, S.; Hussain, A.; Islam, M.; Khan, S.; Albattah, W. Towards Efficient Detection and Crowd Management for Law Enforcing Agencies. In Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; pp. 62–68.
42. Lim, D.Y.; Khanal, J.; Tayara, H.; Chong, K.T. iEnhancer-RF: Identifying enhancers and their strength by enhanced feature representation using random forest. *Chemom. Intell. Lab. Syst.* **2021**, *212*, 104284. [CrossRef]
43. Liu, B. iEnhancer-PsedeKNC: Identification of enhancers and their subgroups based on Pseudo degenerate kmer nucleotide composition. *Neurocomputing* **2016**, *217*, 46–52. [CrossRef]
44. Alsharekh, M.F.; Habib, S.; Dewi, D.A.; Albattah, W.; Islam, M.; Albahli, S. Improving the Efficiency of Multistep Short-Term Electricity Load Forecasting via R-CNN with ML-LSTM. *Sensors* **2022**, *22*, 6913. [CrossRef] [PubMed]
45. Yang, H.; Wang, S.; Xia, X. iEnhancer-RD: Identification of enhancers and their strength using RKPK features and deep neural networks. *Anal. Biochem.* **2021**, *630*, 114318. [CrossRef] [PubMed]

46. Cai, L.; Ren, X.; Fu, X.; Peng, L.; Gao, M.; Zeng, X. iEnhancer-XG: Interpretable sequence-based enhancers and their strength predictor. *Bioinformatics* **2021**, *37*, 1060–1067. [CrossRef] [PubMed]
47. Le, N.Q.K.; Yapp, E.K.Y.; Ho, Q.-T.; Nagasundaram, N.; Ou, Y.-Y.; Yeh, H.-Y. iEnhancer-5Step: Identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and word embedding. *Anal. Biochem.* **2019**, *571*, 53–61. [CrossRef] [PubMed]
48. Asim, M.N.; Ibrahim, M.A.; Malik, M.I.; Dengel, A.; Ahmed, S. Enhancer-dsnet: A supervisedly prepared enriched sequence representation for the identification of enhancers and their strength. In *International Conference on Neural Information Processing*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 38–48.
49. Habib, S.; Hussain, A.; Albattah, W.; Islam, M.; Khan, S.; Khan, R.U.; Khan, K. Abnormal Activity Recognition from Surveillance Videos Using Convolutional Neural Network. *Sensors* **2021**, *21*, 8291. [CrossRef]
50. Habib, S.; Khan, I.; Aladhadh, S.; Islam, M.; Khan, S. External Features-Based Approach to Date Grading and Analysis with Image Processing. *Emerg. Sci. J.* **2022**, *6*, 694–704. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.