JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 07 Issue 01
2024

*Research Article*
https://doi.org/10.56979/701/2024

# Lysine Acetylation Site Prediction in Prokaryotes: A Deep Learning Approach

**Hassan Kaleem[1*], Malik Tahir Hassan[2], Sajid Mahmood[2], and Muhammad Noman Khalid[3]**

[1]Department of Software Engineering Allied Consultants, Irvine California, USA.
[2]School of Systems and Technologies, University of Management and Technology, Lahore, Pakistan.
[3]Medicine and Surgery, Allama Iqbal Medical College, Lahore, Pakistan.
[*]Corresponding Author: Hassan Kaleem. Email: hassan.kaleem@alliedc.com

**Received:** November 11, 2023 **Accepted:** April 21, 2024 **Published:** June 01, 2024

**Abstract:** Post-Translational Modification (PTM) of proteins plays a vital role in both disease and normal states. Protein acetylation is an important PTM in eukaryotes as it greatly changes the properties of a protein including hydrophobicity and solubility. Therefore, in both metabolism and regulatory processes, acetylation and other PTMs perform a critical role. By Investigating and accurately spotting lysine acetylation sites can stop or alter faulty modifications that were previously supposed to occur. This can help in changing the course of microbiological diseases like Bacteremia, UTI's, meningitis and others. Several models have been developed to identify lysine acetylation (Kace) sites with appreciable performances. This manuscript presents an improved approach to identify lysine acetylation (Kace) sites which achieves 0.951, 0.891, 0.813, 0.969, 0.946, and 1.0 MCC for B. subtilis, C. glutamicum, E. coli, G. kaustophilus, M. tuberculosis and S. typhimurium species respectively. Machine Learning algorithms require feature extraction from protein sequences, which is a complex and time taking process. This study has introduced an approach to identify kace sites using a deep learning-based model. The proposed approach significantly outperforms the existing approaches. The experimental results on the benchmark and independent datasets achieve significantly higher accuracy, very close to the actual labels. The source code accurate prokaryotic-lysine-acetylation-site-prediction for the proposed approach is made publicly available online for validation purposes.

**Keywords:** Lysine Acetylation; Post-Translational Modification (PTM); Deep Learning; Protein Acetylation.

## 1. Introduction

There are two steps in protein formation. The first step is transcription and the second is translation. In the transcription phase, first the messenger RNA is formed from DNA which then translates into a specific type of protein based on the information it carries [1]. Every protein has its specific RNA that contains amino acids sequence. When a protein is formed, it may go through some modifications through a process known as Post-Translational Modification (PTM) to enable it to perform various functions. More than 400 types of PTMs have been identified that can occur in human cells [2]. Phosphorylation, Acetylation and Ubiquitination are among the more important post-translational changes [3]. Lysine acetylation (also called Kace) is one of the most commonly occurring PTM with further 3 sub-types described as alpha, sigma and ortho, according to locations. Sigma acetylation is the most important of the three, controlling actin nucleation, cell cycle regulation, chromatin stability and various other important functions including Protein-Protein Interactions (PPIs). The alpha acetylation occurs more commonly in eukaryotes. Disruption in the regulation of Kace can result in aging and various diseases like cancer, immune disorders, cardiovascular problems, etc. Thus, acetylation plays a major role in cell physiology and pathophysiology.

Several experimental methods have been devised including radioactivity methods and mass spectrometry to investigate the exact locations of Kace sites. Recently, Artificial Intelligence (AI) based

methods using machine learning and deep learning models have also been designed for the identification of Kace sites. Although, improvements in speed and accuracy of finding Kace sites have been made, yet limitations like use of elementary algorithms only and limited feature space of methods still need to be addressed and overcome.

Artificial intelligence (AI) is when machines perform tasks that are generally attributed to human intelligence. Machine Learning (ML) is a branch of AI in which machines can obtain skills and learn things. In ML, handcrafted features need to be extracted from the data so that these can be passed to machines (or models) and train them to do predictions in future. The architecture of the machine learning model is shown in Figure 1.

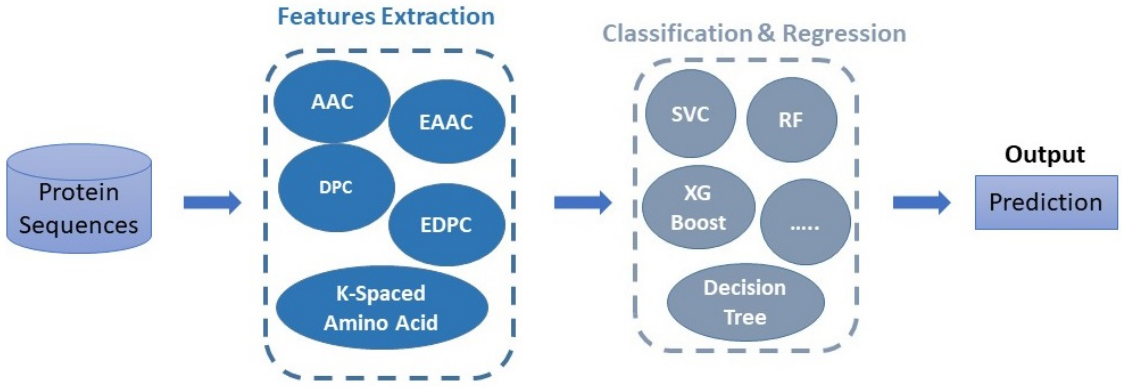



**Figure 1.** Architecture of the Machine Learning Model

Deep Learning (DL) is a branch of ML which uses Artificial Neural Networks (ANN) to learn from data without explicitly providing them with the features of the datasets. We pass raw data to a deep learning model, and it generates features by itself. A neural network is a kind of network which is inspired by the human brain [4]. In deep learning, the model is trained to perform specific tasks by taking raw inputs from labeled data. The dataset could contain sound, text, sequences, or images. The results that are achieved now by deep learning methods were never possible before. These trained models of deep learning can even exceed human-level performance due to ultra-high accuracy. It is because deep learning models use multiple layered neural network architecture and large labeled dataset [5]. The architecture of the deep learning model is shown in Figure 2.

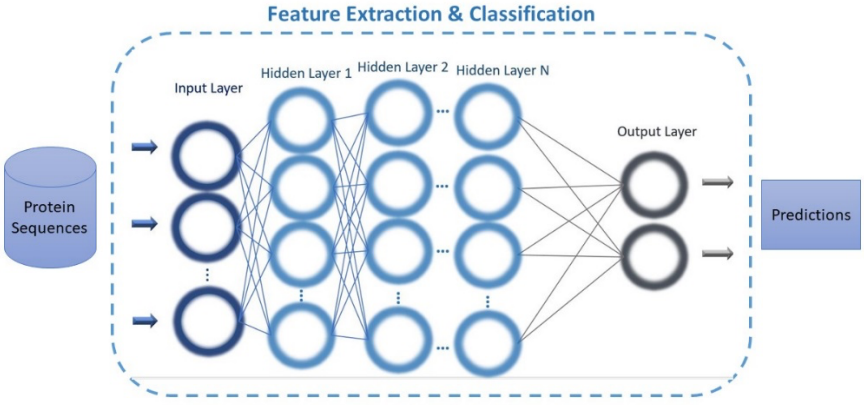



**Figure 2.** Architecture of the Deep Learning Model

Traditionally, many experimental methods have been developed to detect Kace post-translational modifications (PTMs) sites [6]. With the help of technological advancement, detecting performance of Kace sites has been improved significantly. However, if we consider the size of the proteome, we have only identified a small part of the lysine yet as modifying and testing every kind lysine that resides in a protein is a difficult task.

Several ML algorithms have been used in the identification and prediction of lysine acetylation sites. Wankun et al. conducted a study for the prediction of histone acetyltransferase (HAT) specific modification

sites which resides in a protein sequences through GPS PAIL model [7]. They collected 702 Non-redundant acetylation sites which are HATs specific for 205 protein protein substrates. In their experiment, they performed analysis on nine eukaryotic organisms and seven HAT-specific sites. They identify the eukaryotes and prokaryotes sites that play a vital role several biological processes and cellular pathways. There are 58000 acetylation sites identified and characterized in both eukaryotes and prokaryotes species. Multiple algorithms were developed to predict the general acetylation sites, but few algorithms were developed for HAT-specific sites prediction. GPA PAIL provided better results for the identification and prediction of HAT-specific sites.

Qingxiao et al. conducted a study on lysine acetylation sites prediction based on Long-Short Term Memory (LSTM) algorithm [8]. Two machine learning models K-Nearest Neighbor (KNN) and Support Vector Machines (SVM), and a Deep Learning model LSTM, were trained and tested to predict lysine acetylation. However, in their experiments, results concluded that the deep learning-based LSTM model performed better than both machine learning algorithms as KNN algorithm which achieved the accuracy of 0.6969 and SVM model achieved 0.7428. Whereas LSTM model achieved an Accuracy score of 0.8152.

Kai et al. conducted a study on the prediction of reversible Acetyltransferase of Histones (HAT)/Deacetylase of Histones (HDAC)-specific lysine acetylation by using deep learning-based model [9]. They integrated numerous structures of genetic factors with a deep neural network and evaluated the hyper-parameters with element's multitude evaluation, which achieved a notable performance. They employed cross-validations as well as testing sets of data for the validation of the model. The study describes that protein-protein connections could develop enzyme-specific alteration supervisory interactions. Visualization of their results is possible on the Deep PLA server. Moreover, the study also describes cancer's cross-examination of transformation-associated alterations and discovers that regulation of acetylation can be completely disrupted by alterations in cancers and is seriously concerned in the instruction of cancer gesturing. The results of these findings might offer sufficient regulatory information to reveal the implementation of protein alterations in numerous biological procedures and to stimulate the study of the prediction and cure of cancers.

Yingxi et al. conducted a study on Generative Adversarial networks (GAN) [10], the purpose of this study was to predict and analyze of multiple protein lysine modified sites. They worked on a dataset involving 18 lysine modifications. Different features were extracted from the dataset and 1497 features were selected. The model achieved an accuracy of 0.8589 and Matthews correlation coefficient (MCC) of 0.8376 after 10-fold cross-validation testing. On the independent test data, the model achieved 0.8549 Accuracy and 0.8330 MCC values. Generative adversarial networks (GAN) were used to handle the class imbalance problem.

Basith et al. conducted a study on recent trends in developing machine learning approaches for the prediction of lysine acetylation [11]. Their study provides a comprehensive survey on different machine learning predictors. Furthermore, the study has also discussed the key aspects of developing a successful predictor.

Chen et al. developed a predictor called ProAcePred [12] for 9 different prokaryotic species and then further updated it to ProAcePred 2.0 [13] for 6 prokaryotic species.

A stacking-based predictor called STALLION [14] was recently developed for prokaryotic lysine acetylation prediction. STALLION significantly improved the performance accuracy of lysine acetylation site prediction as compared to previous approaches. It achieves the maximum accuracy and Mathew Correlation Coefficient (MCC) values in Kace sites identification. STALLION demonstrates its performance on a dataset containing six different species. These species are Mycobacterium tuberculosis (MT), Bacilus subtilis (BS), E. Coli (EC), Corynebacterium glutamicum (CG), S. typhimurium (ST) and Geobacillus kaustophilus (GK).

Such ML studies give us the idea to understand the differences better in substrate site specificity between prokaryotic and eukaryotic species. Most of the predictors were designed to predict the acetylation in eukaryotes and lacked species specifically.

As STALLION shows the maximum performance, we consider it as a benchmark method and compare our approach with it. It is demonstrated that the proposed method outperforms STALLION for Kace (lysine acetylation sites) prediction. We have applied Deep Learning algorithms for the accurate prediction of lysine acetylation sites. We conducted our experiment on the same six species Mycobacterium

tuberculosis (MT), Bacilus subtilis (BS), E. Coli (EC), Corynebacterium glutamicum (CG), S. typhimurium (ST) and Geobacillus kaustophilus (GK) used by STALLION.

Recently many convolutional neural network (CNN) based algorithms try to improve exploration by adapting mutation strategies during the evolution process in comparison to the hand-crafted CNN architectures and other automatic search methods. The search process becomes fully automatic, eliminating the need for expert knowledge in CNN design. These algorithms take multiple parameters as input and find the best parameters according to the dataset [15] [16] [17].

To date, more than a dozen tools have been developed to identify Kace sites like, Pail [18], LysAcet [19], EnsemblePail [20], N-Ace [21], BPBPHKA [22], PLMLA [23], PSKAcePred [24], KAcePred [25], LAceP [26], AceK [27], SSPKA [28], iPTM-mLys [29], KA-predictor [30], ProAcePred [12], ProAcePred 2.0 [13], Ning et al [31], DNNAce [32], KerasTuner [33], DeepKPred [34], MDC Kace [35] and STALLION [14]. Readers can see the details of these approaches by following the references.

## 2. Materials and Methods

Chen et al. [36] developed a dataset from the PLMD database [37]. It is a non-redundant dataset for six species namely, B. subtilis, C. glutamicum, E. coli, G. kaustophilus, M. tuberculosis and S. typhimurium. The dataset summary is presented in Table 1. They applied CD-HIT [38] to remove the homologous sequences from the dataset by using a sequence identity threshold of 30%. The removal of homologous sequences is crucial for the model performance. The authors also employed varying sequence lengths and concluded that sequences of 21-residue length and K at their center are the best choices. They used an experimental method to validate the Kace (positive) samples and non-Kace (negative) samples. The STALLION [14] also used the same dataset in their study. In this study, we also use the same dataset developed by Chen et al. [36] for a fair comparison of the proposed model with STALLION and ProAcePred 2.0. Table 1. Shows the statistical summary for training and independent dataset.

**Table 1.** Statistical Summary of Dataset

| Species | Positive | Negative | Positive | Negative |
|---|---|---|---|---|
| EC | 6592 | 6592 | 361 | 1384 |
| CG | 1052 | 1052 | 83 | 830 |
| MT | 865 | 865 | 68 | 514 |
| BS | 1571 | 1571 | 125 | 1165 |
| ST | 198 | 198 | 10 | 217 |
| GK | 206 | 206 | 17 | 192 |

The first column represents the species, the second and third column represents the training dataset, and the fourth and fifth column represents the independent dataset for testing. We experimented with several machine learning, Neural Network and Deep Learning methods on the dataset for finding the optimal performing models. The features were extracted through iFeature [39] library and their live server, and Chou's 5-step rule [40] for protein features extraction. Multiple features were extracted including Amino Acid Composition (AAC), Enhanced Amino Acid Composition (EAAC), Dipeptides Composition (DPC), Enhanced Dipeptides Composition (EDPC) and K-Spaced Amino Acid Composition. Different ML algorithms were experimented on the features dataset through Lazypredict [41] [47], a Python library that has an implementation for 40 machine learning algorithms. The best performing algorithm using a method in which dataset is turned into Train and Test split with data size of 80% and 20% respectively achieved the MCC 49% for EC on training dataset, but failed on the independent test dataset [49]. The independent testing achieved 15% overall MCC for all the 6 species [51] [52]. The architecture of the machine learning model is shown in Figure 3.

In our second experiment, a stacking-based deep learning model was applied to the dataset as proposed by Anum et al [42]. This model achieved 67% MCC for ST, 55% for BS, 51% for CG, 49% for MT, 31% for ST and 27% for GK on the training dataset. The model achieved 18% overall MCC for all the 6 species and it also failed to outperform STALLION on the independent dataset [53]. The architecture of the Stacking model is shown in Figure 4.

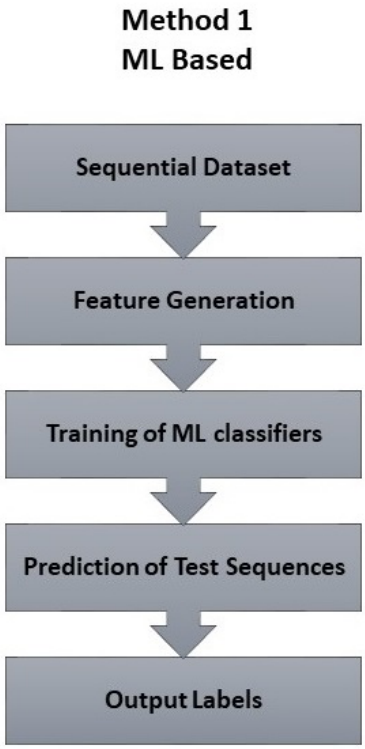



**Figure 3.** Architecture of Machine Learning Model

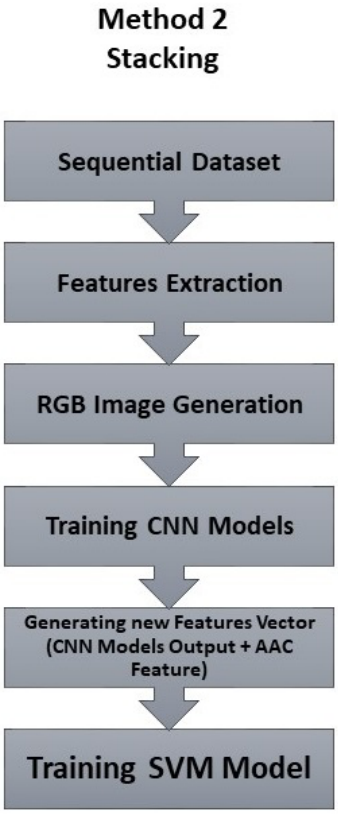



**Figure 4.** Architecture of Stacking Model

After performing these experiments, LSTM was applied to the dataset. LSTM outperformed STALLION on two species by 11% on B. subtilis, and by 41% on E. coli. After that, hyperparameter optimization using KerasTuner [47] [50] was applied to the dataset which outperformed STALLION for all six species. The architecture of the deep learning model is shown in Figure 5.

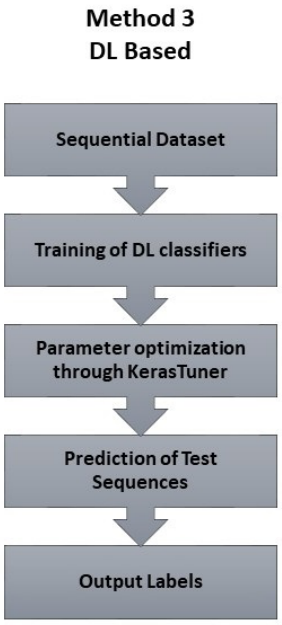



**Figure 5.** Architecture of Deep Learning Model

2.1. Computational Environment

The proposed algorithm is developed by using various technologies including Google Colaboratory and Anaconda Python Jupiter Lab. We used Google Colaboratory because python libraries are already installed on the server, and we don't have to install these manually. Google Colaboratory also provides a Graphics processing unit (GPU) which enhances the training and testing process relatively faster for deep learning algorithms. We have used KerasTuner python library and TensorFlow at the back-end.

2.2. Architecture of proposed Deep Learning Model

In the proposed model, first, we implemented the LSTM model. LSTM model outperformed STALLION for 2 species B. subtilis and E. coli. Parameters used in LSTM are in Table 2.

**Table 2.** Parameters used in LSTM

| **Maximum Features** | **500** |
|---|---|
| Activation Function | Softmax |
| Optimizer | Adam |
| Epochs | 100 |
| Batch-Size | 32 |

LSTM outperformed STALLION by 11% in B. subtilis and by 41% in E. coli. LSTM performed well because the number of the protein sequences for the two species; B. subtilis was 1571 positive and 1571 negative sequences, and E. coli was 6592 positive and 6592 negative sequences. LSTM could not perform well on the other four species due to the smaller datasets.

In the second experiment, KerasTuner is used to tune the model and TensorFlow is used at the back-end. In KerasTuner multiple units are passed to the model, so that it can train and test the model according to these different units. Then the units setting showing the best performance is selected. Table 3 shows parameters used in KerasTuner.

**Table 3.** Parameters used in KerasTuner

| **Maximum Features** | **500** |
|---|---|
| Activation Function Hidden Layer | Relu |
| Activation Function Output Layer | Sigmoid |
| Units | 2,4,8,16,32,64,128 |
| Epochs | 20 |
| Batch-Size | 32 |
| Type | Random Search |
| Objective | Val-Accuracy |
| Max Trials | 10 |

Multiple units were passed to KerasTuner. A neuron or a unit typically represents a single object [44]. Max trails represent the number of hyperparameter combinations that will be tested by KerasTuner and execution per trail is the number of models that will be built and fit to check the robustness of each trail. 2, 4, 8, 16, 32, 64 and 128 units were passed to KerasTuner. Here relu and sigmoid were chosen for activation functions, which makes total of 14 combinations per trail. In this case, max trails are set to 10, and the KerasTuner is configured to find 10 random tuples of the hidden units and activation function. For each trial and execution, KerasTuner is set to fit the model with 20 epochs as configured in the script. Figure 6 shows the proposed deep learning model's training accuracy on given units.

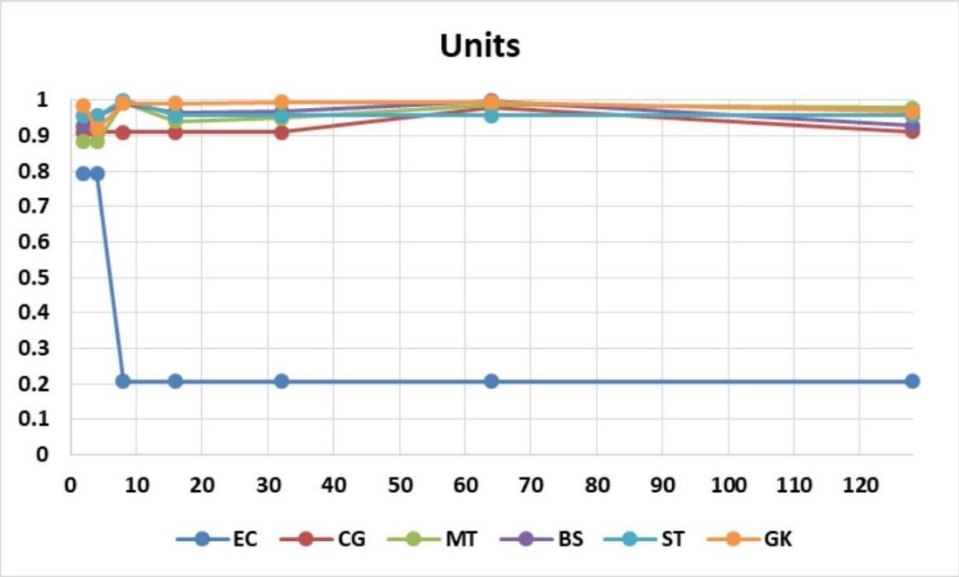



**Figure 6.** Training Accuracy on Given Units

Total 8 parameters were used in KerasTuner. Table 4 shows the relationship between the model and dependent variables. As number of the units are changed for the specie, the accuracy starts to decline.

**Table 4.** Best units used in KerasTuner

| Species | Units |
|---|---|
| E. Coli (EC) | 4 |
| C. Glutamicum (CG) | 64 |
| M. Tuberculosis (MT) | 8 |
| B. Subtilis (BS) | 64 |
| S. Syphimurium (ST) | 8 |
| G. Kaustophilus (GK) | 16 |

2.3. Performance Evaluation Strategies

Four performance measurements that are widely used in other studies [45] [46] are applied here to evaluate the model. These measures include Matthew's correlation coefficient (MCC), Balanced Accuracy (BAcc), Sensitivity (Sn) and Specificity (Sp). Sensitivity is the measure that indicates how many of the positive predictions are actually positive. It's like finding a person who has a disease and predicting him as a patient. Specificity is the measure indicating the accurate predictions of people as non-patients who are actually without a disease. Balanced Accuracy is used when a dataset is not balanced, i.e., there is one class in the majority and the other in minority. The performance values of models trained on imbalanced datasets could be misleading. The F1 score balances the precision and recall and thus is a better indicator of a classifier's performance as compared to accuracy. Matthew's correlation coefficient is more useful, robust, and reliable, because it calculates a combined score based on given labels and predicted labels.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (1)$$

$$BAcc = \frac{Sn + Sp}{2} \quad (2)$$

$$Sn = \frac{TP}{TP + FN} \quad (3)$$

$$Sp = \frac{TN}{TN + FP} \quad (4)$$

Where TP stands for True Positives, TN is True Negatives, FP is False Positives and FN is False Negatives.

**3. Results**

In the first experimental method, machine learning was used to predict the Kace sites. In the first step, features were extracted from protein sequences from EC and ST species. EC dataset contains the most sequences and ST is the smallest among six species. Multiple feature extraction methods were used. A total of 16200 features were extracted from different features extraction method like Amino Acid Composition, Enhanced Amino Acid Composition, Dipeptides Composition, Tripeptides Composition, K-space amino acid, and they were fused together. These features were passed to machine learning classifier separately as well as fused. Total 40 machine learning classifier were tested. 80% data were used for training and 20% were used for testing. After prediction accuracy on the training dataset, all classifiers were then analyzed. LGBM classifier achieved 55% MCC on EC and 11% MCC on ST, which is higher than STALLION model for EC on training dataset, but the classifier failed on the independent dataset. LGBM achieved 9% MCC on EC and 0.25% MCC on ST. The prediction scores of machine learning classifiers were not up to the mark. In order to improve the results, we then tried neural network-based approach to improve the prediction results.

In the stacking method, features were extracted from GAP 0,1,2,3 method and passed to the Neural Network based classifier, CNN, separately. After that, all trained models were saved and passed to the Support Vector Machine (SVM) classifier with the addition of amino acid composition features. 80% data were used for training and 20% were used for testing. On the training dataset Stacking model achieved 59% MCC on EC and 18% MCC on ST. It is higher than the STALLION model for EC. On the independent testing the Stacking model achieved 19.25% MCC on EC and 0.02 on ST. Prediction score of neural network based stacking model was not very good. We then tried with advanced deep learning based model to improve the accuracy.

Both approaches failed because they involve feature extraction from the protein sequences. Preprocessing had been carried out already on these datasets in a way that increased the accuracy of the ProAcePred 1.0 model compared to ProAcePred 2.0. The First 30% threshold CD-HIT was applied to remove similar sequences. After that protein sequence was reduced to 21 sizes and K was placed in the center of the sequence. Due to this, all feature extraction methods failed to extract sufficient features from the dataset. To overcome this problem, we implemented deep learning algorithms which generate features by themselves. But there are some problems in the deep learning algorithms as well because these require large training samples to train themselves.

Firstly, Long-Short-Term-Memory (LSTM) model was tested, LSTM model is the most cited algorithm in the past decade. This model is designed in a way that it does not forget the information as the connection in the neural network recur over a longer period. LSTM require a large amount of dataset to train upon. Because of this reason, the LSTM model performed well on E. coli as it has 13184 sequences and B. subtilis as it has 3142 sequences for training the model. This is why the LSTM model outperformed STALLION achieving 41% better MCC on E. coli and 11% MCC on B. subtilis in the independent testing but failed to perform very well in the other 4 species because they have lesser sequences then these two species. The proposed model, optimized using KerasTuner, was first trained on the training dataset and produced significantly better results with a split of 80-20 as training and testing data respectively. The results on the training data show that the proposed deep learning model outperformed STALLION on the training dataset. MCC values of the produced results of the two models are shown in Table 6. The details of the

specie-specific values of the results produced by the proposed deep learning model for different performance measures are detailed in Table 6.

**Table 5.** STALLION and Proposed Deep Learning Results on Training Dataset. (5-fold cross-validation testing)

| Species | STALLION (MCC) | Proposed DL (MCC) |
|---|---|---|
| EC | 0.390 | 0.784 |
| CG | 0.329 | 0.995 |
| MT | 0.380 | 0.994 |
| BS | 0.295 | 0.997 |
| ST | 0.202 | 1.000 |
| GK | 0.259 | 0.956 |

**Table 6.** Proposed Deep Learning Results on Training Dataset (5-fold cross-validation testing)

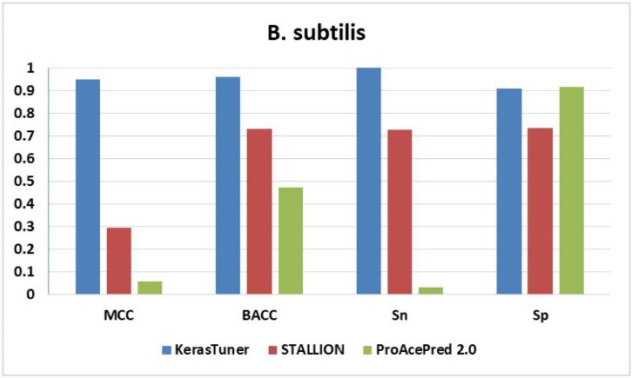
| Species | MCC | BAcc | Sn | Sp |
|---|---|---|---|---|
| EC | 0.784 | 0.881 | 1.000 | 0.761 |
| CG | 0.995 | 0.998 | 1.000 | 0.995 |
| MT | 0.994 | 0.997 | 0.994 | 1.000 |
| BS | 0.997 | 0.998 | 1.000 | 0.997 |
| ST | 1.000 | 1.000 | 1.000 | 1.000 |
| GK | 0.956 | 0.986 | 1.000 | 0.972 |

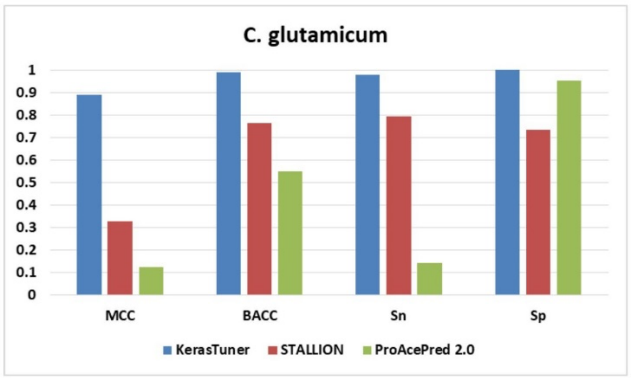
3.1. Comparative Analysis

Two deep learning approaches, Long-Short Term Memory (LSTM) and Sequential model were trained and compared. First, LSTM model was trained that produced good results on the training dataset. The model also performed well on the independent dataset and outperformed STALLION for two species B. subtillis and E. coli. However, it could not produce good results for other species.

Secondly, the sequential model from Keras library was trained. KerasTuner was used for the tuning of the model. KerasTuner is a hyperparameter optimization framework that finds the best hyperparameter values for a model. It uses Bayesian Optimization, Hyperband and Random Search algorithms. We have used random Search to tune the hyperparameters of our model. The proposed deep learning model was trained on the training dataset. After the training of the model, the independent dataset was used to evaluate the model's performance.

The model achieved 0.951, 0.891, 0.813, 0.969, 0.946, and 1.0 MCC values for B. subtilis, C. glutamicum, E. coli, G. kaustophilus, M. tuberculosis and S. typhimurium, respectively. The results produced by the tuned model on the benchmark dataset outperformed all the other predictors on the independent dataset including STALLION and ProAcePred 2.0. Figure 7 shows a comparison of the proposed deep learning model with STALLION and ProAcePred 2.0 on the independent set.



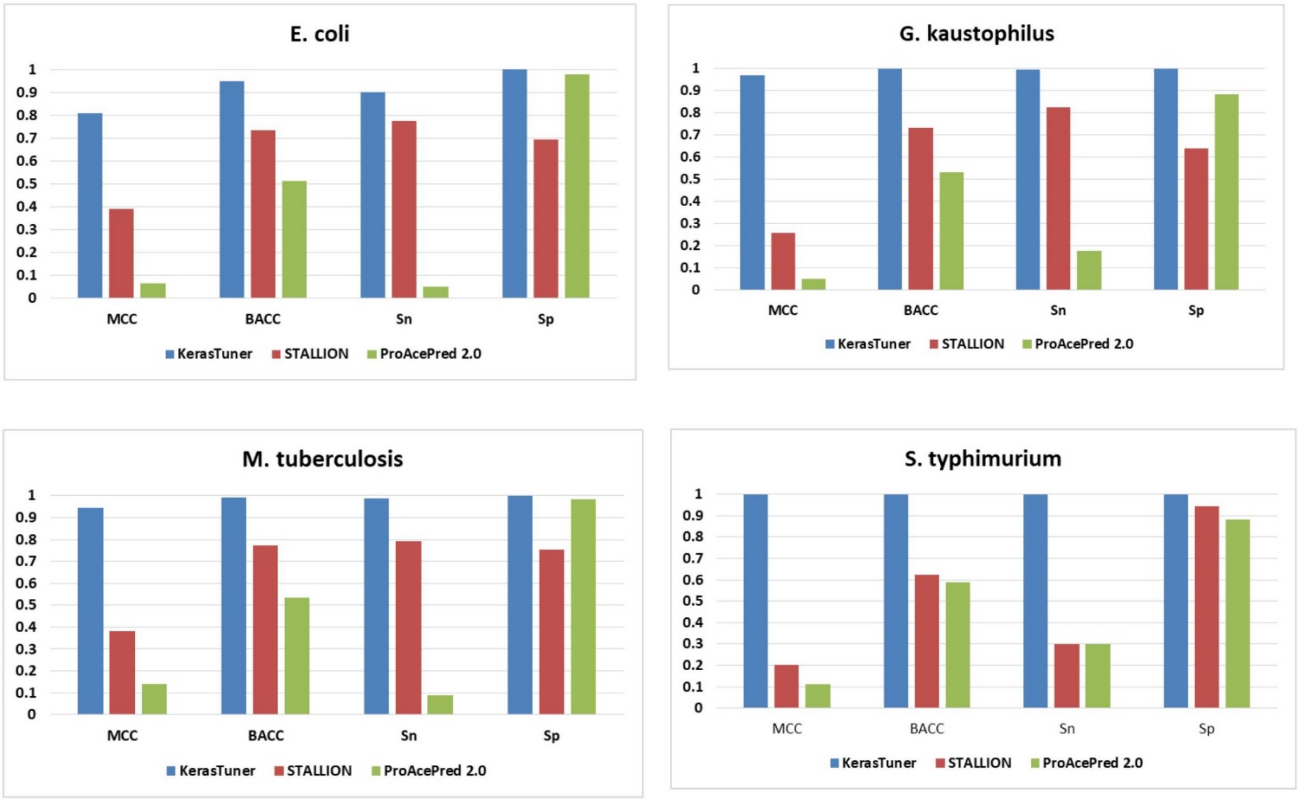



**Figure 7.** Performance comparison of the proposed approach using KerasTuner, STALLION, and ProAcePred 2.0 in classifying Kace and non-Kace sites on the independent test

## 4. Discussion

While conducting the experiments, we observed that deep learning (DL) classifiers are consuming lesser time than machine learning (ML) classifiers. For ML classifiers, first, we have to extract features from the protein sequences. After extraction, these features are passed to the ML classifiers for training. This process takes a lot of time and resources, and its results are still not as accurate as DL classifiers. But there are some limitations in DL classifiers as well. These require a large amount of data to train upon. Due to this very reason, the LSTM model performed well on E. coli having 13184 sequences and B. subtilis having 3142 sequences for training the model, and outperformed STALLION achieving 41% better MCC in case of E. coli and 11% better MCC in case of B. subtilis on the independent set but failed to perform well on the other 4 species having lesser number of sequences. Hyperparameter optimization was performed using KerasTuner, a hyperparameter tuning framework, to overcome this issue. It trains and tests the model on different hyperparameter settings (number of layers and number of nodes in a particular layer) and selects the setting showing the best performance. After conducting the hyperparameter tuning with KerasTuner, the model outperforms all the other classifiers for prokaryotic lysine acetylation site prediction.

For future research direction, both LSTM and KerasTuner based classifiers' code has been shared. We have performed experiment on six species in this study. The same code can be used for other species, and for future development of prediction algorithms in the field of bioinformatics. If a dataset consists of a large number of training sequences, LSTM can be used to design the prediction system. However, if the dataset is small then KerasTuner can be used effectively to optimize hyperparameters for designing a reasonably good prediction system.

## 5. Conclusions

Multiple ML, neural network and stacking-based deep learning models were designed and applied in this work. For ML and neural network-based methods, features extraction is necessary. We have extracted features of protein sequences through multiple ways. These features were given to the ML and neural network-based classifiers to identify Kace sites. In 5-fold cross-validation on the training dataset,

these models performed well but failed to show good performance on the independent dataset. Stacking-based deep learning model uses GAP 0, 1, 2, and 3 features. All these features were fused and given to the model for training. This technique also failed on the independent dataset. Data fusion was also applied to the experiment generating 16021 features for a single protein sequence. But it outperformed only one specie E. coli. The deep learning algorithms outperformed all the other ML-based models for prokaryotic lysine acetylation site prediction. First LSTM was experimented. LSTM outperformed STALLION in two species, B. subtilis and E. coli, but failed to outperform on the other four species. Finally, the proposed deep learning model tuned using KerasTuner, a hyperparameter optimization framework, outperformed the other predictors in all six species and achieved the best accuracy.

**References**


1. FRANCIS CRICK. Central dogma of molecular biology. Nature, 227:561–563, 08 1970.
2. Shahin Ramazi and Javad Zahiri. Post-translational modifications in proteins: resources, tools and prediction methods. Database, 2021, 2021.
3. Suk-Chul Bae and Yong Hee Lee. Phosphorylation, acetylation and ubiquitination: The molecular basis of runx regulation. Gene, 366:58–66, 01 2006.
4. Jürgen Schmidhuber. Deep learning in neural networks: An overview. Neural networks, 61:85–117, 2015.
5. Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. Eai Endorsed Transactions on Security and Safety, 3(9):e2, 2016.
6. Katalin F. Medzihradszky. Peptide sequence analysis. Methods in Enzymology, 402:209–244, 2005.
7. Wankun Deng, Chenwei Wang, Ying Zhang, Yang Xu, Shuang Zhang, Zexian Liu, and Yu Xue. Gps-pail: prediction of lysine acetyltransferasespecific modification sites from protein sequences. Scientific Reports, 6, 12 2016.
8. Qingxiao Xiu, Dancheng Li, Hailong Li, Ning Wang, and Chen Ding. Prediction method for lysine acetylation sites based on lstm network. 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), 10 2019.
9. Kai Yu, Qingfeng Zhang, Zekun Liu, Yimeng Du, Xinjiao Gao, Qi Zhao, Han Cheng, Xiaoxing Li, and Ze-Xian Liu. Deep learning based prediction of reversible hat/hdac-specific lysine acetylation. Briefings in Bioinformatics, 21:1798–1805, 11 2020.
10. Yingxi Yang, Hui Wang, Wen Li, Xiaobo Wang, Shizhao Wei, Yulong Liu, and Yan Xu. Prediction and analysis of multiple protein lysine modified sites based on conditional wasserstein generative adversarial networks. BMC Bioinformatics, 22, 03 2021.
11. Shaherin Basith, Hye Jin Chang, Saraswathy Nithiyanandam, Tae Hwan Shin, Balachandran Manavalan, and Gwang Lee. Recent trends on the development of machine learning approaches for the prediction of lysine acetylation sites. Current Medicinal Chemistry, 29:235–250, 01 2022.
12. Guodong Chen, Man Cao, Kun Luo, Lina Wang, Pingping Wen, and Shaoping Shi. Proacepred: prokaryote lysine acetylation sites prediction based on elastic net feature optimization. Bioinformatics, 34:3999–4006, 06 2018.
13. Guodong Chen, Man Cao, Jialin Yu, Xinyun Guo, and Shaoping Shi. Prediction and functional analysis of prokaryote lysine acetylation site by incorporating six types of features into chou's general pseaac. Journal of Theoretical Biology, 461:92–101, 01 2019.
14. Shaherin Basith, Gwang Lee, and Balachandran Manavalan. Stallion: a stacking-based ensemble learning framework for prokaryotic lysine acetylation site prediction. Briefings in bioinformatics, 23(1):bbab376, 2022.
15. Hassan Kaleem, Sundas Rukhsar, and Muhammad Noman Khalid. Anti-cancer peptides prediction: A deep learning approach. Journal of Computing amp; Biomedical Informatics, 3(02):144–151, 2022.
16. Yu Xue, Yankang Wang, Jiayu Liang, and Adam Slowik. A self-adaptive mutation neural architecture search algorithm based on blocks. IEEE Computational Intelligence Magazine, 16(3):67–78, 2021.
17. Yu Xue and Jiafeng Qin. Partial connection based on channel attention for differentiable neural architecture search. IEEE Transactions on Industrial Informatics, 2022.
18. Ao Li, Yu Xue, Changjiang Jin, Minghui Wang, and Xuebiao Yao. Prediction of n-acetylation on internal lysines implemented in bayesian discriminant method. Biochemical and Biophysical Research Communications, 350:818–824, 12 2006.
19. Songling Li, Hong Li, Mingfa Li, Yu Shyr, Lu Xie, and Yixue Li. Improved prediction of lysine acetylation by support vector machines. Protein and Peptide Letters, 16:977–983, 08 2009.
20. Yan Xu, Xiao-Bo Wang, Jun Ding, Ling-Yun Wu, and Nai-Yang Deng. Lysine acetylation sites prediction using an ensemble of support vector machine classifiers. Journal of Theoretical Biology, 264:130–135, 05 2010.
21. Tzong-Yi Lee, Justin Bo-Kai Hsu, Feng-Mao Lin, Wen-Chi Chang, PoChiang Hsu, and Hsien-Da Huang. N-ace: Using solvent accessibility and physicochemical properties to identify protein n-acetylation sites. Journal of Computational Chemistry, 31:2759–2771, 09 2010.
22. Jianlin Shao, Dong Xu, Landian Hu, Yiu-Wa Kwan, Yifei Wang, Xiangyin Kong, and Sai-Ming Ngai. Systematic analysis of human lysine acetylation proteins and accurate prediction of human lysine acetylation through birelative adapted binomial score bayes feature representation. Molecular BioSystems, 8:2964, 2012.

23. Shao-Ping Shi, Jian-Ding Qiu, Xing-Yu Sun, Sheng-Bao Suo, Shu-Yun Huang, and Ru-Ping Liang. Plmla: prediction of lysine methylation and lysine acetylation by combining multiple features. Molecular BioSystems, 8:1520, 2012.
24. S. B. Suo, J. D. Qiu, S. P. Shi, X. Y. Sun, S. Y. Huang, X. Chen, and R. P. Liang. Position-specific analysis and prediction for protein lysine acetylation based on multiple features. PloS one, 7:e49108–e49108, 2012.
25. Sheng-Bao Suo, Jian-Ding Qiu, Shao-Ping Shi, Xiang Chen, Shu-Yun Huang, and Ru-Ping Liang. Proteome-wide analysis of amino acid variations that influence protein lysine acetylation. Journal of Proteome Research, 12:949–958, 01 2013.
26. Ting Hou, Guangyong Zheng, Pingyu Zhang, Jia Jia, Jing Li, Lu Xie, Chaochun Wei, and Yixue Li. Lacep: Lysine acetylation site prediction using logistic regression classifiers. PLoS ONE, 9:e89575, 02 2014.
27. Cheng-Tsung Lu, Tzong-Yi Lee, Yu-Ju Chen, and Yi-Ju Chen. An intelligent system for identifying acetylated lysine on histones and nonhistone proteins. BioMed Research International, 2014:1–11, 2014.
28. Yuan Li, Mingjun Wang, Huilin Wang, Hao Tan, Ziding Zhang, Geoffrey I. Webb, and Jiangning Song. Accurate in silico identification of species specific acetylation sites by integrating protein sequence-derived and functional features. Scientific Reports, 4, 07 2014.
29. Wang-Ren Qiu, Bi-Qian Sun, Xuan Xiao, Zhao-Chun Xu, and Kuo-Chen Chou. iptm-mlys: identifying multiple lysine ptm sites and their different types. Bioinformatics, 32:3116–3123, 06 2016.
30. Qiqige Wuyun, Wei Zheng, Yanping Zhang, Jishou Ruan, and Gang Hu. Improved species-specific lysine acetylation site prediction based on a large variety of features set. PLOS ONE, 11:e0155370, 05 2016.
31. Qiao Ning, Miao Yu, Jinchao Ji, Zhiqiang Ma, and Xiaowei Zhao. Analysis and prediction of human acetylation using a cascade classifier based on support vector machine. BMC Bioinformatics, 20, 06 2019.
32. Bin Yu, Zhaomin Yu, Cheng Chen, Anjun Ma, Bingqiang Liu, Baoguang Tian, and Qin Ma. Dnnace: Prediction of prokaryote lysine acetylation sites through deep neural networks with multi-information fusion. Chemometrics and Intelligent Laboratory Systems, 200:103999, 05 2020.
33. Kaleem, H., Hassan, M. T., Mahmood, S., & Khalid, M. N. (2023). Deep Learning Algorithms to Predict m7G from Human Genome. Journal of Computing & Biomedical Informatics, 4(02), 110-116.
34. Fan, S., & Xu, Y. (2024). DeepKPred: Prediction and Functional Analysis of Lysine 2-Hydroxyisobutyrylation Sites Based on Deep Learning. *Annals of Data Science*, *11*(2), 693-707.
35. Huiqing Wang, Zhiliang Yan, Dan Liu, Hong Zhao, and Jian Zhao. Mdckace: A model for predicting lysine acetylation sites based on modular densely connected convolutional networks. IEEE Access, 8:214469– 214480, 2020.
36. Guodong Chen, Man Cao, Jialin Yu, Xinyun Guo, and Shaoping Shi. Prediction and functional analysis of prokaryote lysine acetylation site by incorporating six types of features into chou's general pseaac. Journal of Theoretical Biology, 461:92–101, 2019.
37. Haodong Xu, Jiaqi Zhou, Shaofeng Lin, Wankun Deng, Ying Zhang, and Yu Xue. Plmd: an updated data resource of protein lysine modifications. Journal of Genetics and Genomics, 44(5):243–250, 2017.
38. Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. Bioinformatics, 28(23):3150–3152, 2012.
39. Zhen Chen, Pei Zhao, Fuyi Li, André Leier, Tatiana T Marquez-Lago, Yanan Wang, Geoffrey I Webb, A Ian Smith, Roger J Daly, Kuo-Chen Chou, et al. ifeature: a python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics, 34(14):2499–2502, 2018.
40. Sharaf Jameel Malebary, Muhammad Safi Ur Rehman, and Yaser Daanial Khan. icrotok-pseaac: Identify lysine crotonylation sites by blending position relative statistical features according to the chou's 5-step rule. PloS one, 14(11):e0223993, 2019.
41. Ahmed, F., Sumra, I. A., & Jamil, U. (2024). A Comprehensive Review on DDoS Attack in Software-Defined Network (SDN): Problems and Possible Solutions. Journal of Computing & Biomedical Informatics, 7(01).
42. Munir, A., Sumra, I. A., Naveed, R., & Javed, M. A. (2024). Techniques for Authentication and Defense Strategies to Mitigate IoT Security Risks. Journal of Computing & Biomedical Informatics, 7(01).
43. SR Pandala. Lazypredict - https://github.com/shankarpandala/lazypredict, 2020.
44. Anum Rauf, Aqsa Kiran, Malik Tahir Hassan, Sajid Mahmood, Ghulam Mustafa, and Moongu Jeon. Boosted prediction of antihypertensive peptides using deep learning. Applied Sciences, 11(5):2316, 2021.
45. Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner. https://github.com/keras-team/ keras-tuner, 2019.

46. Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. Neural networks, 10(9):1659–1671, 1997.
47. Ran Su, Jie Hu, Quan Zou, Balachandran Manavalan, and Leyi Wei. Empirical comparison and analysis of web-based cell-penetrating peptide prediction tools. Briefings in Bioinformatics, 21:408–420, 01 2019.
48. Shaherin Basith, Balachandran Manavalan, Tae Hwan Shin, and Gwang Lee. Machine intelligence in peptide therapeutics: A next-generation tool for rapid disease screening. Medicinal Research Reviews, 40:1276–1314, 01 2020.
49. Batool, S., Abid, M. K., Salahuddin, M. A., Aziz, Y., Naeem, A., & Aslam, N. (2024). Integrating IoT and Machine Learning to Provide Intelligent Security in Smart Homes. Journal of Computing & Biomedical Informatics, 7(01), 224-238.
50. Abbas, F., Iftikhar, A., Riaz, A., Humayon, M., & Khan, M. F. (2024). Use of Big Data in IoT-Enabled Robotics Manufacturing for Process Optimization. Journal of Computing & Biomedical Informatics, 7(01), 239-248.
51. Khan, A. H., Malik, H., Khalil, W., Hussain, S. K., Anees, T., & Hussain, M. (2023). Spatial Correlation Module for Classification of Multi-Label Ocular Diseases Using Color Fundus Images. Computers, Materials & Continua, 76(1).
52. Khan, M. I., Khan, Z. A., Imran, A., Khan, A. H., & Ahmed, S. (2022, May). Student Performance Prediction in Secondary School Education Using Machine Learning. In 2022 8th International Conference on Information Technology Trends (ITT) (pp. 94-101). IEEE.
53. Abbas, F., Iftikhar, A., Riaz, A., Humayon, M., & Khan, M. F. Use of Big Data in IoT-Enabled Robotics Manufacturing for Process Optimization.