Mathematical Biosciences and Engineering

MBE, 20(2): 2815–2830.
DOI: 10.3934/mbe.2023132
Received: 15 October 2022
Revised: 14 November 2022
Accepted: 17 November 2022
Published: 01 December 2022

http://www.aimspress.com/journal/MBE

*Research article*

# DeepDN_iGlu: prediction of lysine glutarylation sites based on attention residual learning method and DenseNet

**Jianhua Jia*, Mingwei Sun, Genqiang Wu and Wangren Qiu***

School of Information Engineering, Jingdezhen Ceramic University, Jingdezhen 333403, China

* **Correspondence:** Email: jjh163yx@163.com, qiuone@163.com.

**Abstract:** As a key issue in orchestrating various biological processes and functions, protein post-translational modification (PTM) occurs widely in the mechanism of protein's function of animals and plants. Glutarylation is a type of protein-translational modification that occurs at active ε-amino groups of specific lysine residues in proteins, which is associated with various human diseases, including diabetes, cancer, and glutaric aciduria type I. Therefore, the issue of prediction for glutarylation sites is particularly important. This study developed a brand-new deep learning-based prediction model for glutarylation sites named DeepDN_iGlu via adopting attention residual learning method and DenseNet. The focal loss function is utilized in this study in place of the traditional cross-entropy loss function to address the issue of a substantial imbalance in the number of positive and negative samples. It can be noted that DeepDN_iGlu based on the deep learning model offers a greater potential for the glutarylation site prediction after employing the straightforward one hot encoding method, with Sensitivity (Sn), Specificity (Sp), Accuracy (ACC), Mathews Correlation Coefficient (MCC), and Area Under Curve (AUC) of 89.29%, 61.97%, 65.15%, 0.33 and 0.80 accordingly on the independent test set. To the best of the authors' knowledge, this is the first time that DenseNet has been used for the prediction of glutarylation sites. DeepDN_iGlu has been deployed as a web server (https://bioinfo.wugenqiang.top/~smw/DeepDN_iGlu/) that is available to make glutarylation site prediction data more accessible.

**Keywords:** post-translational modification; glutarylation site prediction; DenseNet; attention residual learning; focal loss function

## 1. Introduction

Cells require adaptive strategies to maintain metabolic homeostasis, and post-translational modifications (PTMs) are one of the known adaptive mechanisms [1−3]. Post-translational modifications refers to the covalent addition of certain functional groups to a protein after the translation process [4]. These modifications are critical to functional proteomic because they regulate activity, localization, and interaction with other cellular molecules such as proteins, nucleic acids, lipids and cofactors. Cognition and identification of PTM sites can be of direct correlation to cellular processes, including protein degradation, subcellular localization [5] and cellular signaling events [6]. Glutarylation, one of the PTM which occur at active ε-amino groups of specific lysine residues in proteins, was first experimentally identified by Tan et al. [3] in 2014. The study of Tan's group, which combined chemical and biochemical methods, shows that recognizing and understanding glutarylation sites is vital to scientific investigations in many biological processes, such as metabolic processes and mitochondrial functions. Due to the shortcomings of the labor-intensive and time-consuming of the traditional biological sequencing techniques, convenient and available computational methods are required to reduce the cost of obtaining biological information. The use of computational methods to predict various PTM sites led to the creation of a number of matching predictors that provide novel ideas for the identification of PTM sites, including CarbonylDB [7], SulSite-GTB [8], Mal-Prec [9], iDPGK [10], DeepPPSite [11] and DeepSuccinylSite [12].

Although glutarylation is a PTM reported lately, there are already several prediction tools for glutarylation, and the predictor based on traditional machine learning model first appeared. SVM algorithm seems to be quite popular among researchers in 2018. Since Ju and He [13] proposed the first predictor for glutarylation sites based on a biased support vector machine algorithm, Xu et al. [14] and Huang et al. [15] also successively built iGlu-Lys and MDDGlutar predictors based on SVM algorithm, and both have imbalanced datasets, especially the iGlu-Lys predictor with PSPM feature encoding technique achieved MCCs (Mathews Correlation Coefficients) of 0.5098 and 0.5213 for the 10-fold cross-validation and independent testing, respectively.

The key and challenging step in building predictors of PTM sites is the feature engineering. Whether the manually extracting features from existing web applications [16] or using the complex and varied feature encoding methods [17], as well as the feature selection [18] when the feature dimension is large, this step involves how to effectively provide reliable features for the model while minimizing the complexity of the model computation.

The ratio of non-glutarylation sites to glutarylation sites always surpasses 2:1 and often even 5:1 in the natural environment, as in the case of other PTMs, such as succinylation [19]. To address the severe bias problem of traditional machine learning models [20] caused by the imbalance of positive and negative samples, it is common approach to balance the data before feeding them into the model [21], including up-sampling (increasing minority class samples), down-sampling (decreasing majority class samples) and hybrid-sampling (increasing minority class samples while decreasing majority class samples), which results in an excellent performance for traditional machine learning models. For instance, Ju and Wang [22] in 2020 applied the Positive-Unlabeled Learning approach to build a predictor of glutarylation called PUL-GLU, which is essentially down-sampling. Compared to the MCC of 0.54 at 10-fold cross-validation, it decreased by about half on the independent test set, indicating that the generalization ability of the model is still challenging.

With recent rise of deep learning algorithms, the problem above has been solved to a certain extent.

Deep learning is a sub-discipline of machine learning, which is based on artificial neural network and has demonstrated splendid performance in proteomics, for instance in retention time prediction, MS/MS spectrum prediction, de novo peptide sequencing, major histocompatibility complex-peptide binding prediction, and protein structure prediction [23,24]. Automatic feature extraction is a significant advantage of the deep learning-based method, comparing with the traditional machine learning-based method.

Benefit from the ease of extracting data representation, glutarylation predictors based on deep learning algorithms have emerged. To the best of our knowledge, there are two deep learning-based predictors have been proposed recently. Sheraz Naseer et al. [25] employed a series deep models and combined with embedding method for feature extraction, which eventually demonstrated an outstanding score of 0.94 in terms of accuracy. LSTM and its derived deep networks were applied by Liu et al. [26] in their predictor named DNN-E to improve protein sequence representation and obtain a great performance with accuracy, specificity, sensitivity, and correlation coefficient of 0.79, 0.89, 0.59, and 0.51, respectively, which suggests that there should be a huge potential for deep learning techniques in the field of glutarylation sites prediction.

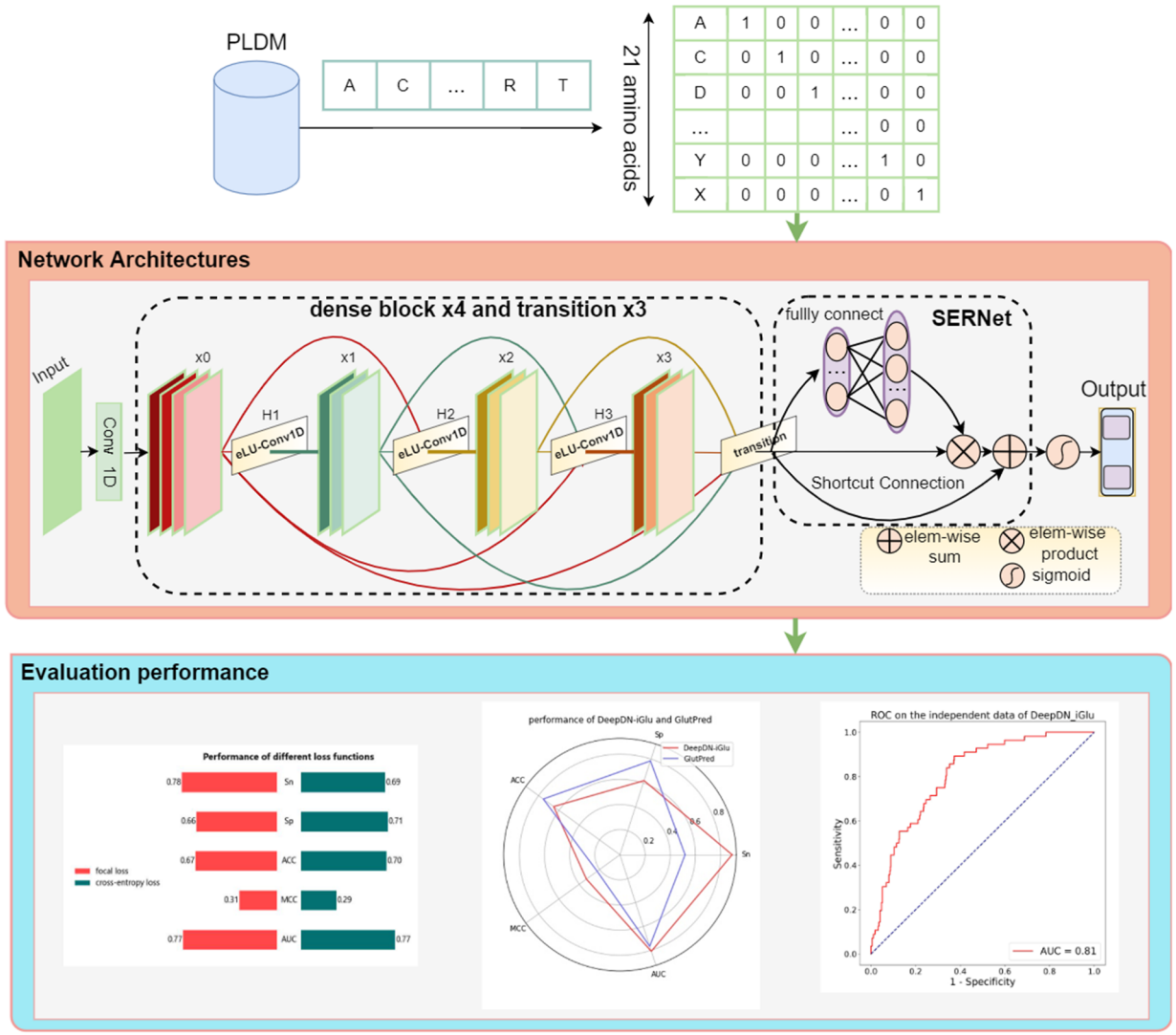


**Figure 1.** The overall flow chart of DeepDN_iGlu.

In this study, we propose a novel deep learning-based prediction model DeepDN_iGlu to meet the current demand in the task of recognizing the glutarylation site. The overall flow chart is shown in Figure 1. The data collecting and encoding procedure, which is ultimately just a straightforward one-hot encoding method is depicted in the figure's top half. Then the core network architecture of this study is depicted in the middle of the figure. It consists of 4 dense blocks with the same structure (a 4-layer dense block with a growth rate of k = 32), 3 transitions layers between them (all consist of an ELU, a 1 × 1 one-dimensional convolution (Conv 1D) and a 2 × 2 average pooling), a layer of SERNet, and a sigmoid classifier to produce the output findings. Finally, the specific evaluation performance will be presented in Section 3.

## 2. Materials and methods

### *2.1. Benchmark dataset*

The previous study by Ju and He [13] served as the source for the benchmark data in this work. To increase the credibility of the data, they extracted 211 proteins with glutarylation sites from the PLDM [27] database and used the CD-HIT [28,29] tool to eliminate proteins with a similarity of more than 40%. In this study, 187 proteins with 646 glutarylation sites were the end result. 167 proteins, which contained 590 glutarylation sites and 3498 non-glutarylation lysine sites, were employed as the source of the training set. The remaining 20 proteins, which contained 56 glutarylation sites and 428 non-glutarylation lysine sites, were then randomly chosen as the independent test set. Then, applying Chou's peptide cleavage method [30], each amino acid sequence that may contain k-glu is represented as Eq (1).

$$f_{\delta}(K) = A_{-\delta}A_{-(\delta-1)} \cdots A_{-2}A_{-1}KA_{+1}A_{+2} \cdots A_{+(\delta-1)}A_{+\delta} \tag{1}$$

where $K$ stands for the central lysine residue, and A for the amino acid residue right next to $K$. The distance between each amino acid residue and the central lysine residue $K$ is shown by the subscript. The distance is 1 if $A$ is adjacent to the central lysine residue $K$, 2 if $A$ is separated from it by an amino acid residue, and so on. The subscript's positive sign (+) denotes a position downstream of the central lysine residue $K$, whereas the subscript's negative sign (–) denotes a position upstream of $K$. $\delta$ is the value of the farthest distance from the central lysine $K$.

The experimental results, according to Ju's article, are better when $\delta = 17$, which means that the amino acid sequence length $L$ (also known as window size), which was used in this investigation, is $17 \times 2 + 1 = 35$. It is worth noting that amino acid sequence complementation was carried out using unknown amino acid $X$ in cases where the central lysine residue $K$ had less than 17 amino acid residues on either end to guarantee that all amino acid sequences were the same length for subsequent tests. We obtained a total of 3988 amino acid sequences after cutting the 187 chosen peptides according to the aforementioned rules, of which the amino acid sequences whose central lysine has been confirmed by prior experiments to be a glutarylation site were identified as positive samples and the remaining amino acid sequences were identified as negative samples. Table 1 displays the benchmark dataset's specifics.

**Table 1.** The specifics of the benchmark dataset.

| Original dataset | Number of proteins | Positive site | Negative site |
|---|---|---|---|
| Training dataset | 167 | 590 | 3498 |
| Testing dataset | 20 | 56 | 428 |

### *2.2. Feature extraction methods*

The appropriate features of protein sequences play very important roles in the prediction of PTM sites. In this paper, three candidate coding methods are used: one hot, EAAC and *K*-Spaced. It should be noted that the value of the second dimension must be set to 21 in order to all three encoding methods to be combined. In this paper, an amino acid sequence after one hot encoding, EAAC encoding and *K*-spaced encoding respectively all end up as two-dimensional matrices with sizes of 35 × 21, 31 × 21 and 210 × 21 respectively, keeping the size of the second dimension as 21 (20 amino acids plus an unknown amino acid *X*), then we can concatenate the matrices by rows. For three matrices, then we can end up with (35 + 31 + 210) × 21 = 276 × 21 input matrices, which is the final encoding matrix after the fusion of the three coding methods.

#### 2.2.1. One hot encoding

One hot encoding is a common coding method in feature engineering, which is favored by researchers for its unique differentiated encoding as well as its simplicity, speed and convenience. First, the amino acids are arranged alphabetically and labeled starting with 0 in one hot encoding. Given that there are 20 amino acid species and an unknown amino acid *X* is added at the end, the amino acids are labeled as "0, 1, 2, 3, 4, 5, 6, 7, …, 20" respectively in alphabetical order "ACDEFGHIKLMNPQRSTVWYX". Second, each amino acid was coded separately, and the length of the encoding sequences was the total number of amino acids (20 amino acids plus one unknown amino acid *X*), and the coding elements were 0 and 1, where the position corresponding to the amino acid label was coded as 1 and the remaining positions were coded as 0. For example, amino acid *A* is coded as (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) amino acid *C* is coded as (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) and so on.

Therefore, in this study, an amino acid sequence of length $L = 35$, is encoded as a 35 × 21 two-dimensional matrix with elements 0 and 1.

#### 2.2.2. EAAC encoding

EAAC is a variation of AAC that treats the full amino acid sequence as a sliding window that may be adjusted in size rather than counting the frequency of occurrence of each individual amino acid within the entire amino acid sequence. For an amino acid sequence, EAAC is encoded in the following specific way.

1) First, set the window size (windows), such as windows = 5, which means that 5 consecutive amino acids are selected each time as the basis for calculating the frequency of each amino acid (including a dummy residue *X*), which is recorded as the set S1.

2) Calculate the frequency of each amino acid in S1, and the sum of the frequencies of all amino acids in S1 is 1.

3) Slide the window backward to Step 1).

4) Repeat Steps 2) and 3) until the last 5 amino acids in the amino acid sequence are boxed.

In this paper, windows = 5, then an amino acid sequence of length 35 ($L$ = 35) is encoded as a two-dimensional matrix with dimensions (35 – 5 + 1) × 21 = 31 × 21. It is worth noting that EAAC encoding degenerates into AAC encoding when windows = $L$ = 35, however when windows = 1, it produces the same output as one hot encoding.

#### 2.2.3. *K*-spaced encoding

In contrast to the encoding approach that focuses only on individual amino acids, *K*-spaced encoding turns its attention to amino acid pairs. This encoding approach extracts the feature information implied by amino acid sequences from another perspective, taking into account the significance of amino acid pairs in feature selection. The steps for *K*-spaced encoding are as follows.

1) First, the maximum interval $k$_max of amino acid pairs is specified, such as $k$_max = 2, which means that the amino acid pair interval $k$ is taken as 0, 1, 2, respectively. The distance between two amino acids (designated as A1 and A2, respectively) that make up an amino acid pair in an amino acid sequence is indicated by the amino acid interval. For example, if $k$ = 0, it means that A1 and A2 are adjacent to each other in the amino acid sequence (interval is 0). The sets S1, S2, and S3 are used to denote the cases of all amino acid pairings that may be derived from an amino acid sequence for $k$ = 0, 1 and 2, respectively.

2) Calculate the frequencies of all possible amino acid pairs (21 × 21 species) occurring in S1, S2 and S3, respectively, and the sum of the frequencies of all amino acid pairs is 1.

In this study, we use $k$_max = 9 and *K*-spaced encoding to create a two-dimensional matrix with a dimension of 21 × (9 + 1) 21 × = 210 × 21 using an amino acid sequence of length $L$ = 35.

### *2.3. Classification model*

The goodness of a classification model directly determines the accuracy of the classification results. In this paper, we adopt DenseNet, which has recently made great achievements in the field of deep learning. To the best of the author's knowledge, this is the first application of DenseNet to the prediction of glutarylation sites, although there have been some related studies on the prediction of other PTM sites, such as succinylation [31] and acetylation [32]. The overall structure of the model is shown in Figure 1, where 'dense block × 4 and transition × 3' means that there are 4 structurally identical dense blocks (all consist of a 4-layer dense blocks with growth rate = 32) and three intervening transition layers (all consist of an ELU, a 1 × 1 Conv 1D and a 2 × 2 average pooling) here. Each sequence of amino acids is transformed into a two-dimensional matrix after simple feature encoding, which is used as the input matrix **X** of the deep learning network. The input matrix **X** retains the original information while extracting the deep features after multiple layers of stacked dense convolutional blocks. Then, the stacked dense convolutional blocks were followed by SERNet which incorporating the attention residual learning method to maximize the retention of the original information and extract the final features. Finally, a sigmoid function was used to obtain the classification results.

### 2.3.1. DenseNet

DenseNet [33] proposed a radical dense connectivity mechanism that interconnects all layers, specifically noting that a later layer will preserve the information of all the layers before it. The reuse of feature maps from different layers enhances the propagation of features and improves the feature extraction capability of the network, while reducing the parameters for network training. The DenseNet connectivity mechanism is illustrated by the network architecture in Figure 1.

The primary module of DenseNet that implements the dense connectivity method is called dense block. Each layer in a dense block serves as the input for the subsequent layer and is connected to all the layers that came before it in the channel dimension. For a dense block with L layers, the total number of connections is $L$ ($L$ + 1)/2. The connections are highly dense, and the output of the Lth layer of the network is formulated as Eq (2).

$$x_L = H_L([x_0, x_1, \ldots, x_{L-1}]) \tag{2}$$

where $H_L(\cdot)$ represents the non-linear transformation function (non-linear transformation), it is a combined layer, which may contain, for example, BN, ReLU, convolution and other layers. In this research, ELU, and Conv 1D are employed together, as shown in Figure 1. Four dense blocks with the same structure were employed (a 4-layer dense block with a growth rate of k = 32).

The transition layer is yet another vital component of DenseNet. The transition layer connects two adjacent dense blocks and reduces the size of the feature map. In this paper, a total of 3 (4 – 1 = 3) transition layers are required as 4 dense blocks are finally selected. The transition layer consists of an ELU, a 1 × 1 Conv 1D and a 2 × 2 average pooling. The transition layer can act as a compression model. Assuming that the number of feature-maps in the dense block before transition is m, the transition layer can generate $\theta \times m$ feature-maps, where $\theta$ ($0 < \theta \leq 1$) is the compression rate. When $\theta = 1$, there is no compression and no change in the number of feature-maps following the transition layer. In this study, $\theta = 0.5$, meaning that the number of feature-maps drops to half from its initial value following the transition layer.

### 2.3.2. SENet combined with attention residual learning

In this work, we suggest a SENet incorporating the attention residual learning method, which is explained as follows: The standard procedure in SENet is to first perform global average pooling in the spatial dimension (which is called Squeeze), after which channel attention is learned through two fully connected layers, normalized with sigmoid (which is called Excitation), and then multiplied with the original matrix A to produce the feature matrix B after weighting in the spatial dimension. We suggest incorporating attention residual learning into SENet in this paper. The specific method, which in theory is based on the Shortcut Connection concept of ResNet, is to add the newly acquired matrix B to the original matrix A following excitation, so we call it SERNet for short, as shown in the SERNet section of Network Architectures of Figure 1. This can meet the goals of both acquiring the spatial dimensional weighting and maintaining the original matrix information. In the experiments, the two fully connected layers in Excitation have an output dimension of 96 and 6, and the activation functions are ReLU and sigmoid, respectively.

### 2.3.3. Focal loss function

The focal loss function is employed in this study to make up for the shortcomings of the traditional cross-entropy loss function, which pays insufficient attention to minority samples due to the significant imbalance in the number of glutarylation sites and non-glutarylation sites samples.

The focal loss function was proposed by Lin et al. [34] to address the problem of model performance degradation caused by category imbalance in sense object detection, which is also a common problem in PTM sites prediction. Therefore, the focal loss function is introduced in this paper to give a new idea to the data imbalance problem arising in the PTM sites prediction task. The focal loss formulation is shown in Eqs (3) and (4).

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases} \tag{3}$$

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log (p_t) \tag{4}$$

where $y$ denotes the sample label, $p$ ($p\epsilon[0,1]$) denotes the probability that the predicted sample belongs to 1, and $\alpha_t$ ($\alpha_t\epsilon[0,1]$) in Eq (4) is called the weighting factor, controlling the shared weight of positive and negative samples on the total loss, and its smaller value represents the lower weight of negative samples. In this paper, $\alpha_t = 0.8, \gamma = 2$ can achieve the better results.

### 2.3.4. Performance evaluation

Scientific and general metrics are the basis for measuring the performance of different models. The prediction problem in this study can be thought of as a dichotomous classification problem in machine learning, and the confusion matrix is the commonly used evaluation metric for classification tasks in this field. In this paper, the four metrics in the confusion matrix are used to evaluate the performance of the classification model, and the specific metrics [35,36] are shown in Eq (5).

$$\begin{cases} Re, Sn = \frac{TP}{TP+FN} \\ Sp = \frac{TN}{TN+FP} \\ Acc = \frac{TP+TN}{TP+TN+FP+FN} \\ MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \end{cases} \tag{5}$$

where TP, TN, FP and FN are abbreviations of True positive, True negative, False positive, and False negative, respectively. TP and TN stand for the number of samples in positive and negative categories that were properly predicted, respectively, whereas FP and FN stand for the number of samples in positive and negative categories that were mistakenly predicted, respectively. Additionally, the receiver operating characteristic (ROC) curve [37]—which is also employed as one of the model evaluation metrics in this paper—is an excellent technique to directly illustrate the prediction outcomes.

In this work, the 10-fold CV is used to evaluate model performance and make prediction findings more stable for comparison with other predictors. The 10-fold CV will be trained ten times, and each time the model is trained, the input data is divided into ten sets (D1, D2, ..., D10). The set D1 is used for model performance testing, and the other nine sets (D2–D10) are used for model training. The process repeats itself until each subset has served as a testing data set a single time.

## 3. Results and discussion

In this section, comparison studies of the feature encoding methods and the functional modules used for the model (such as the focal loss function) are discussed. It is important to note that the models used in the experiments were all those described in Section 2.3 and that the functional module comparison experiments were carried out so that just one condition was altered while the others were left unchanged. Specifically, the model in Experiment 3.1 employed four dense blocks of the same structure and the focal loss function as the loss function. The number of dense block layers was 4 when the loss function comparison experiments were conducted, and the focal loss function was used when the comparison experiments with different numbers of dense block layers were undertaken.

### *3.1. Ablation experiments of feature extraction methods*

We performed ablation experiments to determine which of the three feature encoding methods—or which combination of the three—is best suited as input to this model. The experimental results are displayed in Table 2, with the best results denoted in bold. All results are based on 10-fold cross-validation on the training set. The first three rows of Table 2 determine which feature encoding method is chosen for each experiment. If there is a mark "√" in the corresponding row of each encoding method, it indicates that the method is chosen for this experiment; if there is not, it indicates that method is not chosen. The remaining 7 columns in Table 2 except the first column represent 7 experiments ($C_3^1 + C_3^2 + C_3^3 = 7$). The combination approach between the encoding techniques has been discussed in Section 2.2.

Table 2 shows that when used as an input to a deep learning model, one hot encoding alone offers unparalleled advantages over other encoding methods, which can be easily seen from the fact that its evaluation metrics are higher than those of other individual or combined encoding methods when used alone.

In this regard, we have reason to believe that simple but highly discriminative encoding is more suitable as input for deep learning models that are very powerful in automatically extracting deep features, such as the one hot encoding used in this experiment. This finding might offer suggestions for developing deep learning-based predictions in the future. The one hot encoding is easily accessible from a convenience standpoint, which makes it easier for researchers to focus mainly on the construction of the model.

**Table 2.** Ablation experiments of the feature encoding methods.

| One hot | √ | | | √ | √ | | √ |
|---|---|---|---|---|---|---|---|
| EAAC | | √ | | √ | | √ | √ |
| *K*-Spaced | | | √ | | √ | √ | √ |
| Sn | **0.89** | 0.84 | 0.82 | 0.65 | 0.66 | 0.68 | 0.65 |
| Sp | 0.62 | 0.58 | 0.43 | **0.65** | 0.62 | 0.51 | 0.65 |
| ACC | **0.65** | 0.61 | 0.47 | 0.65 | 0.63 | 0.54 | 0.65 |
| MCC | **0.33** | 0.27 | 0.16 | 0.22 | 0.21 | 0.14 | 0.22 |
| AUC | **0.80** | 0.75 | 0.62 | 0.72 | 0.71 | 0.64 | 0.72 |

### *3.2. Comparison of focal loss function and cross-entropy loss function*

We employed two loss functions in this section to independently predict the glutarylation sites in order to verify if the focal loss function is effective for the imbalanced data. The results are shown in Figure 2. The experimental findings demonstrated that the focal loss function is more sensitive to minority class samples than the cross-entropy loss. After employing the focal loss function, both Sn and MCC improved, reaching 0.78 and 0.31, respectively.

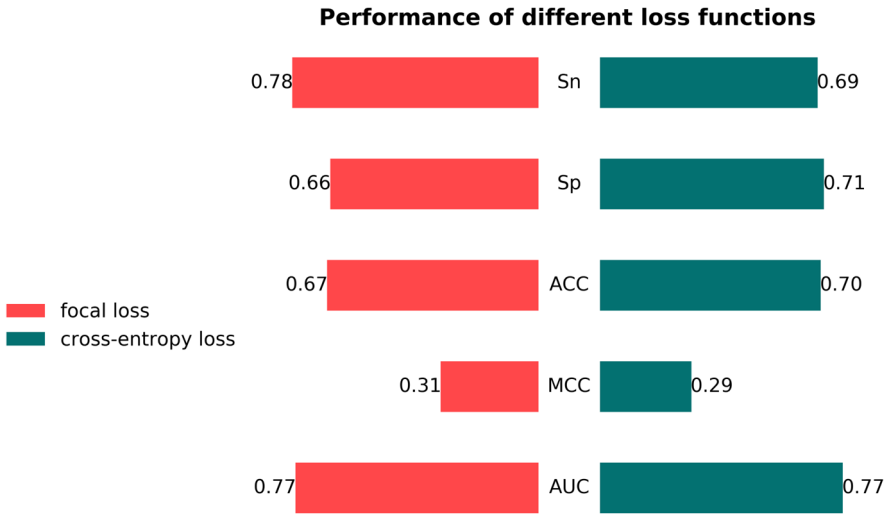


**Figure 2.** Comparison of focal loss function and cross-entropy loss function.

### *3.3. Comparison of the number of dense blocks*

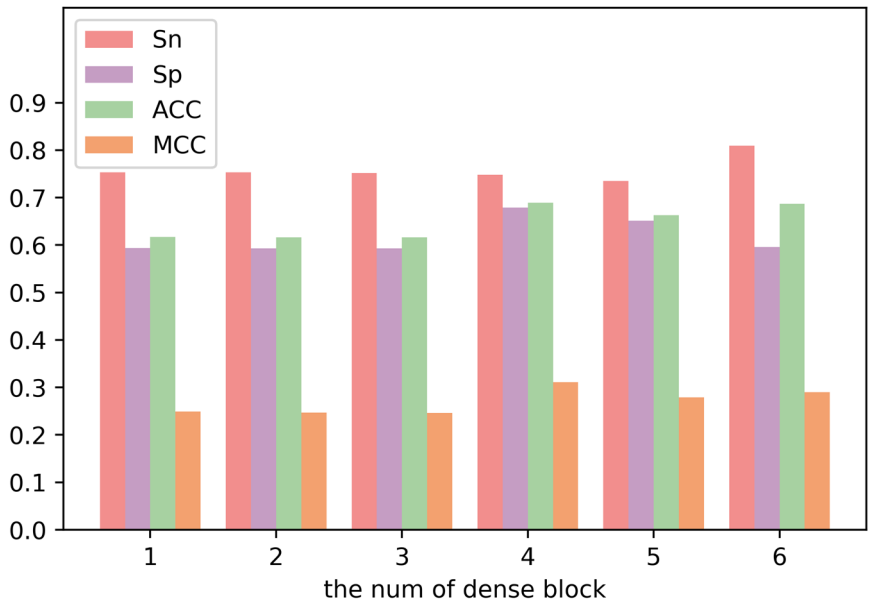


**Figure 3.** Comparison of the number of dense blocks.

This study evaluates the model performance under various numbers of dense blocks since the number of dense blocks in DenseNet is also a significant factor impacting the model performance. In Figure 3, it is intuitively clear that when there are four dense blocks stacked together, the Sp, ACC, and MCC of the model are greater than those in the other scenarios. As a result, in this paper, four dense blocks are chosen to build DenseNet.

### *3.4. Comparison with existing predictors*

Given the availability of existing predictors and the rigorousness of the comparison, we chose two predictors, PUL-GLU and GlutPred, which use the same dataset as this paper, for comparison. Table 3 shows the 10-fold cross-validation performance of DeepDN_iGlu and other predictors. It is worth noting that PUL-GLU screens the training samples (removing some of the negative samples), resulting in a lower imbalance ratio. Thus, our study focuses on a comparison with GlutPred to discuss the predictive performance of the model while maintaining the original ratio of positive and negative samples.

According to Table 3, the Sn of DeepDN_iGlu is typically 9% higher than the other two predictors at the cost of both Sp and ACC falling. MCC is a measure of great importance to the imbalanced data. As can be observed, even though Sp and ACC reduced, MCC and AUC were essentially the same as GlutPred, proving that the predictor proposed in this research, as opposed to the other two predictors, concentrates more on the accurate distinction of samples of minority category (samples with glutarylation site).

**Table 3.** 10-fold cross-validation performance of DeepDN_iGlu and other predictors.

| Predictor | Sn (%) | Sp (%) | ACC (%) | MCC | AUC |
|---|---|---|---|---|---|
| PUL-GLU | 66.56 ± 0.73 | 86.43 ± 0.28 | 79.88 ± 0.29 | 0.5384 ± 0.069 | - |
| GlutPred | 64.80 ± 0.99 | 76.60 ± 0.28 | 74.90 ± 0.32 | 0.3194 ± 0.087 | 78.06 ± 0.29 |
| DeepDN-iGlu | **79.45** ± 8.29 | 63.74 ± 4.44 | 66.00 ± 3.61 | 0.3080 ± 0.068 | 77.25 ± 0.04 |

The prediction performance of the three predictors on the same independent test set is shown in Table 4. With Sn at 89.29%, 27.5% higher than GlutPred, and MCC at 0.3309, both significantly higher than the other two predictors. AUC also outperformed GlutPred by roughly 3%. Figure 4 provides a direct-viewing comparison of the two predictors. This shows that the model put forward in this research has a stronger capacity for generalization and better corroborates the finding seen on the 10-fold CV.

**Table 4.** Comparison with the other predictors on the independent test dataset.

| Predictor | Sn (%) | Sp (%) | ACC (%) | MCC | AUC |
|---|---|---|---|---|---|
| PUL-GLU | 58.93 | 78.97 | 76.65 | 0.2785 | - |
| GlutPred | 51.79 | 78.50 | 75.41 | 0.2238 | 0.7663 |
| DeepDN-iGlu | **89.29** | 61.97 | 65.15 | **0.3309** | **0.8087** |

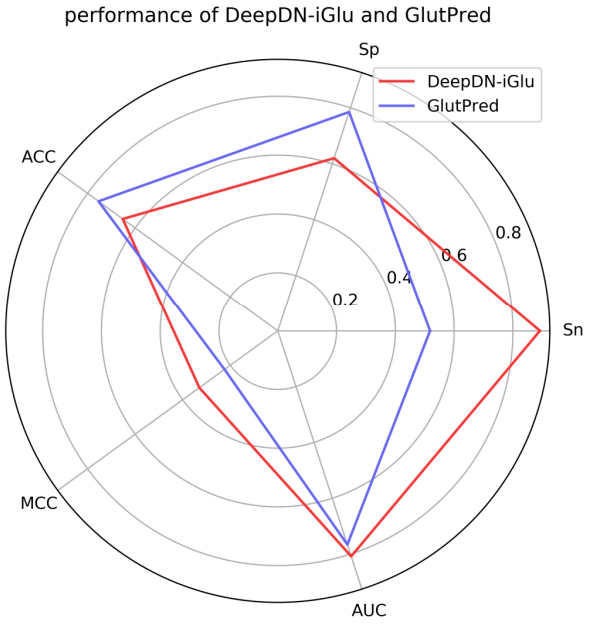


**Figure 4.** Comparison with GlutPred on the independent test dataset.

### *3.5. An available web server for DeepDN_iGlu*

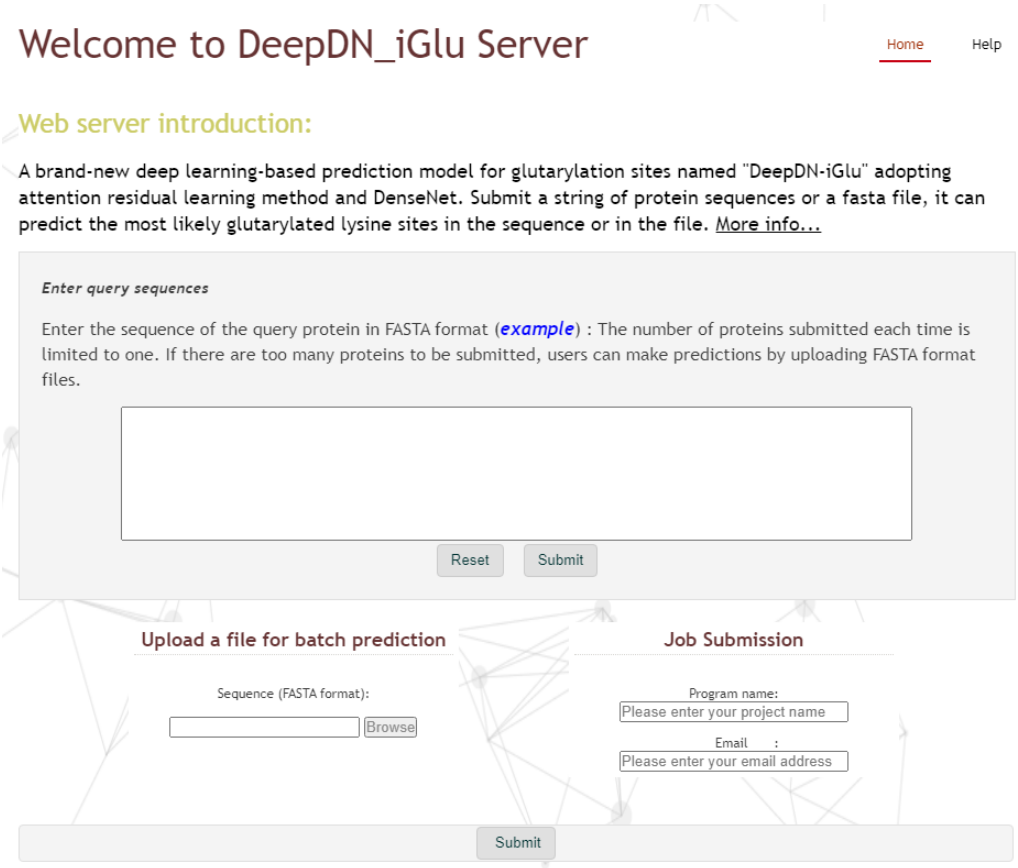


**Figure 5.** Screenshot of DeepDN_iGlu web server.

To facilitate the access to glutarylation site prediction data, DeepDN_iGlu has been implemented as an available web server (https://bioinfo.wugenqiang.top/~smw/DeepDN_iGlu/). A screenshot of the DeepDN_iGlu server's interface is shown in Figure 5. The DeepDN_iGlu server is primarily composed of three sections: "Introduction", "Enter query sequences" and "Job Submission". "Introduction" provides a description of the composition of DeepDN_iGlu, functions and usage. To keep the main interface straightforward, users can click the "More info" button to acquire more detailed information on how to use it. In order to produce a prediction, the user can enter one or more protein sequences in FASTA format using "Enter query sequences" and then make a prediction by clicking the "submit" button. The user can view the specific requirements for the input sequence by selecting the "example" button. With "job submission", users have an easy way to obtain a lot of prediction data; all they need to do is enter their email address, project name, and forecast file in the appropriate areas, press the "submit" button, and they will acquire the prediction results file in their email.

## 4. Conclusions

In this study, we developed a brand-new deep learning-based prediction model for glutarylation sites named DeepDN_iGlu adopting attention residual learning method and DenseNet. According to the experimental results and the various related evaluation data, DeepDN_iGlu has obtained a satisfactory prediction performance with Sn, Sp, ACC, MCC and AUC of 79.45%, 63.74%, 66.00%, 0.3080 and 0.77 at 10-fold cross-validation, while 89.29%, 61.97%, 65.15%, 0.33 and 0.80 accordingly on the independent test set. Additionally, the MCC of 0.33 on the independent test set shows that DeepDN_iGlu has better generalization ability. Another key benefit of DeepDN_iGlu is the feature encoding, which is quite straightforward and only requires one-hot encoding. Finally, a user-friendly web server was designed to make it convenient for researchers to obtain data on lysine glutarylation sites prediction.

To a certain extent, the benefits of deep learning for a big number of data cannot be sufficiently utilized due to the little amount of data in the present glutarylation sites. The training samples are further reduced in the 10-fold cross-validation process, so it causes the model to be slightly less effective for the 10-fold cross-validation than on the independent test set, which will be a problem we need to solve in the future. But as glutarylation research advances, the contradiction of sparse data will gradually disappear, opening up more possibilities for prediction models based on deep learning of glutarylation sites.

## Acknowledgments

The authors are grateful for the constructive comments and suggestions made by the reviewers. This work was partially supported by the National Natural Science Foundation of China (Nos. 61761023, 62162032 and 31760315), the Natural Science Foundation of Jiangxi Province, China (Nos. 20202BABL202004 and 20202BAB202007), the Scientific Research Plan of the Department of Education of Jiangxi Province, China (GJJ190695). These funders had no role in the study design, data collection and analysis, decision to publish or preparation of manuscript.

## Availability of data and materials

The dataset and source code used in this study can be easily derived from https://github.com/Michae-l/DeepDN-iGlu.

## Conflict of interest

The authors declare that they have no competing interests.

## References


1. E. Furuya, K. Uyeda, Regulation of phosphofructokinase by a new mechanism. An activation factor binding to phosphorylated enzyme, *J. Biol. Chem.*, **255** (1980), 11656–11659. https://doi.org/10.1016/s0021-9258(19)70181-1
2. C. Lu, C. B. Thompson, Metabolic regulation of epigenetics, *Cell Metab.*, **16** (2012), 9–17. https://doi.org/10.1016/j.cmet.2012.06.001
3. M. Tan, C. Peng, K. A. Anderson, P. Chhoy, Z. Xie, L. Dai, et al., Lysine glutarylation is a protein posttranslational modification regulated by SIRT5, *Cell Metab.*, **19** (2014), 605–617. https://doi.org/10.1016/j.cmet.2014.03.014
4. S. Ahmed, A. Rahman, M. Hasan, A. Mehedi, S. Ahmad, S. M. Shovan, Computational identification of multiple lysine PTM sites by analyzing the instance hardness and feature importance, *Sci. Rep.*, **11** (2021), 18882. https://doi.org/10.1038/s41598-021-98458-y
5. G. S. McDowell, A. Philpott, New insights into the role of ubiquitylation of proteins, *Int. Rev. Cell Mol. Biol.*, **325** (2016), 35–88. https://doi.org/10.1016/bs.ircmb.2016.02.002
6. L. D. Vu, K. Gevaert, I. De Smet, Protein language: post-translational modifications talking to each other, *Trends Plant Sci.*, **23** (2018), 1068–1080. https://doi.org/10.1016/j.tplants.2018.09.004
7. R. S. P. Rao, N. Zhang, D. Xu, I. M. Moller, CarbonylDB: a curated data-resource of protein carbonylation sites, *Bioinformatics*, **34** (2018), 2518–2520. https://doi.org/10.1093/bioinformatics/bty123
8. M. Wang, X. Cui, B. Yu, C. Chen, Q. Ma, H. Zhou, SulSite-GTB: identification of protein S-sulfenylation sites by fusing multiple feature information and gradient tree boosting, *Neural Comput. Appl.*, **32** (2020), 13843–13862. https://doi.org/10.1007/s00521-020-04792-z
9. X. Liu, L. Wang, J. Li, J. Hu, X. Zhang, Mal-Prec: computational prediction of protein Malonylation sites via machine learning based feature integration, *BMC Genomics*, **21** (2020), 812. https://doi.org/10.1186/s12864-020-07166-w
10. K. Y. Huang, F. Y. Hung, H. J. Kao, H. H. Lau, S. L. Weng, iDPGK: characterization and identification of lysine phosphoglycerylation sites based on sequence-based features, *BMC Bioinf.*, **21** (2020), 568. https://doi.org/10.1186/s12859-020-03916-5
11. S. Ahmed, M. Kabir, M. Arif, Z. U. Khan, D. J. Yu, DeepPPSite: a deep learning-based model for analysis and prediction of phosphorylation sites using efficient sequence information, *Anal. Biochem.*, **612** (2021), 113955. https://doi.org/10.1016/j.ab.2020.113955
12. N. Thapa, M. Chaudhari, S. McManus, K. Roy, R. H. Newman, H. Saigo, et al., DeepSuccinylSite: a deep learning based approach for protein succinylation site prediction, *BMC Bioinf.*, **21** (2020), 63. https://doi.org/10.1186/s12859-020-3342-z

13. Z. Ju, J. J. He, Prediction of lysine glutarylation sites by maximum relevance minimum redundancy feature selection, *Anal. Biochem.*, **550** (2018), 1–7. https://doi.org/10.1016/j.ab.2018.04.005
14. Y. Xu, Y. Yang, J. Ding, C. Li, iGlu-Lys: A Predictor for lysine glutarylation through amino acid pair order features, *IEEE Trans. Nanobiosci.*, **17** (2018), 394–401. https://doi.org/10.1109/TNB.2018.2848673
15. K. Y. Huang, H. J. Kao, J. B. K. Hsu, S. L. Weng, T. Y. Lee, Characterization and identification of lysine glutarylation based on intrinsic interdependence between positions in the substrate sites, *BMC Bioinf.*, **19** (2019), 13–25. https://doi.org/10.1186/s12859-018-2394-9
16. H. J. Al-Barakati, H. Saigo, R. H. Newman, D. B. KC, RF-GlutarySite: a random forest based predictor for glutarylation sites, *Mol. Omics*, **15** (2019), 189–204. https://doi.org/10.1039/c9mo00028c
17. M. E. Arafat, M. W. Ahmad, S. M. Shovan, A. Dehzangi, S. R. Dipta, M. A. M. Hasan, et al., Accurately predicting glutarylation sites using sequential Bi-Peptide-Based evolutionary features, *Genes*, **11** (2020), 1023. https://doi.org/10.3390/genes11091023
18. L. Dou, X. Li, L. Zhang, H. Xiang, L. Xu, iGlu_AdaBoost: identification of lysine glutarylation using the adaboost classifier, *J. Proteome Res.*, **20** (2020), 191–201. https://doi.org/10.1021/acs.jproteome.0c00314
19. J. Jia, Z. Liu, X. Xian, B. Liu, K. C. Chou, pSuc-Lys: Predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach, *J. Theor. Biol.*, **394** (2016), 223–230. https://doi.org/10.1016/j.jtbi.2016.01.020
20. P. Kelchtermans, W. Bittremieux, K. De Grave, S. Degroeve, J. Ramon, K. Laukens, et al., Machine learning applications in proteomics research: how the past can boost the future, *Proteomics*, **14** (2014), 353–366. https://doi.org/10.1002/pmic.201300289
21. L. Dou, F. Yang, L. Xu, Q. Zou, A comprehensive review of the imbalance classification of protein post-translational modifications, *Briefings Bioinf.*, **22** (2021), bbab089. https://doi.org/10.1093/bib/bbab089
22. Z. Ju, S. Y. Wang, Computational identification of lysine glutarylation sites using positive-unlabeled learning, *Curr. Genomics*, **21** (2020), 204–211. https://doi.org/10.2174/1389202921666200511072327
23. B. Wen, W. F. Zeng, Y. Liao, Z. Shi, S. R. Savage, W. Jiang, et al., Deep learning in proteomics, *Proteomics*, **20** (2020), 1900335. https://doi.org/10.1002/pmic.201900335
24. S. C. Pakhrin, S. Pokharel, H. Saigo, D. B. Kc, Deep learning-based advances in protein posttranslational modification site and protein cleavage prediction, in *Computational Methods for Predicting Post-Translational Modification Sites*, Humana Press, (2022), 285–322. https://doi.org/10.1007/978-1-0716-2317-6_15
25. S. Naseer, R. F. Ali, Y. D. Khan, P. D. D. Dominic, iGluK-Deep: computational identification of lysine glutarylation sites using deep neural networks with general pseudo amino acid compositions, *J. Biomol. Struct. Dyn.*, **2021** (2021), 1–14. https://doi.org/10.1080/07391102.2021.1962738
26. C. M. Liu, V. D. Ta, N. Q. K. Le, D. A. Tadesse, C. Shi, Deep neural network framework based on word embedding for protein glutarylation sites prediction, *Life*, **12** (2022), 1213. https://doi.org/10.3390/life12081213

27. H. Xu, J. Zhou, S. Lin, W. Deng, Y. Zhang, Y. Xue, PLMD: an updated data resource of protein lysine modifications, *J. Genet. Genomics*, **44** (2017), 243–250. https://doi.org/10.1016/j.jgg.2017.03.007
28. W. Li, A. Godzik, Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences, *Bioinformatics*, **22** (2006), 1658–1659. https://doi.org/10.1093/bioinformatics/btl158
29. Y. Huang, B. Niu, Y. Gao, L. Fu, W. Li, CD-HIT Suite: a web server for clustering and comparing biological sequences, *Bioinformatics*, **26** (2010), 680–682. https://doi.org/10.1093/bioinformatics/btq003
30. K. C. Chou, Prediction of signal peptides using scaled window, *Peptides*, **22** (2001), 1973–1979. https://doi.org/10.1016/S0196-9781(01)00540-X
31. H. Wang, H. Zhao, Z. Yan, J. Zhao, J. Han, MDCAN-Lys: a model for predicting succinylation sites based on multilane dense convolutional attention network, *Biomolecules*, **11** (2021), 872. https://doi.org/10.3390/biom11060872
32. H. Wang, Z. Yan, D. Liu, H. Zhao, J. Zhao, MDC-Kace: A model for predicting lysine acetylation sites based on modular densely connected convolutional networks, *IEEE Access*, **8** (2020), 214469–214480. https://doi.org/10.1109/access.2020.3041044
33. G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, USA, (2017), 2261–2269. http://doi.org/10.1109/CVPR.2017.243
34. T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, (2017), 2999–3007. https://doi.org/10.1109/ICCV.2017.324
35. M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, *Inf. Process. Manage.*, **45** (2009), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002
36. S. Boughorbel, F. Jarray, M. El-Anbari, Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric, *PLoS One*, **12** (2017), e0177678. https://doi.org/10.1371/journal.pone.0177678
37. T. Fawcett, An introduction to ROC analysis, *Pattern Recognit. Lett.*, **27** (2006), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010



AIMS Press