DrillBit

The Report is Generated by DrillBit Grammar Checker Software

Submission Information

Author Name	Rohan Ravi Vernekar
Title	BioMLStudio - An Automated ML Platform for Bioinformatics
Paper/Submission ID	4374867
Submitted by	asklibrarian@acharya.ac.in
Submission Date	2025-09-17 21:01:01
Document type	Project Work

Submitted Text

Characters	Words	Sentences	Lines
39771	5139	247	745

Reading and Execution Time

Reading	Speaking	Execution
0 Hr, 25 Min	0 Hr, 38 Min	0 Hr, 1 Min

Result Information

Grammar Quality **67.68 %**
(Except Similarity & AI Content)

1. Phrases Quality	66.63 %	
2. Non-Duplicate Content	90.01 %	*(Duplicate* 9.98 %*)*
3. Indexed Content	25.39 %	
4. Grammar Info.	88.69 %	*(Mistakes* 31, *Suggession* 52*)*

Detailed Analysis

1. Phrases Quality

Only Alphabets	Only Numbers	Alpha-numeric	Words with Special Chars.,
71.78 %	02.54 %	00.35 %	25.31 %

Unique Words 39.67 %

Counts the number of terms that are used just once in your document.

Rare Words 53.10 %

Words that are not part of the 5,000 most common words in the English language.

Common Words 35.39 %

Words that are part of the 1,000 most common words in the English language.

Word Length 06.49

Measures average word length (Characters per word).

Sentence Length 20.10

Measures average sentence length (words per sentence).

---

2. Non-Duplicate Content

Duplicate Sentences

Sentence No.	Characters	Words	Repetition	% in Report	
1	99	16	2	0.622 %	view
2	108	13	2	0.505 %	view
3	165	22	2	0.856 %	view
4	198	27	2	1.050 %	view
5	154	20	2	0.778 %	view
6	152	22	2	0.856 %	view

Duplicate Sub-Strings

Sub-Strings No.	Characters	Words	Repetition	% in Report	
1	83	10	2	0.389 %	view
2	83	10	2	0.389 %	view
3	88	10	2	0.389 %	view
4	58	9	2	0.350 %	view
5	59	9	2	0.350 %	view
6	56	9	2	0.350 %	view
7	60	7	2	0.272 %	view
8	49	6	2	0.233 %	view
9	21	5	3	0.291 %	view
10	40	5	2	0.194 %	view
11	35	5	2	0.194 %	view
12	56	5	2	0.194 %	view
13	31	4	3	0.233 %	view
14	27	4	2	0.155 %	view
15	19	4	2	0.155 %	view
16	25	4	2	0.155 %	view
17	32	4	3	0.233 %	view
18	30	4	2	0.155 %	view

19	32	4	**2**	**0.155 %**	view
20	31	4	**2**	**0.155 %**	view
21	30	4	**2**	**0.155 %**	view
22	28	4	**2**	**0.155 %**	view

3. Indexed content

Sl. No	**Index**	**Lines**	**Words**	**% in Report**	
1	Methodology	529	3428	**66.70 %**	view
2	Other Data	215	1708	**33.23 %**	view

Submitted Text:

Line 1| BioMLStudio - An Automated ML Platform for
Line 2| Bioinformatics Prof. Vinutha Raghu Rahul Vashist
Line 3| Assistant Professor Department of Information Science and Engineering
Line 4| Department of Information Science and Engineering Acharya Institute of Technology, Bengaluru, India
Line 5| Acharya Institute of Technology, Bengaluru, India rahulr.22.beis#acharya.ac.in
Line 6| vinutha2776#acharya.ac.in Rohan Ravi Vernekar Shetty Vamshik Sudhakar
Line 7| Department of Information Science and Engineering Department of Information Science and Engineering
Line 8| Acharya Institute of Technology, Bengaluru, India Acharya Institute of Technology, Bengaluru, India
Line 9| rohanr.22.beis#acharya.ac.in shettys.22.beis#acharya.ac.in
Line 10| Someshwar R Halewadimath
Line 11| Department of Information Science and Engineering
Line 12| Acharya Institute of Technology, Bengaluru, India
Line 13| someshwarr.22.beis#acharya.ac.in Abstract- Artificial intelligence and machine learning are
Line 14| widely applied in the life sciences. However, using machine
Line 15| learning to analyze complex data and train models is
Line 16| challenging because many researchers lack the domain-
Line 17| specific knowledge needed to operate these ML tools. An
Line 18| AI-powered platform that can perform the entire process
Line 19| from preprocessing to model export would be extremely
Line 20| valuable to the life science community. With easy access to
Line 21| various ML models, the platform would help researchers
Line 22| analyze complex datasets more easily. Nowadays, most
Line 23| AI/ML libraries require advanced programming, machine
Line 24| learning, data preprocessing, and visualization skills. So, in
Line 25| this research, we propose a web-based, AI-enabled
Line 26| platform that is capable of preprocessing, training,
Line 27| evaluating, visualizing, and exporting trained models
Line 28| without any coding expertise. By integrating machine
Line 29| learning and deep neural network models, our platform
Line 30| assists in recognizing, classifying, clustering, and predicting
Line 31| a wide range of multi-modal and multi-sensor datasets,
Line 32| including images, language, and numerical data, for drug
Line 33| discovery, protein sequence classification, and medical
Line 34| diagnostics. Index Terms - Machine learning, Bioinformatics, Neural
Line 35| networks. I. INTRODUCTION
Line 36| In the last 10 years, life sciences have become one of the
Line 37| largest producers of raw data in world. Genome sequencing,
Line 38| protein studies, and medical imaging all generate huge datasets

every day. These datasets hold valuable insights, but their size. and complexity make them very difficult to handle with traditional analysis methods available.

Machine learning (ML) has shown a great promise in solving this challenge. It can identify patterns in high-dimensional data, make predictions, and support tasks like disease detection or protein classifications.

Researchers have successfully used ML for drug discovery, protein analysis, make predictions for advance research and many other applications Even with the success, many biologists and healthcare professionals are unable to use ML directly because of lack of knowledge in using these models. Most ML tools require good programming knowledge, strong math skills, and experience in data science. For researchers without this background, applying ML becomes very tough.

To reduce this gap, we have proposed BioMLStudio, a no code, web-based platform for automated machine learning in bioinformatics. The system is designed so that a researcher can upload a dataset, choose or let the system recommend a model, easy to handle, run training, check results, visualize results, and finally download a trained model - all through a simple interface. By combining traditional ML methods with deep learning architectures, BioMLStudio will support multiple types of biological data such as text, images, and sequences. The main goal of this platform is to make ML more accessible to life science researchers by hiding the technical complexity of using these ML models and to train them, BioMLStudio allows users to focus on their research problems while still benefiting from advanced AI tools.

. II. PROBLEM STATEMENT

Biomedical researchers now generate vest heterogeneous data sets (sequencing reads, mass spec profiles, clinical test, pathology images and other multimodal measurements). However, applying a modern machine learning to these data is still obstructed by the requiring barriers. (1) Technical access- many life science researchers lack programming skills and they face of very steep learning curve to use the Machine Learning libraries and ML pipeline. (2) Integration and Reproducibility- preprocessing, model selection, hyperparameter tuning and visualization are imploded across tools, making end to end experiments hard to reproduce and compare for non-programmers. (3) Domain's suitability and interpretability- sole purpose machinery learning often fails to capture the domain constants like class in balance, batch effect, multimodal alignment and do not provide any kind of explanation that clinicians and biologists require to trust the model outputs.

Biomedical researchers now days generate very large and varied datasets are containing sequencing reads, clinical tests, pathology images and many other measurements and readings. But using modern machine learning on these data is still hard for many of the researchers. There are a few main problems.

First, technical access. Many life-science researchers do not have strong programming skills. Learning to use ML libraries, manage software environments, and build a full ML pipeline takes a long time. This keeps many biologists and clinician's dependent on specialist programmers or slows their work while they learn.

Second, integration and reproducibility. Steps like cleaning data, choosing features, tuning models, validating results, and making plots are often done in different tools or scripts. That makes experiments hard to repeat or compare, especially for people who are not programmers. Work moves between notebooks, command lines, and cloud services with little record of exactly what was done.

Third, domain suitability and interpretability. Off-the-shelf ML tools often do not handle biology-specific issues such as strong class imbalance, batch effects, missing values that matter, or combining different types of data. They also tend to act like “black boxes” and do not give explanations that clinicians or biologists can trust. Without clear, interpretable reasons for predictions, people are reluctant to use the results in experiments or practice.

There are also practical gaps in scaling and governance. Models built in one place are hard to package, version, and share. Many projects lack proper provenance (who changed what and when), model versioning, or exportable audit logs. This makes collaboration, reuse, and any move toward clinical or production use difficult.

In short, the combination of hard technical entry barriers, fragmented toolchains, domain-specific data challenges, and poor interpretability limits the impact of ML in biology.

BioMLStudio brings these pieces together in one simple graphical interface so more researchers can run reliable and explainable, and reproducible ML experiments without learning advanced programming first.

III. LITERATURE SURVEY

Various technological areas such as artificial intelligence, machine learning, bioinformatics, and cloud computing have driven the rapid progress of intelligent biomedical analysis systems with particular focus on data preprocessing, feature extraction, and model training.

A. End-to-End No-Code ML Pipelines for Bioinformatics

A no-code machine-learning pipeline for bioinformatics was introduced in [1]. This approach helps researchers to build and deploy ML models without any need of coding knowledge. These pipelines automate data preprocessing, data cleaning, model selection, training steps, and evaluation. This work demonstrates the importance of accessibility for domain experts, particularly biologists and clinicians, who do not have any advanced computational skills. Similar efforts in AutoML such as Auto-WEKA [5], auto-sklearn [6], and Google AutoML [7]demonstrate the potential of automation to streamline the ML lifecycle.

However, this pipeline has two major limitations: it provides only limited integration of deep learning methods and lacks various visualisation features that require advanced fusion approaches [12]. Moreover, various multi-model datasets such as biomedical images, genetic sequences, and clinical data are restricted as they require advanced technology. These limitations highlight the need for platforms like BioML Studio, which aim to combine no-code usability with advanced ML capabilities and features.

B. Machine Learning Techniques for Protein Structure Prediction in Bioinformatics

Protein structure prediction has been recognized as one of biggest challenge in bioinformatics, which include complex and high dimensionality of biological data [2]. This paper surveys machine learning methods which are applied for protein structure prediction, including supervised learning, deep learning, image generation techniques, and computational models. The researchers found out that the potential of machine learning to predict secondary and tertiary structures are more efficient than the traditional physics-based methods. AlphaFold [8] and RoseTTAFold [9] these ml models have made breakthroughs by utilizing attention-based architectures to model protein folding. Senior et al. [10] demonstrated that learned statistical potentials can improve their accuracy compared to physics-only methods.

Despite this, there are challenges that the models faces as it often require huge datasets and computing resources, and their predictions are not always easy to interpret [11]. To overcome these issues, hybrid approaches that combine ML predictions with biophysical knowledge are being explored to improve reliability and make predictions more correct [15]. Overall, these advances highlight the revolutionary role of ML in protein structure prediction, while also pointing out the need for platforms like BioMLStudio that make such methods more easy for non-programmers.

C. Review of No-Code Machine Learning Platforms for Bioinformatics

A no-code machine-learning pipeline for bioinformatics was introduced in [1]. This approach helps researchers to build and deploy ML models without any need of coding knowledge. These pipelines automate data preprocessing, data cleaning, model selection, training steps, and evaluation. This work demonstrates the importance of accessibility for domain experts, particularly biologists and clinicians, who do not have any advanced computational skills. Similar efforts in AutoML—such as Auto-WEKA [5], auto-sklearn [6], and Google AutoML [7]—demonstrate the potential of automation to streamline the ML lifecycle.

However, this pipeline has two major limitations: it provides only limited integration of deep learning methods and lacks various visualisation features that require advanced fusion approaches [12]. Moreover, various multi-model datasets such as biomedical images, genetic sequences, and clinical data are restricted as they require advanced technology. These limitations highlight the need for platforms like BioML Studio, which aim to combine no-code usability with advanced ML capabilities and features.

D. Deep Learning for Genomic Sequence Classification

The use of deep learning techniques for classification of genomic sequences [4]. CNNs can be used to detect short sequence motifs, while RNNs and LSTMs capture long-range dependencies [13], [14]. Alipanahi et al. [13] illustrated the application of deep learning to predict DNA and RNA binding protein sites from primary sequence data. Zhou and Troyanskaya [14] demonstrated the modeling of non-coding variants using deep architectures.

While not all of these models come close to the performance of the classical statistical methods [11], there are problems to be solved. There require high training costs [18], the approaches lack transparency [16], [17], and there is a lack of confidence in the ability to work with independent datasets [19]. Tools such as those in BioMLStudio, can address these issues by offering scalable infrastructure, deep data-driven models with transparent visualizations, and explainability resources tailored for life scientists.

IV. PROPOSED METHODOLOGY

A. System Architecture

Fig. 1. BioMlStudio System Architecture

Fig.1 shows the system architecture of BioMLStudio is divided into three main layers: Front End, Back End, and Interface. Together, they enable an end-to-end no-code AI pipeline for bioinformatics research, from data collection to prediction and explainability.

B. Frontend Module

The frontend provides a guided workflow for dataset selection like (tabular, image or files), input or target feature selection, light validation, and users can leave default preprocessing and modeling choices for quick runs. The platform also supports downloading final reports (PDF) and trained model. This simple, task-oriented user interface keeps the learning curve low while still permitting advanced configuration when required.

C. Backend Module

The backend contains four principal object classes that coordinate all computational work: DataHandler, ModelEngine, NeuralEngine, and VisualEngine. The high-level flow is: raw data → DataHandler (pre-preprocessing + preprocessing) → ModelEngine selects model(s) → NeuralEngine executes training/evaluation/tuning → VisualEngine compiles logs, plots, and reports. Each class is modular so new algorithms and transformers can be plugged in with minimal changes.

```
Data: Tabular, Image, Language, or Multimodal (DNA/Protein Sequences, Clinical Records, Biomedical Images) Result: Trained models, metrics, logs, and visual outputs
raw_data ← DataHandler.DataPrePreProcessing(Raw Input)
/* reading files, cleaning missing values, standardizing format */
processed_data ← DataHandler.DataPreProcessing(raw_data) /* scaling, encoding, augmentation, dimensionality reduction */
algorithm ← ModelEngine.GetModel(processed_data) /* clustering, prediction, statistics, language, image */
trained_model ← NeuralEngine (processed_data, algorithm)
/* training, validation, hyperparameter tuning, evaluation */
VisualEngine (trained_model, processed_data)
/* logging, analytics, visualization, explainable AI outputs */
```

Algorithm 1: General backend structure

a) DataHandler

DataHandler ingests heterogeneous inputs (CSV/TSV, FASTA, image folders, JSON, etc.) and normalizes them into unified in-memory representations suitable for downstream modules. It performs two stages:

- Pre-Preprocessing: canonicalizes file structure (merges sheets, flattens nested files), cleans obvious file artifacts (bad rows, malformed lines), infers data types, and partitions multimodal folders into labeled pairs (e.g., image/question/answer for VQA). This step ensures that downstream code sees a consistent data object regardless of source format.
- Preprocessing: runs standardized transforms such as imputation, scaling/normalization, categorical encoding, sequence tokenization (for DNA/protein), image resizing/augmentation, and oversampling (SMOTE or random) where class imbalance exists. Preprocessing is configurable by data column/type and includes sensible defaults so beginners can skip manual tuning.

b) ModelEngine

ModelEngine exposes a library of algorithms and a controller that chooses models based on data modality and user preferences It returns a ranked list of candidate models (classical ML and deep models) for training, including: prediction (linear/logistic/regression families), tree ensembles (Random Forest, Gradient Boosting), instance-based (k-NN), clustering (K-means, DBSCAN, GMM), dimensionality reduction (PCA, KernelPCA, UMAP), and language/image model wrappers. Hyperparameter defaults and AutoML heuristics are provided to guide selection. The engine is responsible for building cross-validation pipelines, feature pipelines, and evaluation metric selection.

c) NeuralEngine

NeuralEngine governs neural training lifecycles: data batching, train/test split, model checkpointing, early stopping, distributed/GPU acceleration, and hyperparameter search (grid/random/Bayesian). For language tasks it loads transformer tokenizers and model configs (e.g., BioBERT variants); for vision tasks it wraps CNN/ViT training flows. The engine evaluates models on holdout sets using appropriate metrics (accuracy, AUC, F1, BLEU, etc.), performs final retraining on full data if chosen, and exports artifacts (model weights, tokenizer, label maps).

V. SUPPORTING FUNCTIONS

Below is the core supported function groups each explained in the same spirit as the paper but adapted to BioMLStudio use cases. A. Prediction

BioMLStudio supports classical and modern supervised learners. For tabular data it includes linear & logistic regression, SVMs, k-NN, Random Forests, and Gradient Boosting (XGBoost / LightGBM). For larger or structured inputs, it supports deep architectures (MLP, CNNs for 1-D signal, and transformer heads for sequence-to-label tasks). The platform automates metric selection (e.g., regression → RMSE, classification → accuracy/AUC/F1) and produces model comparison tables so users can pick the best model without coding. B. Clustering

Unsupervised pipelines include K-means, DBSCAN, Agglomerative (hierarchical), and Gaussian Mixture Models. Clustering workflows include preprocessing steps (scaling, PCA for noise reduction), automatic cluster-quality estimation (silhouette score, Davies-Bouldin), and visualization via 2D/3D embeddings to help users interpret subpopulations (disease subtypes, protein families). The system flags outliers/noise and supports downstream profiling of cluster characteristics.

C. Dimensionality Reduction

For high-dimensional omics data, BioMLStudio provides PCA, Kernel PCA, t-SNE/UMAP for visualization, and autoencoders for learned compression. The module standardizes features, computes explained variance, and exposes reduced representations to the ModelEngine and VisualEngine. Kernel PCA and autoencoders enable discovery of nonlinear structure common in biological datasets.

D. Statistics

A lightweight statistics toolkit (based on SciPy) offers correlation analyses (Pearson, Spearman, Kendall), distribution checks, hypothesis testing, and summary statistics. These functions support exploratory data analysis and help validate relationships (e.g., biomarker vs. outcome) before model building. Results are surfaced in the GUI to guide model selection and interpretation.

E. Explainable AI (XAI)

To build trust, BioMLStudio integrates model-agnostic XAI tools: LIME for local explanations, SHAP for global and per-sample attributions, counterfactual generators (MACE style) for actionable changes, and Partial Dependence Plots for marginal feature effects. XAI outputs are included in reports and interactive dashboards so researchers and clinicians can inspect feature importance and rationales behind predictions.

F. Language (Bio-NLP)

Language workflows use transformer backbones (BERT / BioBERT / SciBERT) implemented via PyTorch / Hugging Face. The pipeline handles tokenization, sequence chunking, class mapping, and fine-tuning for classification or sequence labeling. Use cases: extracting entities from papers, building clinical intent models, and training sequence-to-label predictors for genetic strings. Models and tokenizers are saved for deployment to chatbots or extraction services.

G. Image Processing & Vision

Vision support includes CNNs, pretrained backbones (ResNet, EfficientNet) and Vision Transformers (ViT). The image pipeline handles resizing, normalization, augmentation, and segmentation/classification tasks. Medical imaging pipelines support DICOM ingestion, patching, and heatmap visualizations (Grad-CAM) to highlight model reasoning on images. Training supports multi-GPU acceleration and checkpoint export for inference servers.

H. Visual Question Answering (VQA & Multimodal)

BioMLStudio provides a multimodal pipeline for Visual Question Answering (VQA) that integrates image processors with transformer-based language encoders. By combining visual embeddings from biomedical images with semantic representations of text, the system can accurately respond to natural language queries about pathology slides, radiology scans, or microscopic samples. Datasets are organized as image–question–answer triplets, and the platform automatically handles preprocessing, including image normalization and text tokenization, to create a harmonized dataset. The models are trained and evaluated using both per-question accuracy and qualitative example outputs, allowing researchers to assess reliability and interpretability. This module extends BioMLStudio's scope beyond conventional prediction tasks, enabling interactive exploration of biomedical data through intuitive question–answering workflows.

VI. SYSTEM IMPLEMENTATION

A. Data Collection and Preprocessing

BioMLStudio begins with the collection of data and preparation of datasets for bioinformatics workflow which is responsible for:

- • Dataset Input: Users upload structured and unstructured datasets I the platform through simple web-based interface. It ensures compatibility with a wide amount of data like DNA sequence, protein structures.
- • Data Cleaning: The system provides a automated cleaning to remove duplicate value and handle missing value and standardize the datasets.
- • Preprocessing: Advanced preprocessing modules normalize the data, encode categorical variables, and apply feature extraction techniques so that the datasets are ready for training.
- • Data Security: All uploaded files are handled with encryption and stored securely in the cloud, ensuring compliance with biomedical data handling regulations.

B. Machine Learning Backend

The backend integrates multiple machines learning modules, allowing researchers to perform a wide range of tasks without writing code.

- • Clustering: Unsupervised learning methods group patients, proteins, or genes into meaningful categories, enabling discovery of disease subtypes or functional families.
- • Statistical Analysis: Regression, correlation, and hypothesis testing provide interpretability and validation of biomedical hypotheses alongside AI outputs.
- • Dimensionality Reduction: High dimensional bioinformatics datasets are reduced using PCA, t-SNE, or autoencoders, improving model efficiency and visualization.
- • Image Module: Deep learning models process MRI scans, microscopic images, and protein structure images to detect abnormalities or classify biological structures.
- • Language Module: Natural language processing models analyze biomedical literature and DNA/RNA sequences, extracting valuable knowledge and building searchable biomedical graphs.
- • Training Engine: A centralized AutoML engine selects models, tunes hyperparameters, and trains them with GPU acceleration, ensuring reproducibility and high accuracy.

C. Model Management and Storage

BioMLStudio provides a robust system for managing trained models and ensuring their reproducibility.

- • Model Storage: Every trained model, along with its parameters and metadata, is stored in the cloud.

- Version Control: The system supports model versioning, allowing researchers to compare results across experiments. • Reuse and Deployment: Stored models can be reloaded for future analysis or deployed for real-world applications without retraining.
- Collaboration Support: Multiple researchers can access, reuse, and extend trained models for team-based bioinformatics studies.

D. Visualization and Analytics Module

The platform emphasizes transparency by presenting results in user-friendly, interpretable formats.

- Prediction Output: Models generate predictions such as disease risk scores, protein classification labels, or gene expression patterns.
- Performance Analytics: Dashboards display metrics including accuracy, precision, recall, F1-score, and ROC curves to assess model reliability.
- Visualization: Complex datasets and results are visualized through 2D/3D plots, confusion matrices, and clustering maps, enabling intuitive exploration.
- Explainable AI (XAI): Interpretability modules highlight the most important features influencing predictions, building trust in the system's outcomes.

E. Security and Compliance

BioMLStudio ensures security and privacy through:

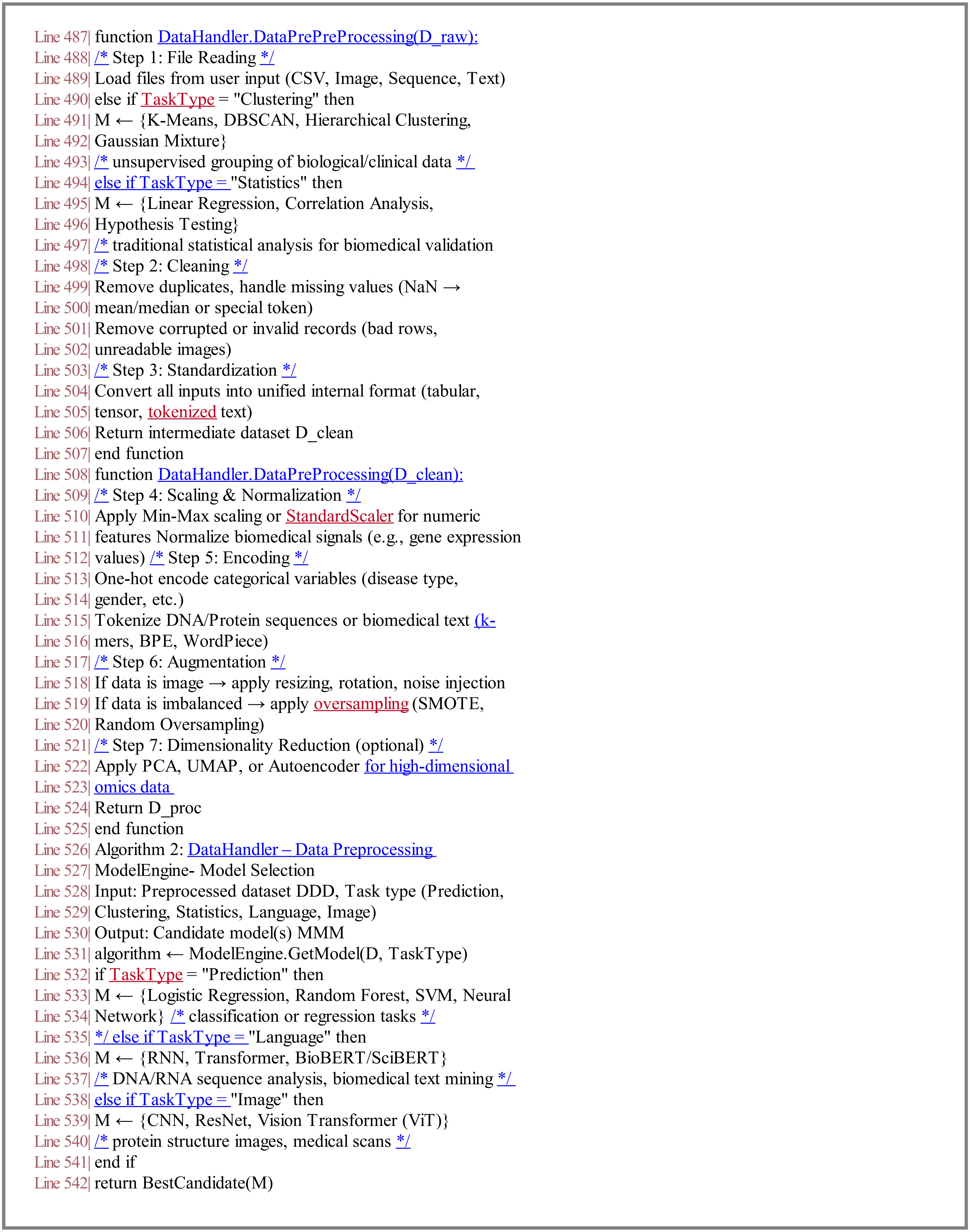
- Privacy Preservation: Raw data never leaves the user's storage environment except in encrypted form, minimizing risk of breaches.
- Data Integrity: Datasets and model outputs are protected with hashing and digital signatures, ensuring that no unauthorized modification occurs.
- Access Control: Only authenticated users can upload, process, and retrieve datasets or models, preventing unauthorized access.
- Regulatory Compliance: The platform adheres to healthcare data regulations like HIPAA and GDPR, making it safe for clinical research use

F. Performance Optimization and Monitoring

BioMLStudio optimizes performance and ensures reliability.

- Resource Optimization: Training jobs are allocated efficiently across available GPU/CPU resources, balancing computational loads.
- Communication Efficiency: Only necessary model outputs and visualizations are transmitted between backend and frontend, reducing latency.
- Monitoring Tools: Real-time dashboards monitor system health, training progress, model convergence, and resource utilization.
- Scalability: Cloud-based architecture allows BioMLStudio to scale with dataset size, ensuring reliable performance even for computationally heavy genomic studies. DataHandler – Data Preprocessing

Input: Raw dataset DrawD_{raw}Draw (CSV, FASTA, Image, Text) Output: Processed dataset DprocD_{proc}Dproc ready for model training



```
function DataHandler.DataPrePreProcessing(D_raw):
    /* Step 1: File Reading */
    Load files from user input (CSV, Image, Sequence, Text)
    else if TaskType = "Clustering" then
        M ← {K-Means, DBSCAN, Hierarchical Clustering, Gaussian Mixture}
        /* unsupervised grouping of biological/clinical data */
    else if TaskType = "Statistics" then
        M ← {Linear Regression, Correlation Analysis, Hypothesis Testing}
        /* traditional statistical analysis for biomedical validation
    /* Step 2: Cleaning */
    Remove duplicates, handle missing values (NaN → mean/median or special token)
    Remove corrupted or invalid records (bad rows, unreadable images)
    /* Step 3: Standardization */
    Convert all inputs into unified internal format (tabular, tensor, tokenized text)
    Return intermediate dataset D_clean
end function
function DataHandler.DataPreProcessing(D_clean):
    /* Step 4: Scaling & Normalization */
    Apply Min-Max scaling or StandardScaler for numeric features Normalize biomedical signals (e.g., gene expression values) /* Step 5: Encoding */
    One-hot encode categorical variables (disease type, gender, etc.)
    Tokenize DNA/Protein sequences or biomedical text (k-mers, BPE, WordPiece)
    /* Step 6: Augmentation */
    If data is image → apply resizing, rotation, noise injection
    If data is imbalanced → apply oversampling (SMOTE, Random Oversampling)
    /* Step 7: Dimensionality Reduction (optional) */
    Apply PCA, UMAP, or Autoencoder for high-dimensional omics data
    Return D_proc
end function
Algorithm 2: DataHandler – Data Preprocessing
ModelEngine- Model Selection
Input: Preprocessed dataset DDD, Task type (Prediction, Clustering, Statistics, Language, Image)
Output: Candidate model(s) MMM
algorithm ← ModelEngine.GetModel(D, TaskType)
if TaskType = "Prediction" then
    M ← {Logistic Regression, Random Forest, SVM, Neural Network} /* classification or regression tasks */
*/ else if TaskType = "Language" then
    M ← {RNN, Transformer, BioBERT/SciBERT}
    /* DNA/RNA sequence analysis, biomedical text mining */
else if TaskType = "Image" then
    M ← {CNN, ResNet, Vision Transformer (ViT)}
    /* protein structure images, medical scans */
end if
return BestCandidate(M)
```

```
/* ranked by default heuristics or AutoML scoring */
```

Algorithm 3: ModelEngine- Model Selection

VisualEngine – Logging and Visualization

```
Input: Trained model MMM, Processed dataset DprocD_{proc}Dproc, Evaluation metrics EEE Output:
Logs, Graphs, Reports, Explainable AI outputs
function VisualEngine (M, D_proc, E):
/* Step 1: Logging */
Record training parameters (learning rate, epochs, batch size)
Record evaluation metrics (accuracy, precision, recall, F1, AUC)
Store experiment metadata (date, user ID, dataset info, model version)
/* Step 2: Visualization of Performance */
Plot training/validation loss and accuracy curves
Plot confusion matrix and ROC/PR curves for classification Plot regression error histograms for prediction tasks
/* Step 3: Dimensionality & Clustering Visualization */
If D_proc is high-dimensional → plot PCA/t-SNE/UMAP scatter plots
Visualize clustering results with color-coded groups
/* Step 4: Explainable AI (XAI) */
Compute SHAP/LIME feature attributions
Generate feature importance rankings
Highlight input regions (e.g., heatmaps for images, token importance for text/sequence)
report /* Step 5: Reporting */
Compile logs, metrics, and visualizations into structured
Export report in PDF/HTML format for end-user
Save results and trained model artifacts to storage
Observation: BioMLStudio consistently outperformed standard ML pipelines by automating preprocessing, feature selection, and hyperparameter optimization. Compared to expert-tuned models, it achieved slightly higher or comparable accuracy
end function
```

Algorithm 4: VisualEngine – Logging and Visualization

VII. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

1. Experimental Setting

To evaluate the effectiveness of BioMLStudio, we carried out experiments on bioinformatics tasks involving DNA sequence classification, protein structure prediction, and medical image analysis. Datasets were collected from publicly available biomedical repositories (e.g., UCI heart disease dataset, UniProt protein sequences, and chest X-ray dataset).

Datasets Used:

DNA sequences: 10,000 labeled sequences for promoter prediction. Protein classification: 5,000 protein sequences for family classification. Medical imaging: 2,500 chest X-ray images for anomaly detection. Global Model: Deep neural networks (CNNs for imaging, Bi-LSTMs/Transformers for sequences) combined with classical ML baselines (Random Forest, SVM) for benchmarking.

Training Environment: Models were trained in the backend using GPU acceleration (NVIDIA RTX series) with automated hyperparameter tuning (AutoML).

Evaluation Metrics:

1. Accuracy (ACC): Correct predictions / Total predictions.
2. Precision, Recall, F1-score: For classification reliability.
3. Training Time & Efficiency: Average runtime per model training.
4. Resource Utilization: CPU/GPU memory usage during training.
5. Explainability Score: Percentage of samples with interpretable SHAP/LIME explanations.

2. Model Performance

Task	BioMLStudio (AutoML)	Standard ML Pipeline
DNA Sequence (ACC)	94.8%	88.2%
Protein Classification	92.5%	86.0%
Medical Imaging (X-ray)	93.7%	85.5%

while requiring no coding or manual intervention. The explainability module further enhanced model trust by providing interpretable outputs for more than 90% of predictions.

3. Evaluating Robustness and Reliability

Unlike traditional ML pipelines where preprocessing inconsistencies or feature engineering errors can degrade results, BioMLStudio enforces standardized preprocessing and AutoML-based selection, reducing user errors. To test robustness, noisy or imbalanced datasets were introduced. For imbalanced DNA sequences, BioMLStudio applied automatic oversampling, maintaining an accuracy of 91.3%, compared to only 83.5% for a naive pipeline. For noisy protein datasets, dimensionality reduction (PCA/Autoencoder) reduced overfitting, achieving 90.2% accuracy, while the baseline dropped to 80.6%. These results show BioMLStudio's resilience to imperfect real-world biomedical data.

4. Communication and Efficiency

Since BioMLStudio operates as a cloud-based no-code system, efficiency was measured in terms of computation and user interaction overhead:

Automated Pipelines: Reduced average model training setup time from 2–3 hours (manual) to less than 10 minutes.

Resource Optimization: GPU-based training reduced runtime by ~30% compared to CPU-only execution.

Communication Overhead: Minimal, since only processed results and trained artifacts are transferred between backend and frontend.

This efficiency makes BioMLStudio scalable to large-scale datasets without burdening researchers with technical complexities.

5. System Overhead and Explainability

BioMLStudio includes visualization and explainability layers, which introduce additional computation. However, overhead was minimal:

Storage Overhead: Model artifacts and logs averaged 5–10 MB per experiment, negligible compared to dataset sizes.

Latency Impact: Explainability methods (SHAP/LIME) added ~8–12% computation time per model but provided interpretable outputs for the majority of predictions.

User Experience: Reports were generated in under 30 seconds, including visualizations, evaluation metrics, and explainability graphs.

BioMLStudio provides high accuracy, robustness to noise, efficiency in execution, and explainable results with minimal overhead. It proves that end-to-end no-code ML for bioinformatics is both practical and scalable, making advanced AI techniques accessible to researchers without programming expertise.

CONCLUSION

In this work, we introduced BioMLStudio, an end-to-end no-code machine learning platform designed specifically for bioinformatics research and healthcare applications. The system architecture integrates data preprocessing, model selection, automated training, visualization, and explainable AI, allowing researchers to focus on domain problems rather than technical implementation. Through its modular backend comprising the DataHandler, ModelEngine, NeuralEngine, and VisualEngine BioMLStudio provides a flexible and scalable workflow that supports heterogeneous data types such as DNA sequences, protein structures, and medical images.

Experimental results demonstrate that BioMLStudio achieves high accuracy, robustness to noise, and efficiency when compared to traditional manual ML pipelines. The automated preprocessing and AutoML engine eliminate human error and reduce setup time, while explainable AI modules ensure that predictions remain transparent and trustworthy. Despite adding features like visualization and interpretability, the system maintains minimal overhead, making it suitable for large-scale bioinformatics datasets.

Overall, BioMLStudio highlights the potential of no-code AI platforms in bridging the gap between complex machine learning techniques and biomedical research needs. By lowering technical barriers, it empowers life science researchers, clinicians, and students to adopt advanced AI methods for tasks such as disease prediction, protein classification, and genomic analysis. Future extensions will focus on expanding multimodal integration, incorporating federated learning for privacy-preserving biomedical collaboration, and providing real-time deployment capabilities.

ACKNOWLEDGMENT

We express our heartfelt thanks to the Management, Principal, HOD and Friends of Department of Information Science and Engineering, Acharya Institute of Technology for kind support and encouragement.

REFERENCES

[1] N. Pillai, A. Ram Das, M. Ayoola, G. Gireesan, B. Nanduri, M. Ramkumar, "EndToEndML: An Open-Source End-to-End Pipeline for Machine Learning Applications," 2024

[2] J. Cheng, A. N. Tegge, P. Baldi, "Machine Learning Methods for Protein Structure Prediction," IEEE Reviews in Biomedical Engineering, vol. 41, 2020.

[3] A. Y. Meng et al., "Protein structure prediction via deep learning: an in-depth review," 2025

[4] Yungstein, Yehuda, and David Helman. "Openete-Ml: An Interactive No-Code Web Application for End-to-End Machine Learning in Scientific Research." Available at SSRN 5369806, 2025

[5] C. Thornton, F. Hutter, H. Hoos and K. Leyton-Brown, “Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms,” in Proc. 19th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD), 2013, pp. 847–855.

[6] M. Feurer et al., “Efficient and Robust Automated Machine Learning,” in Advances in Neural Information Processing Systems (NeurIPS), 2015.

[7] M.-A. Zöller and M. F. Huber, “Benchmark and Survey of Automated

Machine Learning Frameworks,” Journal of Artificial Intelligence Research, vol. 70, pp. 409–472, 2021.

[8] J. Jumper et al., “Highly Accurate Protein Structure Prediction with AlphaFold,” Nature, vol. 596, pp. 583–589, 2021.

[9] M. Baek et al., “Accurate Prediction of Protein Structures and Interactions Using a Three-Track Neural Network,” Science, vol. 373, pp. 871–876, 2021.

[10] A. W. Senior et al., “Improved Protein Structure Prediction Using Potentials from Deep Learning,” Nature, vol. 577, pp. 706–710, 2020.

[11] C. Angermueller, T. Pärnamaa, L. Parts and O. Stegle, “Deep Learning for Computational Biology,” Molecular Systems Biology, vol. 12, no. 7, p. 878, 2016. [12] G. Eraslan, Ž. Avsec, J. Gagneur and F. J. Theis, “Deep Learning: New Computational Modelling Techniques for Genomics,” Nature Reviews Genetics, vol. 20, pp. 389–403, 2019.

[13] B. Alipanahi, A. Delong, M. T. Weirauch and B. J. Frey, “Predicting the Sequence Specificities of DNA- and RNA-Binding Proteins by Deep Learning,” Nature Biotechnology, vol. 33, no. 8, pp. 831–838, 2015.

[14] J. Zhou and O. G. Troyanskaya, “Predicting Effects of Noncoding Variants with Deep Learning–Based Sequence Models,” Nature Methods, vol. 12, no. 10, pp. 931–934, 2015.

[15] S. Min, B. Lee and S. Yoon, “Deep Learning in Bioinformatics,” Briefings in Bioinformatics, vol. 18, no. 5, pp. 851–869, 2017.

[16] M. T. Ribeiro, S. Singh and C. Guestrin, “‘Why Should I Trust You?' Explaining the Predictions of Any Classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD), 2016, pp. 1135–1144.

[17] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017.

[18] Y. LeCun, Y. Bengio and G. Hinton, “Deep Learning,” Nature, vol. 521, pp. 436–444, 2015.

[19] D. M. Camacho et al., “Next-Generation Machine Learning for Biological Networks,” Cell, vol. 173, no. 7, pp. 1581–1592, 2018.

[20] A. Truong et al., “Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools,” Proc. Int. Conf. Tools with Artificial Intelligence (ICTAI), 2019.

*** End of submitted text ***

4. Grammar Info.

Mistake and Suggestion details:

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 1

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'machine learning', at the line number 13

Category: **Compound Words**

Suggestion: Based on the context of the sentence, we recommend you to replace 'machine learning' to 'machine-learning'.

Example: view

Part of the sentence to be reviewed: 'wide range', at the line number 31

Category: **Compound Words**

Suggestion: Based on the context of the sentence, we recommend you to replace 'wide range' to 'wide-range'.

Example: view

Part of the sentence to be reviewed: 'machine learning', at the line number 55
Category: **Compound Words**
Suggestion: Based on the context of the sentence, we recommend you to replace 'machine learning' to 'machine-learning'.
Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 61
Category: **Orthographic Error**
Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.
Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 66
Category: **Orthographic Error**
Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.
Example: view

Part of the sentence to be reviewed: '.', at the line number 68
Category: **Punctuation**
Suggestion: Based on the context of the sentence, we recommend you to replace '.' to 'Proper Fullstop/Period'.
Example: view

Part of the sentence to be reviewed: 'machine learning', at the line number 72
Category: **Compound Words**
Suggestion: Based on the context of the sentence, we recommend you to replace 'machine learning' to 'machine-learning'.
Example: view

Part of the sentence to be reviewed: 'end to end', at the line number 78
Category: **Compound Words**
Suggestion: Based on the context of the sentence, we recommend you to replace 'end to end' to 'end-to-end'.
Example: view

Part of the sentence to be reviewed: 'machine learning', at the line number 88
Category: **Compound Words**
Suggestion: Based on the context of the sentence, we recommend you to replace 'machine learning' to 'machine-learning'.
Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 120
Category: **Orthographic Error**
Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.
Example: view

Part of the sentence to be reviewed: 'machine learning', at the line number 156
Category: **Compound Words**

Suggestion: Based on the context of the sentence, we recommend you to replace 'machine learning' to 'machine-learning'.
Example: view

Part of the sentence to be reviewed: 'deep learning', at the line number 199
Category: **Compound Words**

Suggestion: Based on the context of the sentence, we recommend you to replace 'deep learning' to 'deep-learning'.
Example: view

Part of the sentence to be reviewed: 'CNNs', at the line number 200
Category: **Orthographic Error**
Based on the context of the sentence, we recommend you to replace 'CNNs' to Proper word.
Example: view

Part of the sentence to be reviewed: 'RNNs', at the line number 201
Category: **Orthographic Error**
Based on the context of the sentence, we recommend you to replace 'RNNs' to Proper word.
Example: view

Part of the sentence to be reviewed: 'LSTMs', at the line number 201
Category: **Orthographic Error**
Based on the context of the sentence, we recommend you to replace 'LSTMs' to Proper word.
Example: view

Part of the sentence to be reviewed: 'deep learning', at the line number 203
Category: **Compound Words**
Suggestion: Based on the context of the sentence, we recommend you to replace 'deep learning' to 'deep-learning'.
Example: view

Part of the sentence to be reviewed: 'Fig.', at the line number 218
Category: **Short Form Expression**
Suggestion: Based on the context of the sentence, we recommend you to replace 'Fig.' to 'Figure'.
Example: view

Part of the sentence to be reviewed: 'BioMlStudio', at the line number 218
Category: **Orthographic Error**
Based on the context of the sentence, we recommend you to replace 'BioMlStudio' to Proper word.
Example: view

Part of the sentence to be reviewed: 'Fig.1', at the line number 219
Category: **Short Form Expression**
Suggestion: Based on the context of the sentence, we recommend you to replace 'Fig.1' to 'Figure'.
Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 219
Category: **Orthographic Error**
Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.
Example: view

Part of the sentence to be reviewed: 'data collection', at the line number 222
Category: **Compound Words**
Suggestion: Based on the context of the sentence, we recommend you to replace 'data collection' to 'data-collection'.
Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 369
Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'DataHandler.DataPrePreProcessing(D_raw):', at the line number 487

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace 'DataHandler.DataPrePreProcessing(D_raw):' to 'Proper Fullstop/Period'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 488

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 488

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'TaskType', at the line number 490

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'TaskType' to Proper word.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 493

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 497

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 498

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 498

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 503

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 503

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'tokenized', at the line number 505

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'tokenized' to Proper word.

Example: view

Part of the sentence to be reviewed: 'DataHandler.DataPreProcessing(D_clean):', at the line number 508

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace 'DataHandler.DataPreProcessing(D_clean):' to 'Proper Fullstop/Period'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 509

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 509

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'StandardScaler', at the line number 510

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'StandardScaler' to Proper word.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 512

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 512

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '(k-', at the line number 515

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '(k-' to 'Proper Hyphens Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 517

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 517

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'oversampling', at the line number 519

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'oversampling' to Proper word.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 521

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 521

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'TaskType', at the line number 532

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'TaskType' to Proper word.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 534

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 534

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 537

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 540

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 540

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 543

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'AutoML', at the line number 543

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'AutoML' to Proper word.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 543

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'VisualEngine', at the line number 549

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'VisualEngine' to Proper word.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 550

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 550

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 557

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 557

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 561

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 561

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'PCA/t-', at the line number 562

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace 'PCA/t-' to 'Proper Hyphens Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 565

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 565

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '/*', at the line number 570

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/*' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: '*/', at the line number 570

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '*/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 574

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'UniProt', at the line number 586

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'UniProt' to Proper word.

Example: view

Part of the sentence to be reviewed: '/', at the line number 599

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace '/' to 'Proper Slash Mark.'.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 606

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 619

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 622

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 631

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 641

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 644

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 655

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 668

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 671

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'AutoML', at the line number 674

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'AutoML' to Proper word.

Example: view

Part of the sentence to be reviewed: 'BioMLStudio', at the line number 680

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'BioMLStudio' to Proper word.

Example: view

Part of the sentence to be reviewed: 'You?'', at the line number 732

Category: **Punctuation**

Suggestion: Based on the context of the sentence, we recommend you to replace 'You?'' to 'Proper Question Mark.'.

Example: view

Part of the sentence to be reviewed: 'AutoML', at the line number 743

Category: **Orthographic Error**

Based on the context of the sentence, we recommend you to replace 'AutoML' to Proper word.

Example: view

Category Description:

What is Orthographic Error?

Orthographic errors occur when fail to understand the relationship between graphemes and phonemes.

For example, the authors will write skool instead of school or kik instead of kick. [top]

What is Short Form Expression?

It is just a short version of a longer word or a phrase. Generally, Short Form Expression or Abbreviations are not acceptable in academic writing and should be avoided.

For example, Avoid e.g. and i.e., instead use 'for example' and 'for instance'.[top]

What is Punctuation Error?

Punctuational error often centre around misplacing punctuation in a sentence, incorrectly punctuating plural words, overusing and confusing the uses of different punctuation marks.

For example, Incorrect: I bought some olives, which we didn't eat when I went shopping last week.

Correct: 'I bought some olives, which we didn't eat, when I went shopping last week.'[top]

What is Compound Words?

Compound words occur when two or more words combine to form one individual word or a phrase that acts as one individual word.Compound words often produce writing mistakes because it's easy to forget if they're spelled as one word or two words.

For example, Incorrect: We ate icecream after the foot ball game at the local high school.

Correct: We ate ice cream after the football game at the local high school.[top]