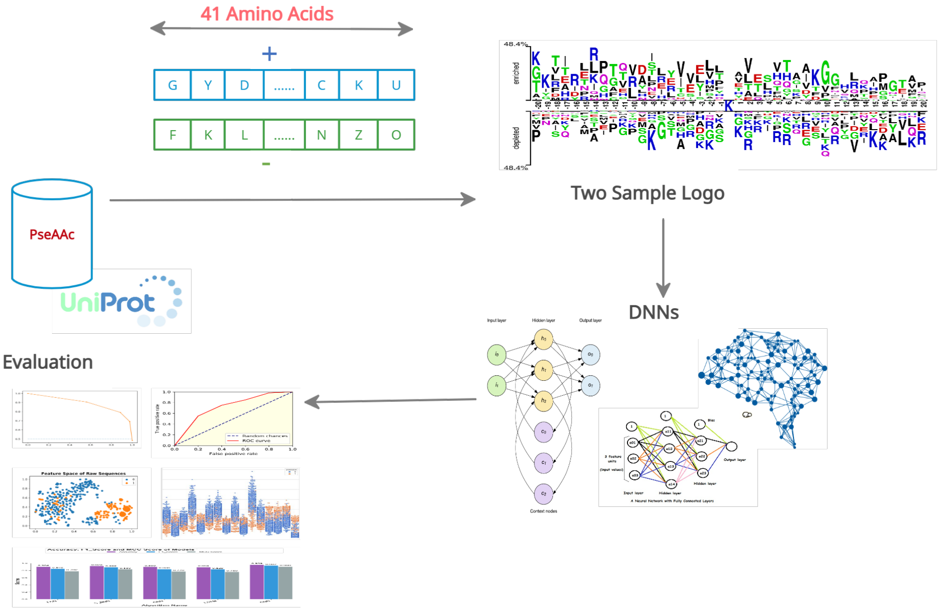
Graphical Abstract

iGluK-Deep:Computational Identification of lysine glutarylation sites using deep neural networks with general Pseudo Amino Acid Compositions

Sheraz Naseer, Rao Faizan Ali, Yaser Daanial Khan, P.D.D Dominic



iGluK-Deep:Computational Identification of lysine glutarylation sites using deep neural networks with general Pseudo Amino Acid Compositions

Sheraz Naseer, Rao Faizan Ali, Yaser Daanial Khan, P.D.D Dominic

[a]*Department of Computer Science, University of Management and Technology, Lahore, 54728, Pakistan*
[b]*Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Perak Darul Ridzuan, 32610 , Malaysia*
[c]*Department of Computer Science, University of Management and Technology, Lahore, 54728, Pakistan*
[d]*Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Perak Darul Ridzuan, 32610 , Malaysia*

Abstract

Lysine glutarylation is a post-translation modification which plays an important regulatory role in a variety of physiological and enzymatic processes including mitochondrial functions and metabolic processes both in eukaryotic and prokaryotic cells. This post-translational modification influences chromatin structure and thereby results in global regulation of transcription, defects in cell-cycle progression, DNA damage repair, and telomere silencing. To better understand the mechanism of lysine glutarylation, its identification in a protein is necessary, however, experimental methods are time-consuming and labor-intensive. Herein, we propose a new computational prediction approach to supplement experimental methods for identification of lysine glutarylation site prediction by deep neural networks and Chou's Pseudo Amino Acid Composition (PseAAC). We employed well-known deep neural networks for feature representation learning and classification of peptide sequences. Our approach opts raw pseudo amino acid compositions and obsoletes the need to separately perform costly and cumbersome feature extraction and selection. Among the developed deep learning-based predictors, the standard neural network-based predictor demonstrated highest scores in terms of accuracy and all other performance evaluation measures and outperforms majority of previously reported predictors without requiring expensive feature extraction process.

Keywords: Deep features, Deep Neural networks, Glutarylation, Lysine, PseAAC.

1. Introduction

Cells are constantly exposed to diverse stressors and variations in energy supply under physiological conditions, which leads to fluctuations in cellular energy status. Responding to these dynamic changes requires cells to use multiple adaptive strategies for maintaining metabolic homeostasis. These strategies include regulation of energy producing pathways, alterations of epigenetic marks, modulation of metabolic enzymes activities using metabolites and protein post translational modifications (PTMs) [1]. PTMs can extend the chemical repertoire of the standard amino acids by modifying an existing functional group or introducing a new one such as amide [2] or phosphate [3, 4], and hence play an important role to form the mature protein product [5]. Different types of PTMs exist, and their roles vary from protein folding, function regulation, catalytic activity to signal transduction.

Lysine Glutarylation (K-glu), identified by Tan et al. [6], is an evolutionary conserved PTM which is characterized by the addition of a glutaryl group (five carbons) to a lysine residue of a protein. K-glu plays a regulatory role in a variety of physiological and enzymatic processes including mitochondrial functions and metabolic processes including amino acid metabolism, fatty acid metabolism, co-enzyme metabolism, mitochondrial metabolism and cellular respiration [6, 7] in eukaryotic and prokaryotic cells. Tan et al. [6] also showed that glutarylation of carbamoyl phosphate synthase 1 (CPS1) inhibits its activity but can be reversed by Sirtuin5 (SIRT5). An association of K-glu with progressive motility of human sperms was shown by Cheng et al. [8] who showed that K-glu occurs in multiple proteins located in the tail of human sperm and is positively correlated with progressive motility of human sperm, indicating its important role in maintaining sperm motility. In addition, it was demonstrated by Zhou et al. [9] that the K-glu widely exists in mammalian serum proteins, and provides insights into the novel mechanism of acute myocardial infarction. K-glu was also identified as a PTM on Human core Histones, basic proteins found in eukaryotic cell nuclei to pack and order the DNA, by Bao et al. [10]. The aforementioned contribution [10] reported that an evolutionarily conserved K-glu at histone H4K91 destabilizes nucleosome in vitro using semi-synthetic glutarylated histones. Bao et al. [10] demonstrated that K-glu influences chromatin structure and thereby results in a global regulation of transcription and defects in cell-cycle progression, DNA damage repair, and telomere silencing. In addition, glutarylated Histones are primarily enriched at promoter regions of highly expressed genes in mammalian cells. Furthermore, it was shown by Bao et al [10] that the down-regulation of glutarylated Histones is tightly associated with chromatin condensation during mitosis and in response to DNA damage.

Owing to the aforementioned facts, it behooves us to better understand the molecular mechanisms of glutarylation and a fundamental step is the identification of glutarylation sites. Although multiple large-scale in-vivo, ex-vivo, and in-vitro methods such as immunoblot and mass spectrometry [6], have been applied to detect glutarylation sites, these experimental methods are time-consuming and labor-intensive. A vast majority of lysine glutarylated substrates and respective glutarylation sites are yet to be discovered. Research community has applied computational methods to solve problems in proteomics and genomics using various data science and machine learning techniques [3, 4], [11, 12, 13, 14, 15, 16, 17, 18, 19]. Similarly, research contributions have been done by using different feature extraction techniques to identify the potential glutarylation sites [20, 21, 22]. Although these contributions show promising results but most of the techniques use human-engineered features. According to Lecun et al. [23] human features have certain limitations as they are time-consuming to calculate due to absence of a feedback mechanism between prediction model and feature extraction mechanism. This makes it impossible to ascertain the quality of features, to develop an effective predictor, until such a candidate model is developed and evaluated. Additionally, development of human-engineered features require domain knowledge and expert human intervention which is sometimes hard to come by [23].

Lecun et al. [23] proposed deep learning to overcome aforementioned limitations. Deep learning is the study of different deep neural network architectures (DNNs) which have enabled many breakthroughs in different scientific disciplines including computer vision, image processing and information security [24, 25] to mention a few. In essence, all DNNs consist of multiple layers of basic mathematical functions, dubbed as neurons, which transform the inputs layer by layer, until the transformed input reaches to last layer of the neural network which uses this transformed input to make the predictions. Each DNN layer receives input from the upper layer and translates it into some representation that subsequent layers use. Each such transformation can be considered as a representation of input data. DNN layers transform their input non-linearly, producing hierarchically abstract, task-specific representations that are insensitive to unimportant variations, but sensitive to significant features. With appropriate DNN training, the representation generated by the last hidden layer, nearest to the output layer of DNN, is so effective in recognizing hidden patterns of input that it is used by the output layer to make predictions. Hence the DNNs provide us a means to generate efficient, task specific and effective deep features which does not require human intervention, domain knowledge and laborious feature selection process [26].

Chou's PseAAC has been used by many researchers for solving PTM problems [27, 28, 29, 30]. But most contributions use PseAAC sequences to extract features which are then used to develop Protein and PTM identification models. As discussed earlier, the human engineered features have limitations which can be addressed by use of DNNs. In this study, by combining deep neural networks with Chou's Pseudo Amino Acid Composition [31] with deep neural networks, we propose an improved predictor for identifying K-glu sites in proteins. For both the tasks of learning a feature representation of peptide sequences and performing classifications, we used well-known DNNs, including Standard neural network (FCN), three variants of recurrent neural networks (RNNs) and convolutional neural network (CNN). We used the 5-step rule of Chou [32] for this purpose which is opted widely in series of publications [33, 4, 34, 15, 33, 14, 28, 35, 17, 19, 36, 29]. Steps of Chou's methodology include (i) Benchmark dataset collection (ii) Preprocessing of raw PseAAC sequences to make them amenable to Machine learning algorithms and extract human engineered features (iii) Implementation and training of prediction model (iv) Evaluation of results and (v) deployment of predictor using webserver. The adopted methodology takes advantage of DNNs' inherent feature learning capabilities and incorporates both feature extraction and model training steps to learn efficient feature representations of constituent Pseudo Amino Acid Compositions (PseAAC) of peptide samples [31]. Multiple candidate DNN-based prediction models are trained, in this study, using aforementioned DNN algorithms to obtain an optimal model for computational identification of K-glu sites. Performance of models developed in this study is evaluated among themselves and with literature using well-known parameters of model evaluation.

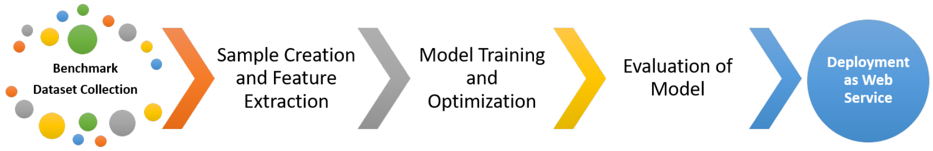

Benchmark Dataset Collection
Sample Creation and Feature Extraction
Model Training and Optimization
Evaluation of Model
Deployment as Web Service

Figure 1: Chou's 5-step rule for PTM prediction

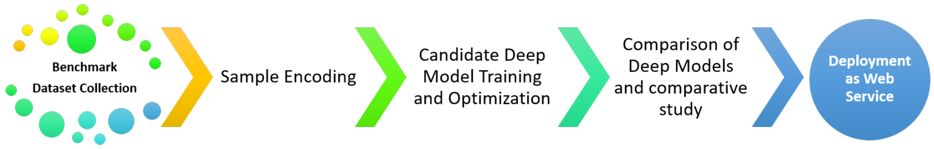

Benchmark Dataset Collection
Sample Encoding
Candidate Deep Model Training and Optimization
Comparison of Deep Models and comparative study
Deployment as Web Service

Figure 2: Methodology adopted for developing Lysine Glutarylation Site Prediction Models

2. Materials and Methods

This study's methodology is based on Chou's 5-step rule [32] comprising of five distinct stages as shown in figure 1. As discussed earlier and shown in figure 1, steps of Chou's methodology include, benchmark dataset collection, preprocessing of raw PseAAC sequences to make them amenable to Machine learning algorithms and extract human engineered features, model training for PTM site prediction, results evaluation and model deployment using webserver to ensures that the research community is able to use the proposed advancements in discipline. The approach adopted in this study is gleaned from the Chou's five-step rule. Instead of relying on human engineered features, our methodology combines the feature extraction and model training step using DNNs. Once a DNN model is sufficiently trained, the intermediate layers of DNN transform PseAAC sequences to meaningful deep representations while output layer of DNN perform prediction using deep representation learned by earlier layers. Since both the representation learning subsystem and site prediction subsystem work in unison, the optimizer module of DNN uses the loss score of the output layer as the feedback signal to improve both above-mentioned DNN subsystems. This methodology is shown in figure 2. For this research, Several DNN-based models were trained and evaluated using standard performance evaluators of prediction models to obtain an optimal model for predicting K-glu sites. The emphasis of this section is on the first three steps of methodology shown in figure 2, while the last two steps of the suggested methodology are elaborated in following sections.

2.1. Benchmark Dataset Collection

We used the advanced search and annotation capabilities of UniProt to create benchmark dataset for this analysis [37]. Quality of benchmark dataset was ensured by selecting protein sequences where K-glu was detected and investigated experimentally. Using Chou's PseAAC [31] a peptide sequence with a K-glu positive site can be shown as follows:

$$f_{\pi}(P) = A_{-\pi}A_{-(\pi-1)}\ldots A_{-2}A_{-1}KA_{+1}A_{+2}\ldots A_{+(\pi-1)}A_{+\pi}$$

Where K reflects the positive K-glu amino acid 'Lysine' and 'A' reflects the positive site's neighboring amino acid residues. The symbol π denotes indexes of PseAAC sequence residues where the left-hand residues of K-glu site are at $-\pi$ indexes and right-hand side indexes are at respective π indexes.

We extracted the positive and negative samples of length μ from experimentally verified proteins PseAAC sequences. Based on the empirical observations, For both positive and negative samples, length of sequence, denoted by μ, is fixed at 41 for current study. Positive sequences were produced by fixing the index of k-glu site at $\pi = 21$ and attaching twenty leftside and twenty rightside neighbor residues of the site to achieve the standard length sequence. For positive samples with $\mu < 41$, symbol X was used as a dummy amino acid residue and attached on both sides of the sequence to achieve standard length. Same methodology was adopted to extract negative samples from acquired protein PseAAC sequences. The above procedure resulted in 954 positive and 1451 negative samples and the reference data set comprised a total of 2405 peptide samples. Since the natural ratio of both classes is imbalanced in K-glu identification problem, we opted to preserve this ratio rather than balancing the class ratio. As any model, which assumes a balance in two classes when this is not the case, is bound to perform poorly when deployed in real world because the assumption of class balance will become false. To assure the real world performance of models proposed in this study, we chose to preserve the actual class ratio between samples of both positive and negative classes. Application of CD-Hit to remove homology resulted in severely reduced dataset with 49 positive sequences and 89 negative sequences, even at threshold of 0.8, so we chose not to remove homologous

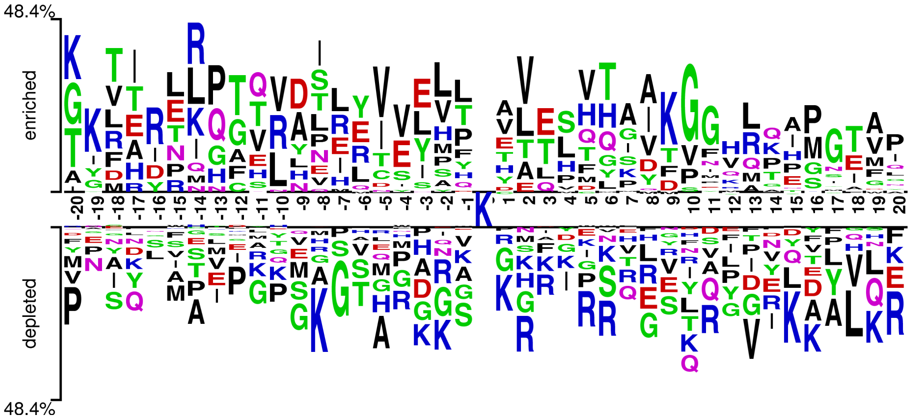


Figure 3: Two sample logo of Benchmark Dataset

samples. The final benchmark dataset of 2405 samples can be represented as follows:

$$K = K^{+} \cup K^{-}$$

Where K^{+} represents positive 954 samples and K^{-} represents negative 1451 samples. The class proportions of the positive and negative comparison classes were 40% and 60% respectively. The dataset is available at https://mega.nz/folder/J5FFkQ5Y#3to9NOaDlflVIjSlKukpKw. In order to help answering questions about sequence biases around GlutarylLysine sites, a two sample logo, proposed by vacic et al [38], was generated to visualize residues that are significantly enriched or depleted in the set of K-glu fragments. The Two Sample Logo of benchmark dataset, as shown in Figure 3, contains 41 residue fragments, 20 upstream and 20 downstream, from all Lysines found in experimentally verified glutarylated proteins. The positive sample contains 954 fragments around experimentally verified K-glu sites, while the negative sample contains majority of remaining Lysines from the same set of proteins, 2405 in total. Significant variances in the nearby Lysines were found between the glutarylated and non-glutarylated sites. In the depleted position, residues K, L, R and G were more frequently observed while in enriched region P, V and E were observed frequently. Multiple amino acid residues were found stacked at some over- or under-represented positions of the surrounding sequences in samples suggesting significant information between the positive and negative samples. The above results indicate that raw sequences can be used to differentiate between the samples of two classes.

Table 1: Encoding of amino acid used in this study

X	A	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	Y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

2.2. Sample Encoding

DNNs require input sequence data in quantitative form for processing. We applied a basic quantitative encoding of PseAAC sequences, shown in Table 1, to avoid causing bias in dataset due to encoding technique. The quantitative encoding is done in accordance with Table 1 where first row displays the IUPAC symbols of amino acids and corresponding entries in second row show the integer used to represent the amino acid in encoded sample. The resulting benchmark dataset consisted of integer strings which are readily amenable to DNNs. The benchmark dataset was divided into a training set of 1683 samples and a test set of 722 samples and both training and test sets retained the original class ratio.

2.3. Candidate Deep Model Training and Optimization

This section explains training and optimizing of DNN candidate models for predicting K-glu sites. The study conducted experiments using well-known neural network architectures such as Fully Connected Neural Networks (FCNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) with simple units, Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) units respectively. For optimization of DNN candidate models, we adopted the Randomized Hyperparameter search methodology of Bergstra et al. [39]. Randomized Hyperparameter search offers better hyperparameters for DNNs with limited computational budget by performing a random search over large hyperparameter space. This is achieved by randomly sampling the hyperparameters from the space and evaluating the performance of models created using these parameters. For each DNN used to predict the K-glu site, the following subparagraph provide a brief introduction and architecture.

2.3.1. Fully Connected Neural Network

Standard neural networks or Fully connected neural networks (FCNs) are classic architectures of deep neural networks. FCN is said to be completely connected as each neuron in preceding layer is connected to each neuron in the next layer of neural network. The FCN is intended to approximate the

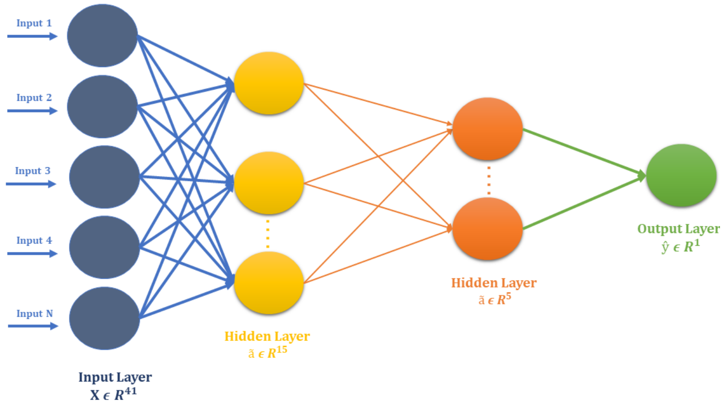


Figure 4: Architecture of Standard Neural Network for K-glu site identification

learning function f^* which can be defined as $y = f^*(\alpha, x)$ and assigns a class label y to input x using appropriate parameters α. The function of FCN is to learn the best set of parameters α such that the mapping $y = f^*(\alpha, x)$ provides the best possible approximation to f^*. The FCN used to analyze glutaryllysine, shown in 4, consists of two fully connected layers consisting of 15 and 5 rectified linear neurons (relu) respectively. The output layer is based on a single Sigmoid neuron to perform binary classification. In order to minimize negative logarithmic loss, the model is optimized using stochastic gradient decent with a learning rate of 0.01. For training the FCN, only the training set was used, which was further divided into trainset and validation set with a 70/30 partition ratio. It is relevant to mention that FCN and other DNNs were never allowed to see the test set to ascertain the generalization capability of resulting K-glu prediction models. Once trained, the predictive model was tested on test set and performance was evaluated using standard performance evaluation metrics.

2.3.2. Recurrent Neural Networks

A limitation of FCN is its inability to share the weights learned by individual neurons which results in failure to identify similar patterns occurring at different positions of sequences [40]. Recurrent neural network (RNN) overcomes this constraint by using a looping mechanism over time steps to address the aforementioned problem [41]. RNN manipulates sequence vectors $x_1, \ldots, x_n$ by employing a recurrence of the form $a_t = f_\alpha(\gamma_{t-1}, x_t)$ where f is an activation function, α is a collection of parameters used at each phase

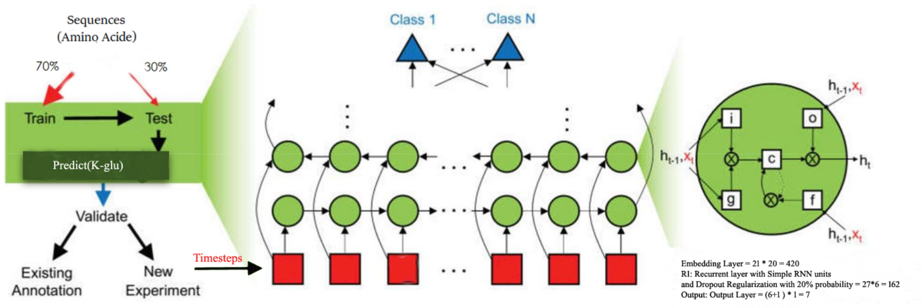

Sequences
(Amino Acide)
70%
30%
Train
Test
Predict(K-glu)
Validate
Existing
Annotation
New
Experiment
Timesteps
Class 1
Class N
h_{t-1}, x_t
i
o
c
g
f
h_t
h_{t-1}, x_t
h_{t-1}, x_t
Embedding Layer = 21 * 20 = 420
Rl: Recurrent layer with Simple RNN units
and Dropout Regularization with 20% probability = 27*6 = 162
Output: Output Layer = (6+1) * 1 = 7

Figure 5: Architecture of RNNs for K-glu site identification.

t and x_t is input at timestep t.

For current study, the candidate RNN based models are developed by three variations of Recurrent Neurons i.e simple RNN unit, a gated recurring unit (GRU), and a Long Short-Term (LSTM) unit. The architecture shared by these three RNNs is shown in figure 5 where the green circles show RNN units used in the network, and the red squares show different timesteps of the sequence being classified. In a simple RNN neuron, the parameters controlling the connections, from the input to the hidden layer, the horizontal connection between the activations, and from the hidden layer to the output layer, are shared. Forward pass in a simple RNN neuron can be formulated by following set of equations:

$$\begin{aligned} a^t &= g(W_a[a^{t-1}, X^t] + b_a) \\ y^t &= f(W_y * a^t + b_y) \end{aligned}$$

Where t denotes the current time step, g represents an activation function, X^t denotes input at timestep t, b_a describes the bias, W_a presents cumulative weights and a^t is activation output at timestep t. This activation a^t can be used to calculate the predictions y_t at time t if desired. Table 2 displays the architecture of the RNN model with SimpleRNN neurons. This model make use of an embedding layer to project the amino acid sequence in vector space $\mathbb{R}^{20}$ and convert the semantic relationships into geometric relationships. These geometric relationships of sequence vectors are interpreted by following layers of DNN model to learn deep feature representations which in turn are appraised by output layer, consisting of a single sigmoid unit, to make predictions.

In many applications, DNNs with simple RNN neurons show promising re-

Table 2: Architecture of RNN based on SimpleRNN neurons for lysine glutarylation site predictions

Layer Type	No. of Weights
Embedding Layer	21 * 20 = 420
Recurrent layer with Simple RNN units and Dropout Regularization with 20% probability	27*6 = 162
Output: Output Layer	(6+1) * 1 = 7

sults, but these neurons are prone to vanishing gradients and show limited ability to learn long-term dependencies. To rectify this limitation of simple RNN neurons, research community has proffered many updated recurrent neuron architectures, including GRU proposed by Cho et al. [42] and LSTM proposed by Hochreiter et al. [43] to counter the problem of vanishing gradients and enable learning of long-term dependencies.
Gated Recurrent Unit (GRU), proposed by Cho et al. 2014 [42], is better at learning long term relationships in sequence data. At each stage t, the GRU unit uses the memory variable $H^t = a^t$ which contains an updated description of all samples passed through the unit. Since the GRU unit superimpose the value of H^t with the candidate value $\bar{H}^t$ at each step t, this overwriting is regulated by the update gate Γ_u. GRU neuron function can be formulated in following set of equations:

$$\begin{aligned}
\bar{H}^t &= tanh(W_c[\Gamma_r * H^t, X^t] + b_c \\
\Gamma_r &= \sigma(W_r[H^{t-1}, X^t] + b_r) \\
\Gamma_u &= \sigma(W_u[H^{t-1}, X^t] + b_u) \\
H^t &= \Gamma_u * \bar{H}^t + (1 - \Gamma_u) * H^{t-1} \\
a^t &= H^t
\end{aligned}$$

In the above set of equations, W_r, W_c and W_u denote respective weights, the corresponding bias terms are illustrated by b_r, b_c and b_u, X^t represents the input at timestep t, $\sigma$is the logistic regression function and a^t represents activations at time step t. The architecture of the implemented RNN model built with GRU is the same as that of simple RNN except the fact that GRU neurons are used in recurrent layer. For glutaryllysine site prediction, Table 3 displays the architecture of GRU based RNN model. Hochreiter et al. [43]

Table 3: Architecture of RNN based on GRU neurons for lysine glutarylation site predictions

Layer Type	No. of Weights
Embedding Layer to convert numeric sequence into vector sequence	21 * 20 = 420
Recurrent layer with GRU units and Dropout Regularization with 20% probability	81*6 = 486
Output: Output Layer	(6+1) * 1 = 7

Table 4: Architecture of RNN based on LSTM neurons for lysine glutarylation site predictions

Layer Type	No. of Weights
Embedding Layer	21 * 20 = 420
Recurrent layer with LSTM units and Dropout Regularization with 20% probability	116*6 = 696
Output: Output Layer	(8+1) * 1 = 9

presented LSTM with some modifications in RNN unit architecture, which is a more effective generalization of GRU. Notable differences in GRU and LSTM cells are outlined below:

1. For $\bar{H}^t$ computation, generic LSTM units do not use relevance gate Γ_r.
2. Instead of Update gate Γ_u, LSTM units use two different gates including Output gate Γ_o and Forget gate Γ_f. Output gate monitors the exposure of the memory cell contents H^t to compute activation outputs of LSTM unit for other hidden units in the network. Forget gate manages the amount of overwrite on H^{t-1} to achieve H^t i.e. how much memory cell content needs to be forgotten for memory cell.
3. In LSTM, the contents of the memory cell may not be equal to the activation a^t which is contrary to GRU architecture.

Except for LSTM units in recurrent layer, the RNN model built with the LSTM neurons has the same architecture as that of simple RNN and GRU models. The architecture of the glutaryllysine prediction model developed using LSTM neurons based RNN is shown in Table 4.

Table 5: Architecture of CNN for lysine glutarylation site predictions

Layer Type	No. of Weights
Embedding Layer	21 * 20 = 420
Convolution 1D with 10 filters of size 3	((3 * 20) + 1)* 10 = 610
Maxpooling 1D	Not Applicable
Dropout with 25% of probability	Not Applicable
Convolution 1D with 10 filters of size 3	((3 * 10) + 1)* 10)= 310
Maxpooling 1D	Not Applicable
GlobalAveragePooling1D	Not Applicable
Output: Output layer with one sigmoid	(10+1)* 1 = 11

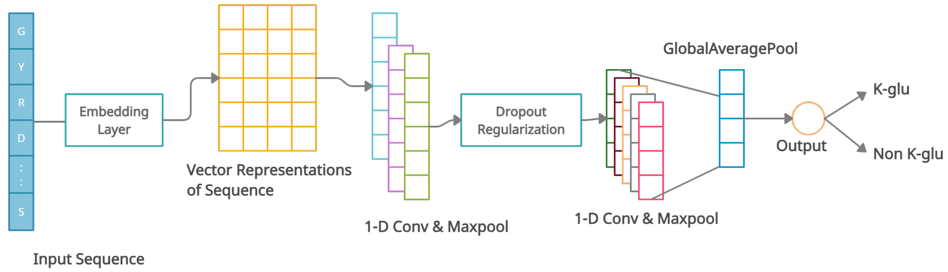


Figure 6: Architecture of CNN for K-glu site identification.

2.3.3. Convolutional Neural Network

The CNN is a neural network structure primarily designed for the analysis of data with complex spatial relationships like images or videos [44]. CNN tries to learn a filter that can transform input data to the right output prediction. In addition to its capacity for handling large amounts of data, CNN can build local connections to learn feature maps, share training parameters among connections and reduce dimensions using the subsampling operations. These characteristics help CNN to understand the spatial features of inputs despite their locality in the input data, a property known as location invariance. The architecture of the K-glu identification model based on CNN is shown in figure 6. The suggested CNN-based model was developed with an embedding layer, two convolution-maxpool blocks, a global averaging layer and an output layer of sigmoid neuron. Every sample of the peptide x with a length of $\mu = 41$ is encoded by the embedding layer in the form of tensor $X \in R^{\rho * \mu}$ where $\rho \in R^{20}$ is the representation vector of each amino acid residue in R^{20}. Two convolution layers and a sub-sample layer are combined to form each convolution block. The convolution layer of both the blocks consisted of 10 1-D convolution units. Every n-dimensional output sample is flattened into a 1-D array of 10 scalars which in turn is used by the output layer to predict the labels. A single sigmoid unit that performs binary classification is employed in the output layer to make predictions. The architectural components of CNN are elaborated in Table 5.

3. Results

This section explains the performance results of multiple DNN based predictors developed in this research to predict K-glu site location. Notable evaluation metrics used in this study include receiver operating characteristics curve (ROC) curve, precision-recall curve and point metrics, including mean average precision (mAP), accuracy, F1 measurements, and matthew's correlation coefficient (MCC) to find the best DNN-based K-glu prediction model. All models were evaluated on test data which was not used during the predictor training phase. This was done to ensure the fairness of results and to evaluate the generalization capability of predictors being evaluated. An overview of the model evaluation parameters used in this study is given in the following subsection which illuminates adequate discussion of the results of the evaluation. In order to ensure fairness, all evaluation results come from independent test samples which were not used in training phase of DNN based models.

3.1. Precision-Recall Curve and Mean Average Precision

For the evaluation of prediction models, precision and recall are essential indicators. Precision measures the relevance of the positive outcomes predicted by the model while recall measures the sensitivity of the model for positive samples. A high precision and recall rating implies that returned positive class predictions contain high ratio of true positives (high Precision) while predicting the majority of positive class samples in the data set are (High Recall). Precision-Recall curve is achieved by plotting both of these metrics against each other. Precision-Recall plots can provide the viewer with an accurate prediction of future classification performance because they evaluate the fraction of true positives among positive predictions [45]. In precision-recall space, the closer a predictor's score is to the perfect classification point (1,1), the better the predictor is and vice versa. The

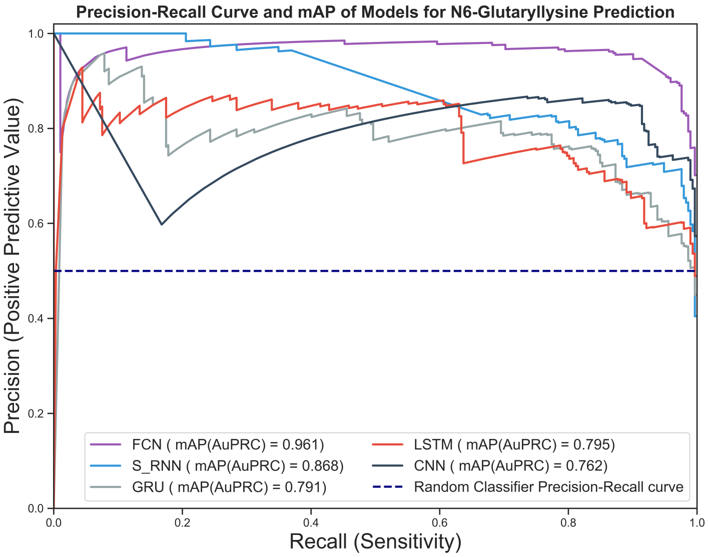


Figure 7: DNN-based K-glu Prediction Models' Precision-Recall Curves and mAP scores

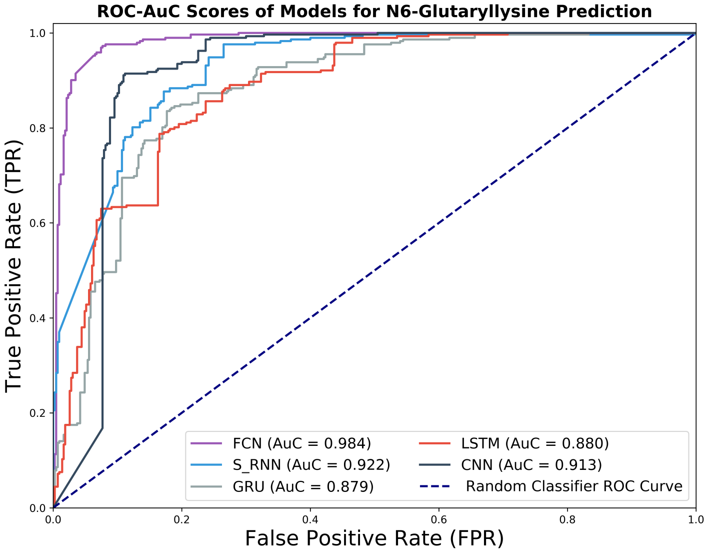


Figure 8: ROC Curve and AUC Scores for DNN based K-glu Prediction Models

precision-recall curve of the candidate DNN based predictors is shown in figure 7. As can be seen in figure 7, since FCN model's curve is closest to the perfect prediction point (1,1) in the precision-recall space as compared to the scores of other predictors, this shows the better performance of FCN-based predictive model. All the other DNN based predictors performed poorly as compared to FCN as depicted by their respective curves with CNN model showing the least performance. In the legend portion of figure 7, the mean average precision (mAP) scores for DNN models are shown. mAP is defined as the region under the precision-recall curve. The greater the classifier mAP score is, the better the prediction performance of classifier is and vice versa. In the legend section of figure 7, map scores are shown for all candidate K-glu prediction models. The FCN based model showed best score of 0.961 followed by the S_RNN with value of 0.868. The least scores were shown by CNN-based model which scored 0.762 for K-glu PTM site predictions. All DNN based predictors achieved more than 70% scores.

3.2. Receiver Operating Characteristics Curve (RoC) and Area under RoC

The ROC Curve is a plot of False Positive Rate and True Positive Rate also known as recall. In a sense, the ROC curve illuminates the cost benefit analysis of the classifier under evaluation [46]. False Postive rate is defined as the ratio of false positive (FP) to total negative samples and measures the fraction of negative examples that are misclassified as positive. This is considered as the cost because any further action taken on FP result, considering it a positive prediction, is wasted. True positive rate, which is measured as the fraction of positive examples that are correctly predicted, can be considered as the benefit because the correct positive predictions done by classifier help solve the problem being investigated. In ROC space, the False Positive Rate (FPR) is plotted on the x-axis and the True Positive Rate (TPR) on the y-axis. Point (0,1) is a perfect ranking with a false positive rating of 0 and a true positive rating of 1, respectively. The closer the curve to the point (0,1) for the classifiers, the better the output of the corresponding classifier. Figure 8 shows the ROC curve of the glutaryllysine predictor models. The results of the ROC curve corroborate the assessments demonstrated by precision-recall curve. Here too the FCN model results overshadow the other models' results. The results were marginally lower in other DNN models based on CNN, LSTM, RNN and GRU.

Sometimes it is desirable to summarize the ROC curve insights of a model

Table 6: Comparison of AuC, De-long's p-value, 95% Confidence Interval of AuC and Accuracy of proposed Models.

Proposed Model	**Area Under Curve**	**95%CI**	**p-Value**	**Accuracy%**
FCN Model	0.984	[0.975-0.991]	0	94.3
SRNN Model	0.922	[0.903-0.940]	0	84.1
GRU Model	0.879	[0.853-0.903]	$1.3e-193$	82.7
LSTM Model	0.880	[0.856-0.904]	$2.43e-207$	80.4
CNN Model	0.913	[0.889-0.936]	$4.79e-256$	90.1

to a single scalar value that shows the performance of the model. The area under an ROC curve, called the AuC, is one of these popular methods. Not only does AuC minimizes the ROC curve outcomes to a single value, it also illuminates statistical insights of the model's performance. AuC is equivalent to the probability that the classifier will rank a randomly chosen positive sample higher than a randomly chosen negative instance. Additionally AuC is also equivalent to the Wilcoxon test of ranks [46]. Legend section of figure 8 shows the AuC values for the models developed in this study. The FCN-based prediction model shows the highest AuC value of 0.98 while The model built with GRU obtained the least rating of 0.88. The scores achieved by remaining three models were distributed between these two score values for other three DNN-based models.

The comparison of the overall diagnostic accuracy of two models is frequently addressed by comparing the resulting paired AuCs using Delong's method [47] of non-parametric comparison of two or more RoC curves. We used the fast implementation of Delong's method by Sun et al. [48] to calculate the p-values by comparing each AuC with random classifier. We also constructed the 95% Confidence interval using AuC for DNN based predictors developed in this study. AuC scores, Delong p-value scores and 95% confidence Intervals of AuC are shown in Table 6.

3.3. Accuracy and F1-Measure

For models trained using balanced datasets, accuracy is the most common model evaluation metric. It demonstrates the fraction of correctly predicted samples to total number of samples for model under evaluation. The accuracy scores for the glutaryllysine prediction models, calculated over independent test set, are shown in figure 9. As can be seen from the figure 9, the FCN and CNN based deep models showed an accuracy value above 90 percent and least accuracy value of 84 percent is demonstrated by LSTM based RNN model.

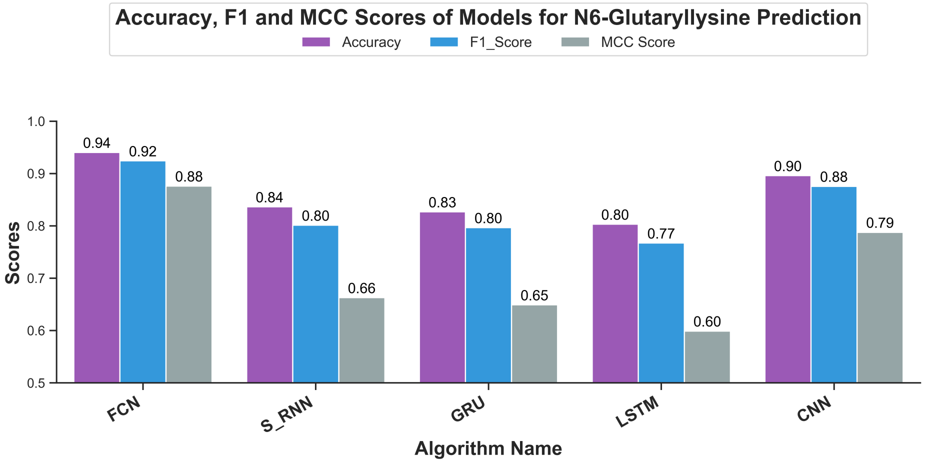


Figure 9: Accuracy, F1-Measure and MCC scores of DNN based K-glu prediction models

In situations, where an optimal combination of precision and recall is required in the form of single scalar value, F1-measure is also a popular alternative. F1 measure is calculated as a harmonic mean of precision and recall scores of a model. Figure 9 indicates the predictive glutaryllysine models' F1 values which validates the earlier evaluation demonstrated by AuC and mAP scores. The FCN model achieved optimal score of 98.9 percent and the second position was attained by the CNN model with F1 score of 92 percent. The model based on LSTM based RNN gave a weak rating of 77 percent.

An other metric, matthews correlation coefficient (MCC), is considered an effective solution overcoming the class imbalance issues prevalent in accuracy and other binary classification model evaluators. Originally developed by Matthews in 1975 for comparison of chemical structures [49], MCC was brought into limelight again by Baldi and colleagues [50] in 2000 as a standard performance metric for binary classification models with a natural extension to the multiclass case. According to Chicco et al., MCC is a more accurate statistical metric that generates a high score only if good results were obtained in the prediction of all four groups of the confusion matrix (true positives, false negatives, true negatives, and false positives), in proportion to both the size of positive elements and the size of negative elements in the dataset [51]. Figure 9 shows the results of MCC for all DNN based models developed in this study. The reader can confirm that the FCN based model

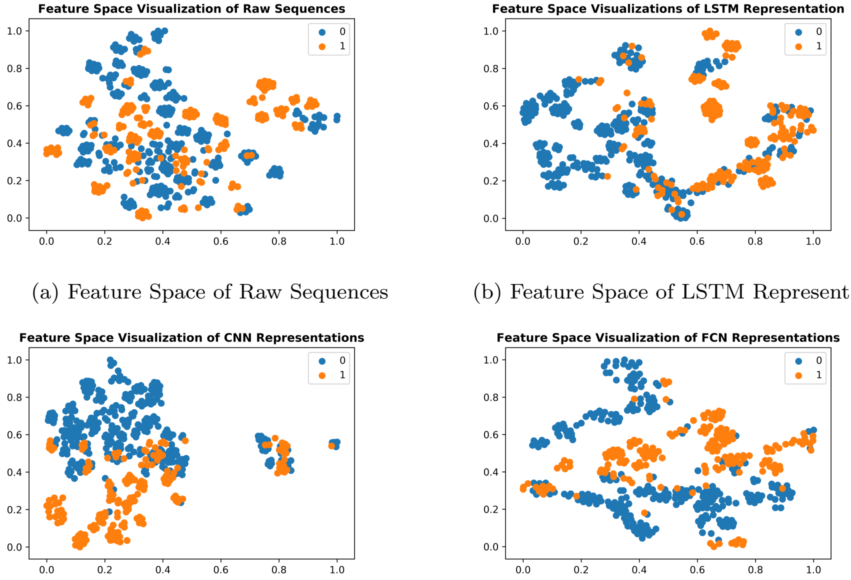
Table 7: Comparison of Proposed k-glu predictor with literature

Predictor	Sensitivity	Specificity	Acc(%)	MCC
Proposed FCN Predictor	0.95	0.91	94.3	0.88
Ju and He [52]	0.52	0.78	75.4	0.22
Xu et al. [21]	0.89	0.97	96.6	0.84
Huang et al. [20]	0.677	Not Reported	63.8	0.28
Duo et al. [53]	0.73	0.72	Not Reported	Not Reported
Ju and Wang [54]	0.59	0.79	76.6	0.27

showed best performance with an MCC value of 0.88 and CNN is not far behind with an MCC value of 0.79. As depicted by previous evaluation metrics, LSTM based predictor turned out to be the least suitable model for K-glu prediction with MCC score of 0.60. These result show that FCN-based prediction model is the finest of all DNN based models, developed in this study for glutaryllysine prediction.

4. Comparative Analysis and Discussion

This section provides the comparison of proposed FCN based predictor with notable contributions from literature. The comparison results are shown in Table 7. The first K-glu predictor GlutPred was proposed by Ju and He [52] which made use of maximum relevance minimum redundancy (mRMR) feature selection algorithm. The performance scores achieved by GlutPred for Sensitivity, Specificity, Accuracy and MCC are 0.52, 0.78, 75.4,0.22 respectively. Xu et al. [21] used position-specific propensity matrix (PSPM) features developed their predictor, iGlu-Lys, using SVM algorithm. iGlu-Lys showed promising results and achieved 0.89, 0.97, 96.6, 0.84 for sensitivity, specificity, accuracy and MCC respectively. Huang et al. [20] proposed a prediction model by incorporating maximal dependence decomposition (MDD)-identified substrate motifs into an integrated SVM classifier. Sensitivity, Accuracy and MCC scores of predictor, proposed by Huang et al., reached up to 0.677,63.8 and 0.28 respectively while specificity score was not reported. Another notable work in lysine glutarylation prediction belongs to Duo et al. [53] who proposed an adaboost based model, iGlu_AdaBoost, developed from features selected using Chi2 following incremental feature selection algorithm



(a) Feature Space of Raw Sequences

(b) Feature Space of LSTM Representation

(c) Feature Space of CNN Representation

(d) Feature Space of FCN Representation

Figure 10: Feature Space Visualizations of deep representations for positive and negative K-glu samples

(IFS). iGlu_AdaBoost showed sensitivity and specificity 0.73 and 0.72 on independent dataset while accuracy and MCC scores were not reported for the same. Ju and wang [54] used un-verified positive k-glu sites with SVM algorithm to devise PUL-GLU. Evaluation scores of PUL-GLU on independent data were 0.59, 0.79, 76.6 and 0.27 for Sensitivity, Specificity, Accuracy and MCC.

Although aforementioned contributions performed notable work for prediction of lysine glutarylation but all of them rely on the quality of the features extracted or selected by the feature extraction algorithms. Models proposed in this study are different because they accept raw PSeAAC sequences as input and do not rely on the quality of features to perform better predictions. Additionally, the results shown by our optimal model are comparable to the iGlu-Lys, which has shown the optimal scores as compared to other conventional feature based predictors. Although the specificity and accuracy score of iGlu-Lys [21] is highest, the value of sensitivity and MCC is much lower than that of proposed FCN model. The higher specificity shows that iGlu-Lys ranks a query lysine site as non k-glu with higher probability than that of a positive k-glu site which is not the case with FCN based predictor as shown by higher sensitivity score of the same. In addition, it has been shown [51] that MCC is a better performance evaluator than accuracy in non-balanced class problems. The higher score of MCC for FCN model shows the higher performance of proposed approach for lysine glutarylation site prediction.

To understand the feature representations learned by the DNN's nonlinear transformation, visualization of feature space serves as an important tool. For creating visualizations in this study, we computed the output for test set sequences from the penultimate layer of each trained DNN based model and projected this output to 2D space using T-SNE, proposed by Maaten and Hinton [55]. T-SNE uses non-linear statistical approach to project data from higher dimensions to lower dimensions which can then be easily plotted. In this study, the 2D projection of each deep representation was plotted based on class labels to understand the distribution of sequences belonging to each class. For plotting the visualizations, matplotlib and seaborn package of python were used. Visualizations are shown in figures 10a, 10b, 10c and figure 10d.
Feature space visualization for raw PseAAC sequences is shown in figure 10a.

As visible in the figure, positive and negative sequences are jumbled up and no clear separation is available which means any classifier using this representation will have a hard time separating the sequences from both classes to perform predictions. Figures 10b, 10c and 10d illustrate the effect of non-linear transformations of three DNNs used in this study to separate both classes in respective feature space for achieving better predictions. As illuminated in figure 10b, which shows the feature representation learned by LSTM model, the reader can see that this model was not successful enough to separate both classes in the learned representation before passing the features to output layer and this resulted in relatively poor evaluation scores of LSTM-RNN based predictor. Figure 10c sheds light on representation learned by CNN based model. It can be seen that this model is relatively successful, as compared to LSTM-RNN based model, to learn a representation which separates the samples belonging to both classes although small overlap of samples of different classes is seen in some regions. The most successful feature representation is achieved by FCN based model, as illustrated in figure10d. The data distribution of positive and negative samples in FCN representation is shown as violin plot in Figure 12. As can be seen in aforementioned figures, the FCN model was able to learn the representation in which the positive and negative samples are sufficiently separated from each other enabling better K-glu site identification by output layer. The relationship of deep features, in FCN representation, is further illustrated by the heatmap to identify correlation between individual deep featuresin Figure 11. As can be seen, almost all deep features are sufficiently independent of each other which means each deep feature adds value to enable better classification of K-glu sites. Additionally, the reader can verify from the figure 10d that FCN representation is most successful in separating the samples belonging to different classes with minimal overlap. Although negative samples are distributed in two different clusters but the overlap between positive and negative samples in minimal. This means any classifier consuming this representation will be able to separate both classes with less effort and achieve better predictions. This fact is also corroborated by the evaluation results discussed in earlier sections. An additional benefit of our approach is automatic learning of feature representation using stochastic gradient descent. This approach removes the need to use expensive feature engineering process. In addition, the proposed DNN based predictors developed in this contribution demonstrates only the initial step towards employing deep learning for lysine glutarylation PTM site prediction and further study will draw on the research described in this study

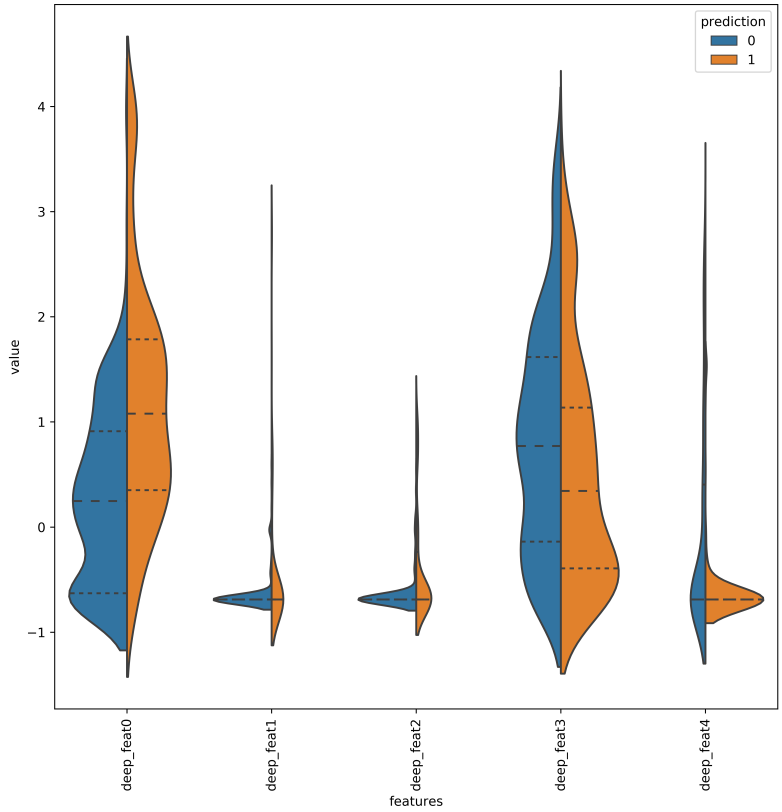


Figure 11: Violin plot showing the data distribution of deep features learned by FCN Model

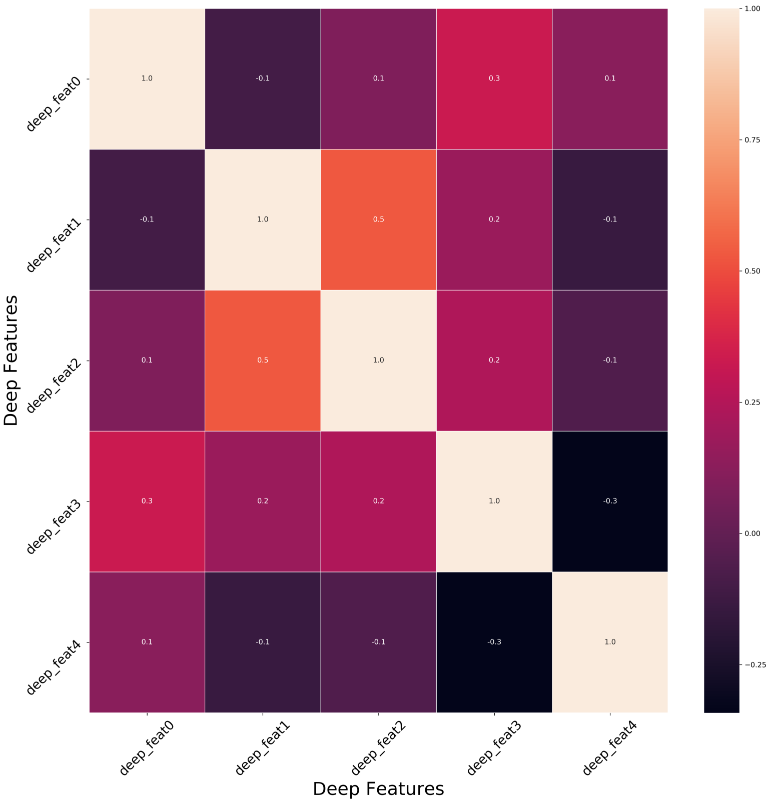


Figure 12: Relationship of Deep Features learned by FCN

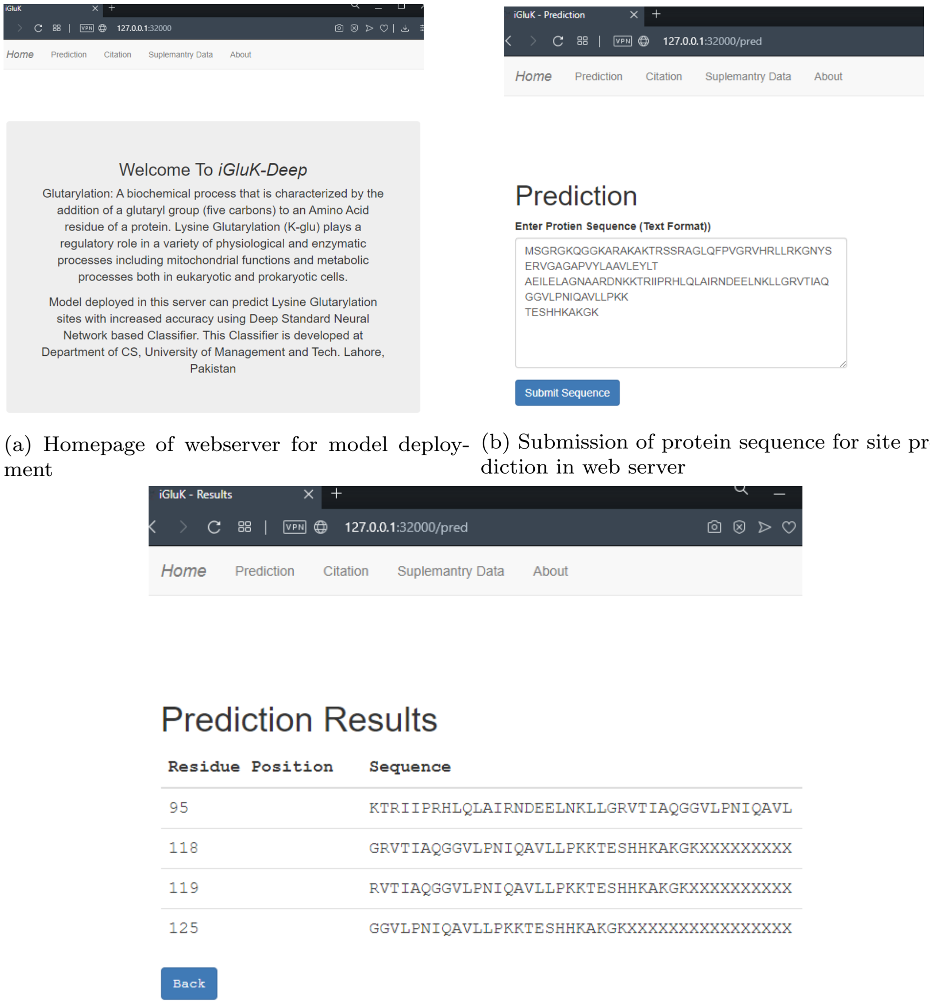
to devise better DNN predictors for the same.

5. Model Deployment as Web Service

The final phase for Chou's 5-step rule is the deployment of the predictor in the form of a web application for making it available to the general research community. To achieve this goal, a web application was developed in this study utilizing the best performing FCN based iGluK-Deep model as shown in figure 13. In the name iGluK-Deep, the iGluK portion represents identification of glutarylation and K represents IUPAC symbol of lysine while the word 'Deep' connects the model's roots to the DNNs used to develop this model. The web application can accept peptide samples in the form of strings and return the predicted lysine sites which are highly likely to be glutarylated. Homepage of iGluK-Deep webserver is shown in figure 13a while figure 13b highlights the peptide sequence submission process for computing K-glu sites. Figure 13c calls attention towards the result page showing the predicted lysine sites likely to be glutarylated and the corresponding μ length sequence of residues. The model is temporarily deployed at http://18.224.94.143/. We believe that our humble effort to improve the predictability of lysine glutarylation will be of service to research community.

6. Conclusions

In this study, we proposed an improved, effective, and less cumbersome approach, based on deep neural networks, for identification of lysine glutarylation sites in proteins. The proposed approach makes use of Chou's Pseudo Amino Acid Composition with deep neural networks to identify lysine glutarylation PTM sites. We employed well-known DNNs including Standard neural network (FCN), three variants of recurrent neural networks (RNNs) and convolutional neural network (CNN) for both the tasks of learning a feature representation of peptide sequences and performing the classifications. From all DNN based predictors in this study, the FCN based predictor reached up to the highest performance scores evaluated using well-known model evaluation metrics. The model achieved 94.3% accuracy and 0.88 Matthew correlation coefficient (MCC) score for independent test set. The comparisons of proposed FCN based predictor with notable research contributions were performed which shows the efficacy of proposed predictor. Based on aforementioned evaluation and comparison results, it is concluded



Welcome To *iGluK-Deep*

Glutarylation: A biochemical process that is characterized by the addition of a glutaryl group (five carbons) to an Amino Acid residue of a protein. Lysine Glutarylation (K-glu) plays a regulatory role in a variety of physiological and enzymatic processes including mitochondrial functions and metabolic processes both in eukaryotic and prokaryotic cells.

Model deployed in this server can predict Lysine Glutarylation sites with increased accuracy using Deep Standard Neural Network based Classifier. This Classifier is developed at Department of CS, University of Management and Tech. Lahore, Pakistan

Prediction

Enter Protien Sequence (Text Format))

```
MSGRGKQGGKARAKAKTRSSRAGLQFPVGRVHRLLRKGNYS
ERVGAGAPVYLAAVLEYLT
AEILELAGNAARDNKKTRIIPRHLQLAIRNDEELNKLLGRVTIAQ
GGVLPNIQAVLLPKK
TESHHKAKGK
```

(a) Homepage of webserver for model deployment

(b) Submission of protein sequence for site prediction in web server

Prediction Results

Residue Position	Sequence
95	KTRIIPRHLQLAIRNDEELNKLLGRVTIAQGGVLPNIQAVL
118	GRVTIAQGGVLPNIQAVLLPKKTESHHKAKGKXXXXXXXXX
119	RVTIAQGGVLPNIQAVLLPKKTESHHKAKGKXXXXXXXXXX
125	GGVLPNIQAVLLPKKTESHHKAKGKXXXXXXXXXXXXXXXX

(c) Prediction results with site position in given protein sequence

Figure 13: Webserver for identification of Lysine Glutarylation

that the proposed predictor will help the research community to efficiently and accurately identify lysine glutarylation and enable better understanding of this protein modification.

References

[1] E Furuya and K Uyeda. Regulation of phosphofructokinase by a new mechanism. An activation factor binding to phosphorylated enzyme. *Journal of Biological Chemistry*, 255(24):11656–11659, 1980. Publisher: American Society for Biochemistry and Molecular Biology.

[2] Sheraz Naseer, Waqar Hussain, Yaser Daanial Khan, and Nouman Rasool. Sequence-based Identification of Arginine Amidation Sites in Proteins Using Deep Representations of Proteins and PseAAC. *Current Bioinformatics*, 15(8):937–948, January 2021.

[3] Sheraz Naseer, Waqar Hussain, Yaser Daanial Khan, and Nouman Rasool. iphoss (deep)-pseaac: Identify phosphoserine sites in proteins using deep learning on general pseudo amino acid compositions via modified 5-steps rule. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.

[4] Yaser Daanial Khan, Nouman Rasool, Waqar Hussain, Sher Afzal Khan, and Kuo-Chen Chou. iPhosY-PseAAC: Identify phosphotyrosine sites by incorporating sequence statistical moments into PseAAC. *Molecular Biology Reports*, pages 1–9, 2018.

[5] Muhammad Awais, Waqar Hussain, Yaser Daanial Khan, Nouman Rasool, Sher Afzal Khan, and Kuo-Chen Chou. iPhosH-PseAAC: Identify phosphohistidine sites in proteins by blending statistical moments and position relative features according to the Chou's 5-step rule and general pseudo amino acid composition. *IEEE/ACM transactions on computational biology and bioinformatics*, 2019. Publisher: IEEE.

[6] Minjia Tan, Chao Peng, Kristin A. Anderson, Peter Chhoy, Zhongyu Xie, Lunzhi Dai, Jeongsoon Park, Yue Chen, He Huang, Yi Zhang, Jennifer Ro, Gregory R. Wagner, Michelle F. Green, Andreas S. Madsen, Jessica Schmiesing, Brett S. Peterson, Guofeng Xu, Olga R. Ilkayeva, Michael J. Muehlbauer, Thomas Braulke, Chris Mühlhausen, Donald S. Backos, Christian A. Olsen, Peter J. McGuire, Scott D. Pletcher,

David B. Lombard, Matthew D. Hirschey, and Yingming Zhao. Lysine Glutarylation Is a Protein Posttranslational Modification Regulated by SIRT5. *Cell Metabolism*, 19(4):605–617, April 2014.

[7] Matthew D Hirschey and Yingming Zhao. Metabolic regulation by lysine malonylation, succinylation, and glutarylation. *Molecular & Cellular Proteomics*, 14(9):2308–2315, 2015. Publisher: ASBMB.

[8] Yi-min Cheng, Xiao-nian Hu, Zhen Peng, Ting-ting Pan, Fang Wang, Hou-yang Chen, Wen-qiong Chen, Yu Zhang, Xu-hui Zeng, and Tao Luo. Lysine glutarylation in human sperm is associated with progressive motility. *Human Reproduction*, 34(7):1186–1194, July 2019.

[9] Boda Zhou, Yipeng Du, Yajun Xue, Guobin Miao, Taotao Wei, and Ping Zhang. Identification of Malonylation, Succinylation, and Glutarylation in Serum Proteins of Acute Myocardial Infarction Patients. *PROTEOMICS–Clinical Applications*, 14(1):1900103, 2020. Publisher: Wiley Online Library.

[10] Xiucong Bao, Zheng Liu, Wei Zhang, Kornelia Gladysz, Yi Man Eva Fung, Gaofei Tian, Ying Xiong, Jason Wing Hon Wong, Karen Wing Yee Yuen, and Xiang David Li. Glutarylation of histone H4 lysine 91 regulates chromatin dynamics. *Molecular Cell*, 76(4):660–675, 2019. Publisher: Elsevier.

[11] Sheraz Naseer, Rao Faizan Ali, Amgad Muneer, and Suliman Mohamed Fati. iAmideV-Deep: Valine Amidation Site Prediction in Proteins Using Deep Learning and Pseudo Amino Acid Compositions. *Symmetry*, 13(4), 2021.

[12] Waqar Hussain, Yaser Daanial Khan, Nouman Rasool, Sher Afzal Khan, and Kuo-Chen Chou. SPalmitoylC-PseAAC: A sequence-based model developed via Chou's 5-steps rule and general PseAAC for identifying S-palmitoylation sites in proteins. *Analytical biochemistry*, 568:14–23, 2019.

[13] Waqar Hussain, Yaser Daanial Khan, Nouman Rasool, Sher Afzal Khan, and Kuo-Chen Chou. SPrenylC-PseAAC: A sequence-based model developed via Chou's 5-steps rule and general PseAAC for identifying S-prenylation sites in proteins. *Journal of Theoretical Biology*, 468:1–11, May 2019.

[14] Sheraz Naseer, Waqar Hussain, Yaser Daanial Khan, and Nouman Rasool. NPalmitoylDeep-PseAAC: A Predictor for N-Palmitoylation sites in Proteins using Deep Representations of Proteins and PseAAC via modified 5-steps rule. *Current Bioinformatics*, 15, June 2020.

[15] Waqar Hussain, Iqra Qaddir, Sajid Mahmood, and Nouman Rasool. In silico targeting of non-structural 4B protein from dengue virus 4 with spiropyrazolopyridone: study of molecular dynamics simulation, ADMET and virtual screening. *VirusDisease*, pages 1–10, 2018.

[16] Yaser Daanial Khan, Mehreen Jamil, Waqar Hussain, Nouman Rasool, Sher Afzal Khan, and Kuo-Chen Chou. pSSbond-PseAAC: Prediction of disulfide bonding sites by integration of PseAAC and statistical moments. *Journal of theoretical biology*, 2018.

[17] Fuyi Li, Yanan Wang, Chen Li, Tatiana T. Marquez-Lago, André Leier, Neil D. Rawlings, Gholamreza Haffari, Jerico Revote, Tatsuya Akutsu, and Kuo-Chen Chou. Twenty years of bioinformatics research for protease-specific substrate and cleavage site prediction: a comprehensive revisit and benchmarking of existing methods. *Briefings in bioinformatics*, 2018.

[18] Nouman Rasool, Saima Iftikhar, Anam Amir, and Waqar Hussain. Structural and quantum mechanical computations to elucidate the altered binding mechanism of metal and drug with pyrazinamidase from Mycobacterium tuberculosis due to mutagenicity. *Journal of Molecular Graphics and Modelling*, 80:126–131, 2018.

[19] Jiangning Song, Yanan Wang, Fuyi Li, Tatsuya Akutsu, Neil D. Rawlings, Geoffrey I. Webb, and Kuo-Chen Chou. iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. *Briefings in bioinformatics*, 2018.

[20] Kai-Yao Huang, Hui-Ju Kao, Justin Bo-Kai Hsu, Shun-Long Weng, and Tzong-Yi Lee. Characterization and identification of lysine glutarylation based on intrinsic interdependence between positions in the substrate sites. *BMC bioinformatics*, 19(13):13–25, 2019. Publisher: BioMed Central.

[21] Yan Xu, Yingxi Yang, Jun Ding, and Chunhui Li. iGlu-Lys: A Predictor for Lysine Glutarylation Through Amino Acid Pair Order Features. *IEEE Transactions on NanoBioscience*, 17(4):394–401, October 2018.

[22] Md Arafat, Md Ahmad, SM Shovan, Abdollah Dehzangi, Shubhashis Roy Dipta, Md Hasan, Al Mehedi, Ghazaleh Taherzadeh, Swakkhar Shatabda, Alok Sharma, and others. Accurately Predicting Glutarylation Sites Using Sequential Bi-Peptide-Based Evolutionary Features. *Genes*, 11(9):1023, 2020. Publisher: Multidisciplinary Digital Publishing Institute.

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436, May 2015.

[24] Sheraz Naseer, Rao Faizan Ali, P.D.D Dominic, and Yasir Saleem. Learning representations of network traffic using deep neural networks for network anomaly detection: A perspective towards oil and gas it infrastructures. *Symmetry*, 12(11), 2020.

[25] Sheraz Naseer and Yasir Saleem. Enhanced Network Intrusion Detection using Deep Convolutional Neural Networks. *KSII Transactions on Internet and Information Systems*, 12(10), October 2018.

[26] Sheraz Naseer, Waqar Hussain, Yaser Daanial Khan, and Nouman Rasool. Optimization of serine phosphorylation prediction in proteins by comparing human engineered features and deep representations. *Analytical Biochemistry*, 615:114069, February 2021.

[27] Dan Zhang, Zhao-Chun Xu, Wei Su, Yu-He Yang, Hao Lv, Hui Yang, and Hao Lin. iCarPS: a computational tool for identifying protein carbonylation sites by novel encoded features. *Bioinformatics*, None:None, 2020.

[28] Shi-Hao Li, Jun Zhang, Ya-Wei Zhao, Fu-Ying Dao, Hui Ding, Wei Chen, and Hua Tang. iPhoPred: a predictor for identifying phosphorylation sites in human protein. *IEEE Access*, 7:177517 – 177528, 2020.

[29] Wang-Ren Qiu, Bi-Qian Sun, Hua Tang, Jian Huang, and Hao Lin. Identify and analysis crotonylation sites in histone by using support vector machines. *Artificial Intelligence In Medicine*, 83:75–81, 2017.

[30] Ya-Wei Zhao, Hong-Yan Lai, Hua Tang, Wei Chen, and Hao Lin. Prediction of phosphothreonine sites in human proteins by fusing different features. *Scientific Reports*, 6:34817, 2016.

[31] Kuo-Chen Chou. Some remarks on protein attribute prediction and pseudo amino acid composition. *Journal of theoretical biology*, 273(1):236–247, 2011. Publisher: Elsevier.

[32] Kuo-Chen Chou. Using subsite coupling to predict signal peptides. *Protein Engineering*, 14(2):75–79, 2001. Publisher: Oxford University Press.

[33] Sheraz Naseer, Waqar Hussain, Yaser Daanial Khan, and Nouman Rasool. iPhosS(Deep)-PseAAC: Identify Phosphoserine Sites in Proteins using Deep Learning on General Pseudo Amino Acid Compositions via Modified 5-Steps Rule. *IEEE-ACM transactions on computational biology and bioinformatics*, PP, November 2020.

[34] Sheraz Naseer, Rao Faizan Ali, Suliman Mohamed Fati, and Amgad Muneer. iNitroY-Deep: Computational Identification of Nitrotyrosine Sites to Supplement Carcinogenesis Studies Using Deep Learning. *IEEE Access*, 9:73624–73640, 2021.

[35] Hao Lv, Fu-Ying Dao, Zheng-Xing Guan, Hui Yang, Yan-Wen Li, and Hao Lin. Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method. *Briefings in Bioinformatics*, None:None, 2020.

[36] Lidong Wang, Ruijun Zhang, and Yashuang Mu. Fu-SulfPred: Identification of Protein S-sulfenylation Sites by Fusing Forests via Chou's General PseAAC. *Journal of theoretical biology*, 461:51–58, 2019.

[37] The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, January 2019.

[38] Vladimir Vacic, Lilia M Iakoucheva, and Predrag Radivojac. Two Sample Logo: a graphical representation of the differences between two sets of sequence alignments. *Bioinformatics*, 22(12):1536–1537, 2006. Publisher: Oxford University Press.

[39] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *JMLR*, page 305, 2012.

[40] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[41] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

[42] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[44] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[45] Takaya Saito and Marc Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3):e0118432, March 2015.

[46] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[47] Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pages 837–845, 1988.

[48] Xu Sun and Weichao Xu. Fast implementation of delong's algorithm for comparing the areas under correlated receiver operating characteristic curves. *IEEE Signal Processing Letters*, 21(11):1389–1393, 2014.

[49] Brian W Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[50] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[51] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):6, 2020.

[52] Zhe Ju and Jian-Jun He. Prediction of lysine glutarylation sites by maximum relevance minimum redundancy feature selection. *Analytical Biochemistry*, 550:1–7, June 2018.

[53] Lijun Dou, Xiaoling Li, Lichao Zhang, Huaikun Xiang, and Lei Xu. iGlu_adaboost: Identification of Lysine Glutarylation Using the AdaBoost Classifier. *Journal of Proteome Research*, 2020.

[54] Zhe Ju and Shi-Yun Wang. Computational Identification of Lysine Glutarylation Sites Using Positive-Unlabeled Learning. *Current Genomics*, 21(3):204–211, 2020.

[55] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.