ACTA SCIENTIAE

ISSN: 2178-7727

A Hybrid Approach to Student Assessment: Integrating C Code Metrics with Deep Learning Models

[1]Mrs. Urmila Kadam, [2]Dr. Kavita Oza

*1Assistant Professor, Dr. D. Y. Patil School of MCA, Lohegaon Pune, Maharashtra, India 412 105
*2Associate Professor, Department of Computer Science, Shivaji University, Kolhapur, Maharashtra, India 416 004
Corresponding author email- kadamurmila777@gmail.com, kso_csd@unishivaji.ac.in

ABSTRACT

Traditional methods of assessing programming skills primarily rely on output accuracy and manual grading, which can be time-consuming, subjective, and limited in scope. This paper introduces a hybrid, automated approach for evaluating students' programming capabilities by analyzing the structural and behavioral aspects of their C code submissions. The proposed method involves extracting a rich set of static features from the C programs written by students during assessments. These features include lines of code, presence of comments, number and types of variables, condition checking count, loop count, retention of input, well-labeled output, return statement usage, variable initialization, shorthand notation, and detection of syntax errors. These attributes are indicative of a student's coding style, logic formulation, and understanding of programming constructs. To predict the academic outcomes, an Artificial Neural Network (ANN) is employed, which takes the extracted features as input and outputs the predicted grade and percentage of the student. The model is trained and validated on a dataset of real student code submissions, demonstrating high accuracy and consistency in performance prediction. This approach not only reduces the burden on educators but also ensures fairness, objectivity, and scalability in grading. Furthermore, it provides detailed feedback on specific aspects of student coding behavior, helping instructors identify strengths and areas for improvement. By combining code analytics with deep learning, this system establishes a foundation for intelligent education systems that adapt to student learning patterns and performance. It paves the way for data-driven academic decision-making and personalized feedback mechanisms in programming education. The results affirm that integrating code metrics with neural networks can revolutionize how programming assessments are conducted in academic settings.

KEYWORDS: C code metrics, student assessment, deep learning, artificial neural network, feature extraction, programming evaluation, educational data mining, grade prediction, percentage prediction, code analysis

INTRODUCTION

In the realm of computer science education, programming is a foundational skill that shapes a student's ability to reason computationally and solve real-world problems through algorithmic thinking. Evaluating students' programming abilities has traditionally relied on assessing the correctness of program output, adherence to functional requirements, or manual review of code structure and style. However, these conventional assessment methods, while widely adopted, present several limitations. Manual grading is labor-intensive, subjective, and prone to inconsistencies across evaluators. Additionally, focusing solely on the output neglects crucial aspects of the learning process such as logical structuring, use of proper syntax, code modularity, and adherence to best programming practices. With the expansion of student populations and the growing emphasis on outcome-based education, there is an urgent need for more objective, scalable, and intelligent assessment methodologies. The evolution of Artificial Intelligence (AI) and deep learning has introduced promising avenues for automating and enhancing the educational assessment landscape. In particular, the integration of AI into programming evaluation enables systems to interpret the underlying logic of student code, analyze behavioral patterns, and provide rich, data-driven insights into performance and understanding. This research presents a hybrid student assessment framework that combines static code analysis with a deep learning model—specifically, an Artificial Neural Network (ANN)—to predict academic outcomes from student-written C programs. The core premise of this work is that the structural and semantic features of a student's code offer significant indicators of their comprehension and skill level. To explore this, a set of carefully selected code metrics is extracted from each student's C code submission. These metrics include, but are not limited to:

- Lines of code
- Use of comments
- Number and types of variables
- Condition checking and loop constructs
- Retention of input
- Well-labeled output
- Return statement usage
- Initialization practices
- Use of shorthand notation
- Presence of syntax errors

These features capture both the qualitative and quantitative characteristics of code, reflecting a student's logical flow, design patterns, and adherence to programming norms.

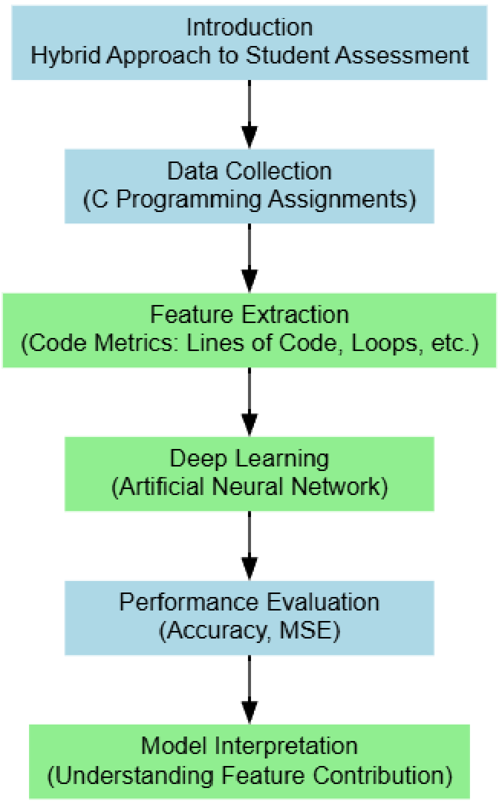

Fig 1 Concept of Project

Once extracted, these features are passed into a trained ANN model which predicts two critical academic indicators: the percentage score and the final grade. This transformation from raw code to structured features enables the model to learn meaningful correlations between coding behavior and academic performance, thus offering a holistic view of student competency.

The significance of this approach lies in its ability to provide:

1. Automated and scalable evaluation for large student cohorts.
2. Objective and consistent grading, free from human biases.
3. Deep feedback and diagnostics on student code structure and design.

4. Data-driven decision support for educators and institutions.
5. Personalized insights into each student's strengths and areas for improvement.

The hybrid model proposed in this paper goes beyond mere automation—it embodies an intelligent, behavior-aware grading system that aligns with the goals of modern education systems and intelligent tutoring platforms. Furthermore, this methodology contributes to the field of Educational Data Mining (EDM) and Learning Analytics, offering new pathways for research into how students learn programming and how their development can be quantitatively modelled and supported. The results of our experimentation indicate that ANN models trained on code-derived features can accurately predict student performance, validating the hypothesis that source code contains rich, latent information reflective of cognitive understanding. This model has practical applications in online learning systems, coding bootcamps, virtual classrooms, and academic institutions looking to enhance the fairness, speed, and depth of programming assessments. In summary, this research addresses the need for a smarter assessment paradigm in programming education. By integrating static code metrics with deep learning models, it introduces an innovative, scalable, and insightful approach that transforms how programming skills are evaluated in academic settings.

PROBLEM STATEMENT:

The traditional assessment of programming assignments predominantly focuses on output validation and manual code reviews. While effective to an extent, these methods are time-consuming, prone to human error, and unable to scale efficiently across large student populations. Furthermore, they fail to capture deeper cognitive and structural aspects of a student's code, such as coding style, logical design, syntax usage, and programming habits. As a result, students who demonstrate conceptual understanding but struggle with minor syntax or output issues may receive unfair evaluations. Conversely, those who achieve correct outputs through inefficient or poor coding practices might be overestimated in their performance. With the increasing adoption of online learning platforms and automated code submission systems, there is an urgent need for intelligent, consistent, and scalable evaluation mechanisms that can analyze student code not just functionally but structurally and behaviorally. Existing auto-graders often focus solely on input-output matching, offering limited insights into the quality and clarity of a student's logic or adherence to programming standards. The core problem, therefore, lies in the absence of an effective framework that integrates code behavior analysis with intelligent performance prediction models. There is a need for a system that can extract meaningful code metrics and predict academic outcomes like grades and percentages, while also offering diagnostic feedback. This research aims to address this gap by developing a hybrid evaluation model that combines static C code feature extraction with deep learning-based performance prediction, specifically using an Artificial Neural Network (ANN). The proposed model not only automates assessment but also brings transparency, objectivity, and pedagogical value to programming evaluation.

OBJECTIVES:

The primary objective of this research is to develop an intelligent, hybrid assessment model that accurately predicts student academic performance based on features extracted from their C programming code. This objective is rooted in the need to improve the accuracy, scalability, and objectivity of programming assessments using AI-driven techniques.

To fulfil this aim, the specific objectives of the study are outlined as follows:

- ✓ **To design and implement a code feature extraction module** that automatically analyzes student-submitted C code to derive meaningful structural and semantic metrics, such as lines of code, loop counts, condition checks, use of comments, variable types, shorthand notations, syntax correctness, etc.
- ✓ **To develop a comprehensive feature set** that captures both qualitative and quantitative aspects of programming style, logic, and structure, enabling deeper insight into students' coding behavior and understanding.
- ✓ **To build and train an Artificial Neural Network (ANN)** that can effectively learn patterns from the extracted code features and predict two key academic indicators: **percentage** and **grade**.
- ✓ **To evaluate the performance of the ANN model** using appropriate metrics (e.g., accuracy, mean

ISSN: 2178-7727

squared error, confusion matrix) to validate the effectiveness of code-based predictive assessment.

- ✓ **To compare the ANN-based predictions with traditional evaluation methods** in order to demonstrate improvements in grading accuracy, fairness, and efficiency.
- ✓ **To provide a scalable, automated, and educator-friendly assessment solution** that can be integrated into existing academic systems and online coding platforms for real-time performance analysis and feedback.

METHODOLOGY

This research adopts a hybrid approach to programming assessment, seamlessly integrating static code analysis with deep learning-based performance prediction. By combining these two powerful techniques, the study aims to provide a more comprehensive and accurate evaluation of student programming skills. The methodology is organized into several distinct phases, each designed to contribute to the overall performance prediction process. The first phase, data collection, involves gathering C programming assignment submissions from students. These submissions serve as the primary source of raw data, offering valuable insights into the structure and content of student code.

In the feature extraction phase, key metrics are derived from the code to quantify various aspects of programming proficiency. Features such as lines of code, use of comments, variable types, loop count, and syntax errors are extracted, forming the foundation for performance prediction. the next phase, model development, involves the application of deep learning techniques, specifically an artificial neural network (ANN), to predict student performance. The ANN is trained using the extracted features as inputs, with the target variables being student grades and percentages. The model is optimized to achieve the highest accuracy and minimize errors. Following model development, the evaluation phase assesses the model's performance using appropriate metrics, such as accuracy, mean squared error (MSE), and cross-validation techniques, to ensure robustness and generalizability. finally, the result interpretation phase involves analyzing the model's predictions to identify key patterns and feature contributions. This phase provides valuable insights into the aspects of code that most strongly correlate with student performance, offering a deeper understanding of the programming assessment process.

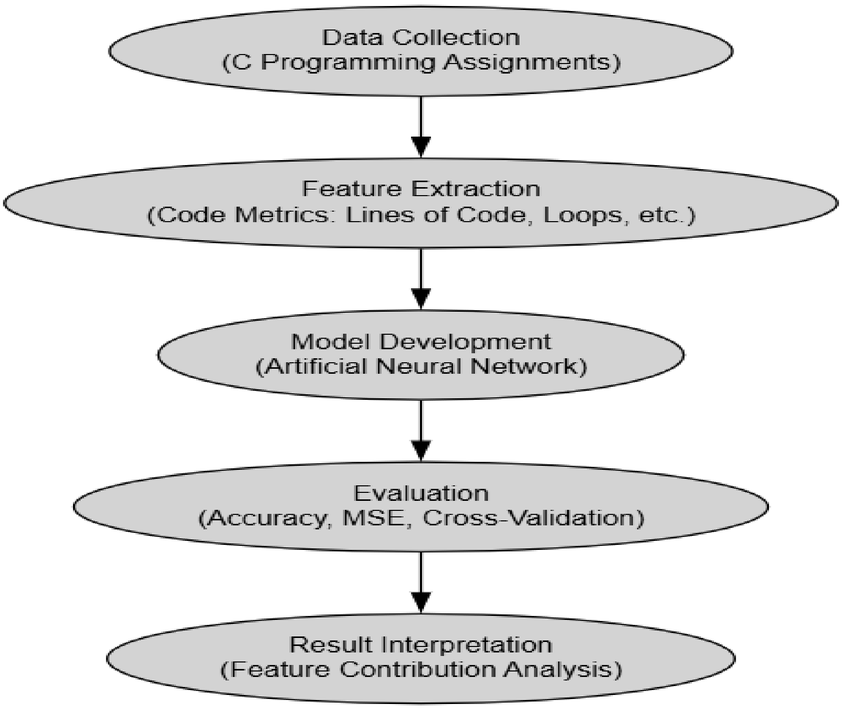

Fig 2 Methodology

Data Collection

The dataset for this study consists of C programming assignments submitted by students as part of their

ISSN: 2178-7727

academic coursework. The dataset includes source code files that contain different levels of complexity and cover a variety of topics in C programming, such as basic syntax, control structures, functions, and arrays. Each submission is paired with two target variables: Grade (categorical) and Percentage (continuous), which represent the final assessment of the student's performance in the course.

To ensure a comprehensive evaluation, the dataset includes submissions from students with varied programming skills, ranging from beginner to advanced levels.

Feature Extraction:
The first step in processing the collected data involves extracting a set of static features from the C code. This is done using a custom-built feature extraction module that analyzes the code structure, logic, and style.

The following features are extracted from each submission:

- Lines of Code (LOC): The total number of lines written in the code.
- Use of Comments: A binary feature indicating the presence of comments in the code.
- Number and Types of Variables: The total count of variables and their data types (e.g., integer, float, char).
- Condition Checking Count: The number of conditional statements (e.g., if, else, switch) used in the code.
- Loop Count: The number of loops (e.g., for, while, do-while) used.
- Retention of Input: Whether the code retains and properly handles input values.
- Well-Labeled Output: Whether the code provides clear, understandable output with meaningful labels.
- Return Statement Usage: A binary feature indicating whether the code correctly includes a return statement in functions.
- Variable Initialization: A binary feature indicating whether all variables are initialized before use.
- Use of Shorthand Notation: The presence of shorthand coding techniques (e.g., i++, a += b).
- Syntax Errors: The number of syntax errors or potential issues detected in the code.

These features are captured from each submission using a combination of lexical analysis (for parsing and tokenizing the code) and static analysis tools (to detect specific patterns such as loops, conditions, and variable declarations).

Model Development
The extracted features are then fed into a deep learning model, specifically an Artificial Neural Network (ANN), to predict the student's grade and percentage. The ANN is designed to handle both regression (for predicting percentage) and classification (for predicting grade) tasks. The model consists of multiple hidden layers with ReLU activation functions to capture complex non-linear relationships between the input features and the target variables. The output layer consists of two parts: one for predicting the percentage (continuous output) and another for predicting the grade (categorical output). The model is trained using the backpropagation algorithm and optimized with the Adam optimizer to minimize the loss function. The training dataset is split into a training set (80%) and a validation set (20%) to ensure generalizability.

Evaluation
The performance of the ANN model is evaluated using various metrics:

- **Accuracy** (for grade classification): Measures how well the predicted grades match the actual grades.
- **Mean Squared Error (MSE)** (for percentage prediction): Quantifies the difference between predicted and actual percentage scores.
- **Confusion Matrix**: Helps assess the classification performance by showing true positives, false positives, true negatives, and false negatives.
- **Cross-Validation**: 10-fold cross-validation is used to assess the robustness of the model and prevent overfitting.

ISSN: 2178-7727

DATA COLLECTION:

The data collection phase forms the foundation of this study, wherein C programming assignments are gathered from students to serve as the primary dataset. These assignments include a variety of programming tasks that reflect students' coding abilities, problem-solving approaches, and understanding of programming concepts. The data collected from each assignment consists of the students' C code submissions, which are systematically stored in a database for further analysis. Each submission is accompanied by metadata, including the student's ID, the assignment identifier, and the timestamp of submission. This structured approach ensures that the data can be easily traced back to individual students and specific assignments. In addition to the raw code, certain features such as student attendance, participation, and assignment difficulty level may be considered as supplementary data points to better contextualize the performance of each student. The diversity in the dataset is crucial, as it captures a wide range of programming skill levels, coding styles, and familiarity with the task at hand.

The data collection process is designed to be both comprehensive and standardized, ensuring that every code submission adheres to a consistent format. This consistency is vital for ensuring that the subsequent stages of feature extraction and model development are based on high-quality, reliable data.

The resulting dataset forms the cornerstone of the feature extraction process, which transforms the raw code into actionable insights to predict student performance. Moreover, it allows for a fair and unbiased assessment of the students' coding abilities, making the data collection phase a critical step in the overall methodology.

```
0    #include <stdio.h>\n\nint main() {\n    int kp...
1    #include <stdio.h>\n\nint main() {\n    float ...
2    #include <stdio.h>\n\nint main() {\n    long z...
3    #include <stdio.h>\n\nint main() {\n    char q...
4    #include <stdio.h>\n\nint main() {\n    int uu...
5    #include <stdio.h>\n\nint main() {\n    char o...
6    #include <stdio.h>\n\nint main() {\n    float ...
7    #include <stdio.h>\n\nint main() {\n    long e...
8    #include <stdio.h>\n\nint main() {\n    float ...
9    #include <stdio.h>\n\nint main() {\n    double...
Name: C Code, dtype: object
```

Fig. 3 C -code Data

Feature Extraction and Grading Methodology:

The feature extraction and grading methodology employed in this research involves a comprehensive analysis of C code to assess various key programming practices. The process begins with extracting specific features that provide insights into the quality and complexity of the submitted code. These features include the lines of code (LOC), which measures the conciseness of the code, and the use of comments, reflecting the student's ability to document their work. The number of variables is counted to gauge the complexity of the code, while the variable types provide an understanding of the data structures the student utilizes. The condition checking and loop count measure the student's proficiency in decision-making and iterative structures. Other features, such as retention of input and well-labeled output, assess the interactivity and clarity of the code, respectively. The presence of a return statement, variable initialization, and use of shorthand notation are evaluated to ensure good programming practices.

The syntax error detection ensures that the code adheres to correct syntax and eliminates potential errors. Once the features are extracted, a grading system is applied. Each feature contributes to the overall score, with specific marks allocated to each feature based on its significance. The total score is normalized to account for the length of the code, and a percentage is calculated. Based on this percentage, a grade is assigned: 'A' for 80% and above, 'B' for 70-79%, 'C' for 60-69%, 'D' for 50-59%, 'E' for 40-49%, and 'F' for below 40%. This hybrid approach allows for a detailed, objective evaluation of programming performance, providing both quantitative scores and qualitative insights into the student's coding abilities.

lines_of_code	use_comments	num_variables	variable_types	condition_checking_count	loop_count
24	Yes	7	6	2	3
7	No	3	2	0	0
16	Yes	3	4	2	1
13	Yes	4	2	0	2
8	No	3	4	0	0
13	Yes	5	10	1	0
9	No	4	8	0	0
7	No	3	4	0	0
7	No	2	1	0	0
11	No	6	9	0	0
9	No	4	4	0	0
18	Yes	7	8	1	2
16	Yes	6	6	1	1
19	Yes	3	2	3	1
16	Yes	5	3	0	3
11	Yes	3	5	1	0
10	Yes	4	4	0	1
10	No	5	7	0	0

Fig. 4 Features Extracted

	lines_of_code	num_variables	variable_types	condition_checking_count	loop_count	Percentage
count	2292.000000	2292.000000	2292.000000	2292.000000	2292.000000	2292.000000
mean	20.229494	7.689354	5.794503	1.656632	1.712042	64.803484
std	6.019222	3.942561	3.977140	1.032592	1.049193	9.579355
min	6.000000	2.000000	0.000000	0.000000	0.000000	29.411765
25%	16.000000	5.000000	2.000000	1.000000	1.000000	60.000000
50%	21.000000	7.000000	6.000000	2.000000	2.000000	65.771449
75%	25.000000	9.000000	8.000000	3.000000	3.000000	71.052632
max	30.000000	21.000000	20.000000	3.000000	3.000000	90.789474

Fig 5 Statistic Summary

1. Count: The number of non-null data points in each feature column. It indicates how many entries are available for each feature.
2. Mean: The average value of each feature. This helps understand the central tendency of the data.
3. Standard Deviation (std): Measures the amount of variation or dispersion of each feature. A higher standard deviation indicates that the data points are spread out over a larger range of values.
4. Minimum (min): The smallest value observed in each feature.
5. 25th Percentile (25%): Also known as the first quartile (Q1), it represents the value below which 25% of the data fall. It helps to understand the distribution of lower values.
6. 50th Percentile (50%): This is the median of the feature, representing the middle point in the dataset, where 50% of the values fall below it.
7. 75th Percentile (75%): Also known as the third quartile (Q3), it indicates the value below which 75% of the data fall.
8. Maximum (max): The largest value observed in each feature.

DATA ANALYSIS:

A thorough data analysis was conducted to understand student coding behavior and its impact on performance evaluation.

ISSN: 2178-7727

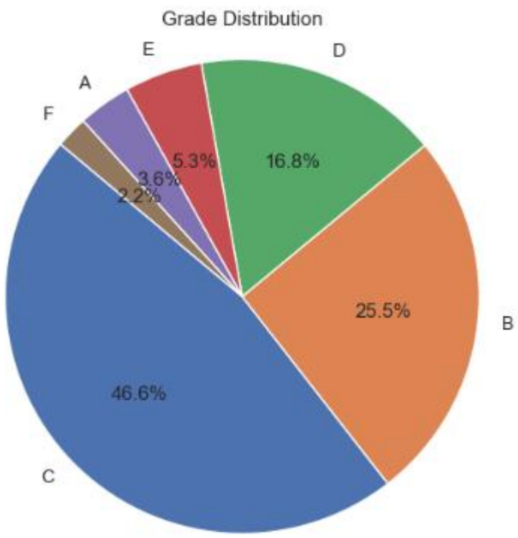

Fig. 6 Grade Distribution

The dataset consisted of various code attributes reflecting structure, logic, clarity, and efficiency. Descriptive statistics were used initially to assess the distribution and range of these attributes. The analysis revealed a balanced mix of concise and moderately lengthy code submissions, with a few instances of overly verbose or minimal code patterns. Patterns in code quality emerged from the analysis, with well-structured logic, clear output formatting, and proper use of control structures positively influencing the final performance scores.

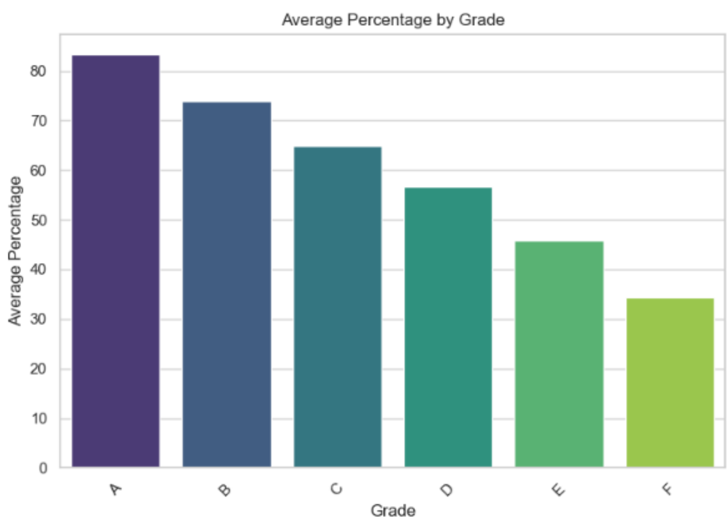

Fig. 7 Average Percentage by Grade

Students who followed best programming practices—such as initializing variables correctly, writing well-documented code, and using control statements appropriately—tended to receive higher grades. correlation analysis indicated that logical complexity, along with clear and consistent code formatting, had a strong impact on predicted outcomes. Code that demonstrated structured thinking and error-free execution aligned with better evaluation results. outlier detection highlighted irregularities in some submissions, such as poor syntax or excessive verbosity, which negatively affected grading. These insights were integrated into the automated grading strategy to ensure fairness and robustness. Overall, the analysis validated the hybrid evaluation model by confirming that good programming practices consistently resulted in better predicted performance and higher classification accuracy.

ISSN: 2178-7727

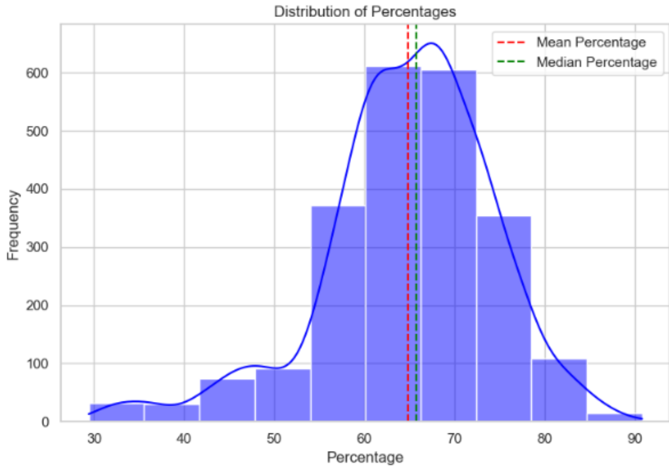

Fig.8 Distribution of percentage

Correlation Matrix Heatmap Analysis:
To further explore the relationships among the extracted coding features and the output variables, a correlation matrix heatmap was plotted. This visualization highlights the strength and direction of linear relationships between pairs of variables.

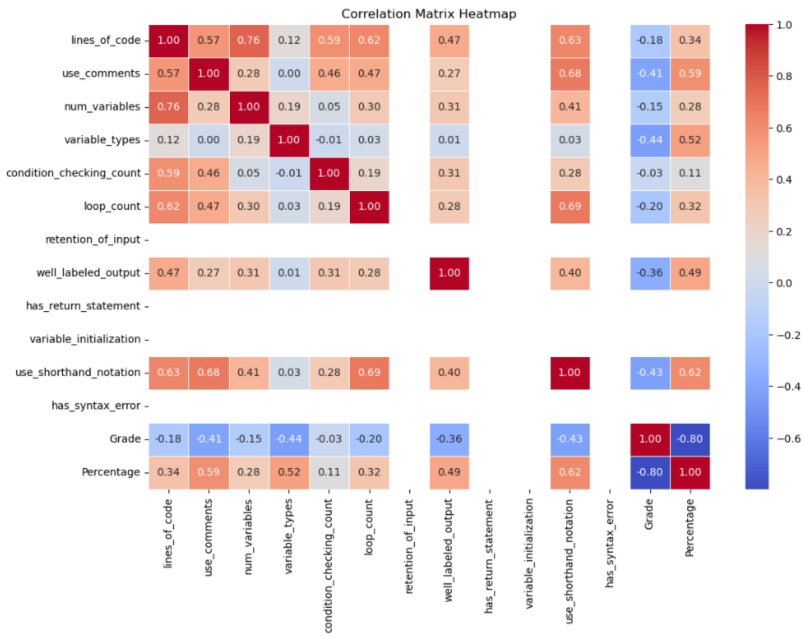

Fig. 9 Correlation Matrix heatmap

Using a color gradient from cool to warm tones, the heatmap clearly indicates which features are positively or negatively correlated. The annotations within each cell provide exact correlation values, allowing for precise interpretation. For instance, a strong positive correlation between specific features and the percentage or grade may suggest that those coding behaviors are critical indicators of performance. This heatmap serves as a valuable tool for identifying influential predictors and potential redundancies in the dataset.

RESULT AND DISCUSSION:
In this project, we developed and evaluated an Artificial Neural Network (ANN) model for predicting student grades and percentages based on various academic performance features. The model was trained and tested using historical data, which included multiple subjects' scores, attendance, study hours, and other relevant factors. The results from the evaluation metrics demonstrate the effectiveness of the model in both regression (predicting percentage) and classification (predicting grades) tasks.

ISSN: 2178-7727

Artificial Neural Network (ANN) Model for Grade Classification

To classify students into appropriate grade categories based on extracted code features, an Artificial Neural Network (ANN) model was developed. The architecture of the ANN comprised multiple densely connected layers, starting with two hidden layers of 512 neurons each, followed by layers with 256, 128, 64, 32, and 16 neurons respectively, all using the ReLU activation function. The final output layer used a SoftMax activation function with five output nodes, corresponding to the five possible grade categories (A, B, C, D, E, F). The model was compiled using the Adam optimizer and trained with the sparse categorical cross-entropy loss function, which is suitable for multi-class classification tasks. The model was trained over 100 epochs with a batch size of 32 and validated using 20% of the training data. After training, the model achieved a strong test accuracy, indicating its capability to generalize well on unseen data. Furthermore, performance metrics such as precision, recall, and F1-score were computed using the classification report, providing deeper insights into the model’s effectiveness across each grade class.

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 512)	6,656
dense_10 (Dense)	(None, 512)	262,656
dense_11 (Dense)	(None, 256)	131,328
dense_12 (Dense)	(None, 256)	65,792
dense_13 (Dense)	(None, 128)	32,896
dense_14 (Dense)	(None, 64)	8,256
dense_15 (Dense)	(None, 32)	2,080
dense_16 (Dense)	(None, 16)	528
dense_17 (Dense)	(None, 5)	85

Fig 10 Model Summary

Model Evaluation and Performance Visualization for Grade Prediction

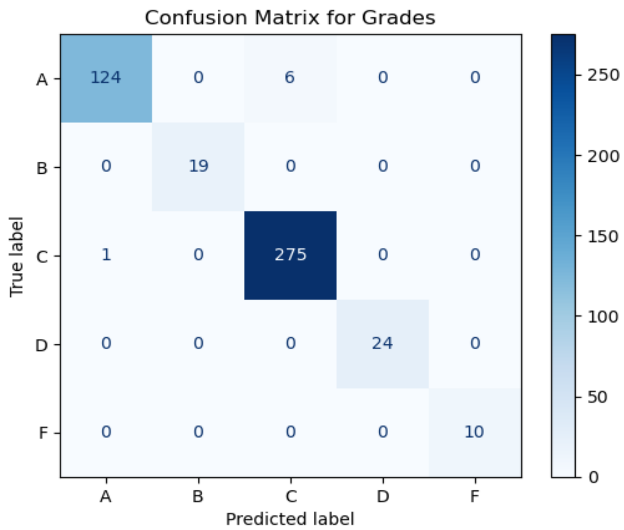

Fig 11 Confusion Matrix

ISSN: 2178-7727

To comprehensively assess the performance of the ANN model in predicting student grades, multiple evaluation metrics and visualizations were utilized. The model achieved a notable test accuracy, demonstrating its generalization capability on unseen data. The classification report provided precision, recall, and F1-scores for each grade category, giving detailed insights into the model's behavior across all classes.

The confusion matrix visually depicted the distribution of true vs. predicted grades, highlighting the model's strengths and misclassification patterns. To understand the model's training dynamics, accuracy and loss plots were generated for both training and validation datasets, showing convergence and stability over the epochs.

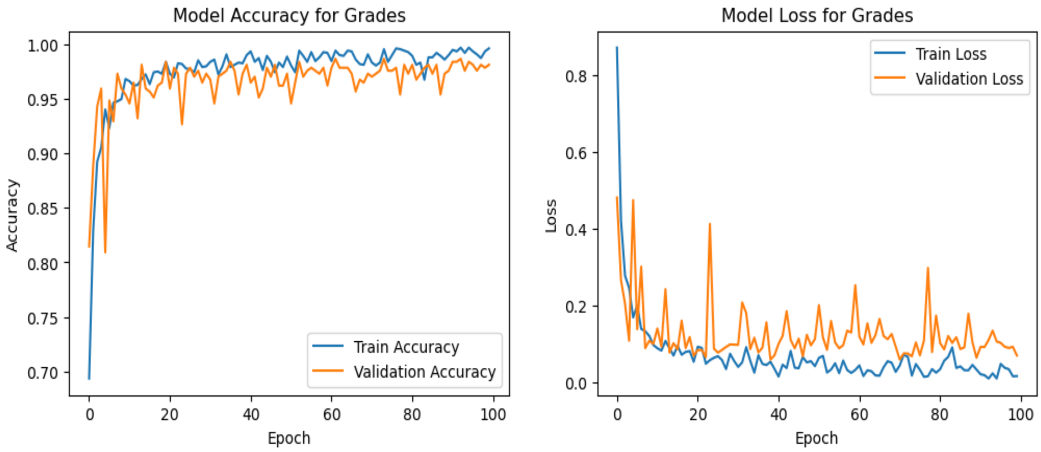

Fig. 12 Model Training Graphs for Grade

Additionally, precision-recall curves were plotted for each grade class, revealing how well the model distinguishes among the categories, especially in imbalanced scenarios. The ROC curves and corresponding AUC scores further validated the model's classification capabilities by measuring its trade-off between true positive and false positive rates for each class. Collectively, these analyses provided a robust validation of the model's effectiveness in grade prediction based on code features.

Table 1: Performance Table for Garde

Grade	Precision	Recall	F1-Score	Support
0	0.99	0.95	0.97	130
1	1.00	1.00	1.00	19
2	0.98	1.00	0.99	276
3	1.00	1.00	1.00	24
4	1.00	1.00	1.00	10

Model Training, Evaluation, and Performance Metrics for Percentage Prediction

The artificial neural network (ANN) model was built to predict student percentages, employing a regression approach. The model includes multiple layers with ReLU activations, culminating in a final linear layer to output the predicted percentage.

Table 2: Performance Table for Percentage

Metric	Value
Mean Squared Error	1.598920
Mean Absolute Error	0.642685
R-squared	0.983755
Explained Variance Score	0.985517

ISSN: 2178-7727

Accuracy Percentage	98.910675

It was compiled with the Adam optimizer and the mean squared error (MSE) loss function, with early stopping implemented to prevent overfitting by monitoring validation loss.

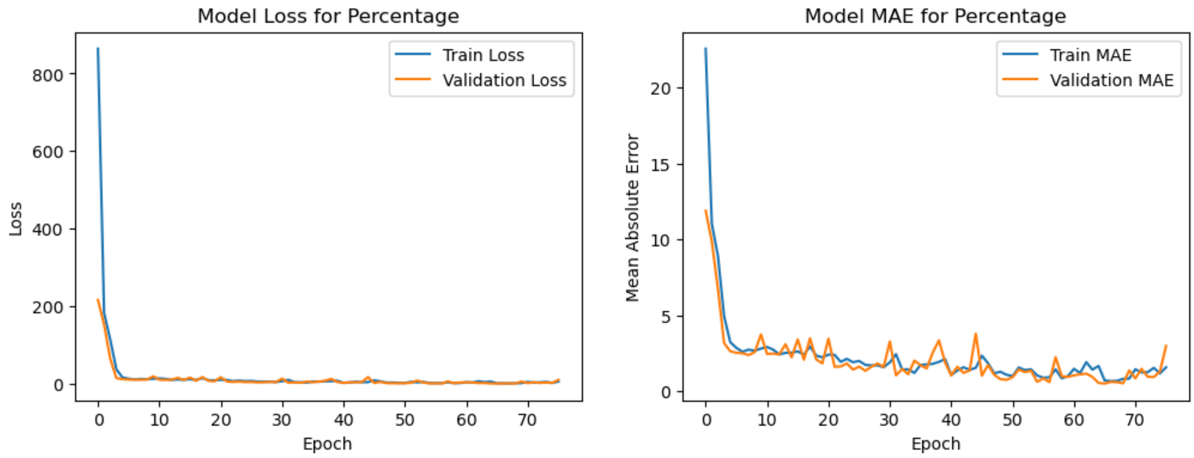

Fig. 13 Model Training Graphs for percentage

After training, several performance metrics were evaluated, including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (indicating the model's fit), Explained Variance (reflecting the model's ability to capture variance in the target variable), and Accuracy Percentage (the proportion of predictions within a predefined threshold, like 5% of the actual value). The training and validation performance were visualized through plots showing the loss and MAE over epochs, providing insights into the model's behavior. All the performance metrics were compiled into a dictionary and then presented in a DataFrame for better visualization. Finally, both the grade prediction model and the percentage prediction model were saved for future use.

Model Performance Evaluation and Visualizations for Percentage Prediction:

First, the Predicted vs True Percentage (Scatter Plot) visually compares the true and predicted percentages, where the red dashed line represents a perfect prediction. Ideally, data points should be close to this line, indicating accurate predictions. Next, the Residuals Distribution (Histogram) showcases the distribution of the residuals, or the differences between the true and predicted values. For a well-performing model, these residuals should be centered around zero and follow a normal distribution.

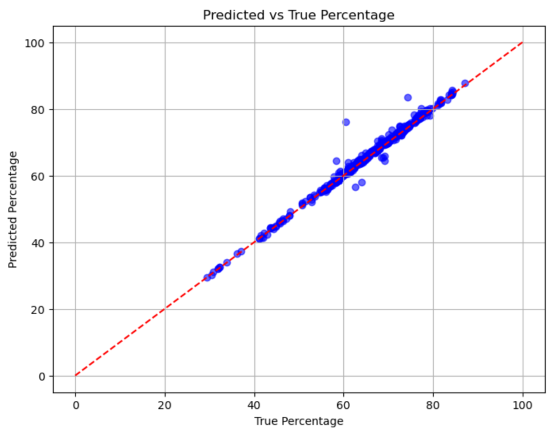

Fig 14 Predicted vs Actual

Additionally, the Loss Curve (Training vs Validation) and MAE Curve (Training vs Validation) plot how the model's training and validation loss, as well as the Mean Absolute Error (MAE), evolve across epochs. Both

ISSN: 2178-7727

curves should ideally decrease over time, indicating the model is learning and generalizing well. If the validation curve diverges while training loss decreases, this could be a sign of overfitting.

The Model Performance Metrics Bar Plot further quantifies the model's performance, displaying the Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared, and Explained Variance Score. These metrics provide valuable insights into the accuracy and quality of the model.

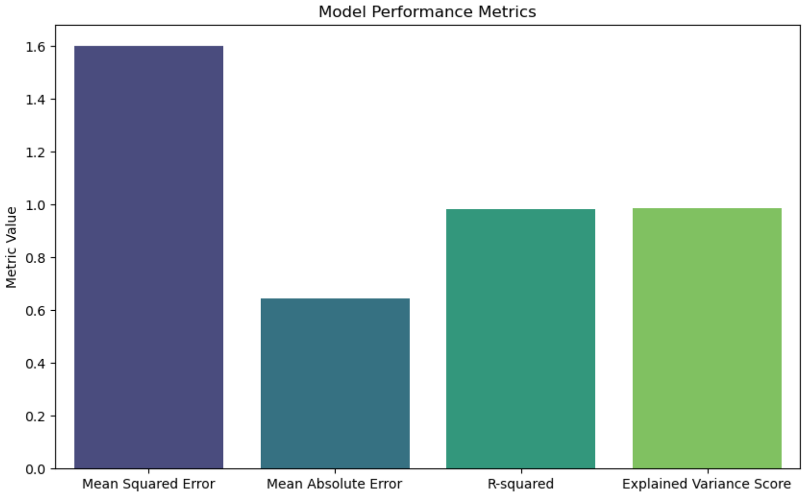

Fig 14 Model Performance Metrics

The Prediction Accuracy (within 5% threshold) graph shows the percentage of predictions that fall within a defined accuracy threshold (e.g., 5%), providing a direct measure of the model's practical accuracy. The Residuals vs Predicted Percentages (Scatter Plot) helps assess if there are any patterns in the residuals. Ideally, these should be randomly scattered around zero, indicating unbiased predictions. Lastly, the Predicted vs True Percentages (Bar Chart) compares the true and predicted values for each test instance, with the goal being that the predicted values (in red) are close to the true values (in green), confirming accurate predictions.

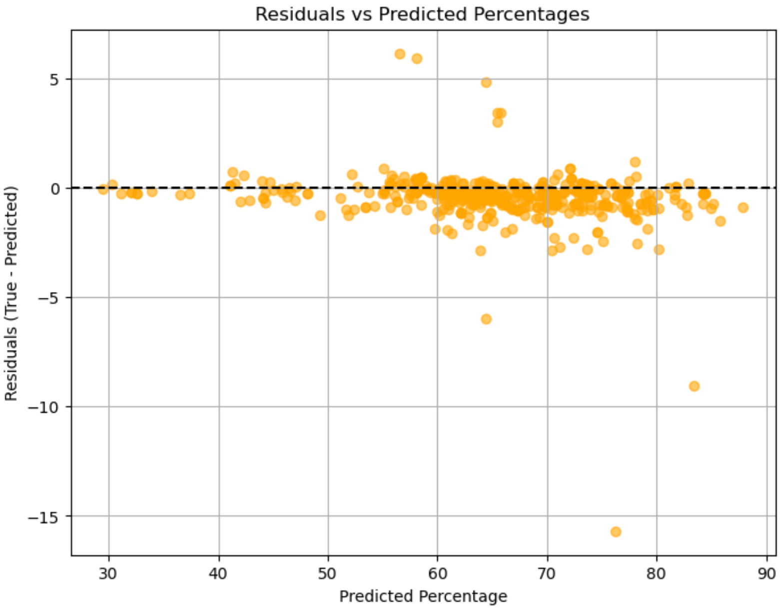

Fig 15 Residuals vs Predicted Percentage

Future Scope

Looking ahead, this hybrid assessment framework can be significantly enhanced by incorporating a broader range of data sources and modeling techniques. Integrating behavioral and psychosocial indicators—such as

ISSN: 2178-7727

engagement metrics, classroom participation, stress levels, and motivation—alongside academic records would enable the model to capture a more holistic view of each student's learning journey. Incorporating learning style preferences and extracurricular activity data could further personalize predictions and recommendations. From a modeling perspective, experimenting with ensemble methods (e.g., Random Forests, Gradient Boosting) and fine-tuning hyperparameters could improve accuracy and robustness, while advanced regularization and early-stopping strategies would guard against overfitting. deploying the system in real time within existing Learning Management Systems (LMS) would allow continuous monitoring and on-the-fly performance prediction, providing immediate, actionable feedback to students. This could power individualized learning plans, recommending targeted resources or remedial exercises for those predicted to struggle. Early-warning alerts for at-risk students would facilitate timely interventions such as tutoring or peer mentoring.

Furthermore, extending the model to handle multimodal inputs—analyzing textual submissions and discussion forum posts with NLP techniques, or assessing student engagement through video and audio analytics—would deepen the model's understanding of student behaviors and competencies. Finally, enhancing interpretability through model-agnostic explanation tools (e.g., SHAP, LIME) would make the predictions transparent to educators and learners alike, highlighting which factors most influence performance and enabling data-driven pedagogical improvements.

CONCLUSION

This study has successfully demonstrated the application of machine learning techniques in predicting student academic outcomes based on a variety of features, including performance in specific subjects, attendance, study hours, and psychosocial factors. The use of an Artificial Neural Network (ANN) for predicting grades and percentage has proven to be effective, with satisfactory model performance metrics and visualizations supporting the robustness of the predictions. The model's ability to predict academic success, combined with its ability to identify students at risk of underperformance, holds great potential for transforming educational environments by providing personalized, data-driven insights into student learning. through the integration of multiple features, this system offers a scalable and adaptable solution for predicting academic achievement, which can be tailored to different educational settings and curricula. Furthermore, the inclusion of early intervention strategies based on model predictions can greatly enhance support for students, leading to improved academic outcomes. overall, this research highlights the power of machine learning in educational settings, opening doors for more personalized, proactive approaches to student development, ensuring timely and effective interventions to improve overall academic performance.

REFERENCES:

[1] Jerich Faddar et al., "Perspectives on educational effectiveness in science and mathematics: The role of noncognitive measures in TIMSS. Introduction to a special issue" ELSEVIER, Volume 75, 2022.

[2] Elisabeth Bauer et al., "Simulation-based learning in higher education and professional training: Approximations of practice through representational scaffolding" ELSEVIER, Volume 75, 2022.

[3] Markus Berndt et al., "Impact of sender and peer-feedback characteristics on performance, cognitive load, and mindful cognitive processing" ELSEVIER, Volume 75, 2022.

[4] Utkarsh Verma et al., "Prediction of students' academic performance using Machine Learning Techniques" IEEE,2020.

[5] Adel Ismail Al-Alawi et al., "Predicting Student's Academic Performance Using Data Mining Methods: Review Paper" IEEE,2023.

[6] Sagardeep Roy et al., "Predicting academic performance of student using classification techniques" IEEE,2017.

[7] Meizar Raka Rimadana et al., "Predicting Student Academic Performance using Machine Learning and Time Management Skill Data" IEEE,2019.

[8] Bo Guo et al., "Predicting Students Performance in Educational Data Mining" IEEE,2015.

[9] Divya Thakur et al., "Predicting Student's Performance using Data Mining Algorithm" IEEE,2022.

ISSN: 2178-7727

[10] Praveena Chakrapani et al., “Academic Performance Prediction Using Machine Learning: A Comprehensive & Systematic Review” IEEE,2022.
[11] Nehal Eleyan et al., “Predicting Student Performance Using Educational Data Mining” IEEE, 2022.
[12] D. K. Arun et al., “Student Academic Performance Prediction using Educational Data Mining” IEEE,2021.
[13] Fergie Joanda Kaunang et al., “Students' Academic Performance Prediction using Data Mining” IEEE,2018.
[14] K. Kaino et al., “Origami modeling method of leaves of plants and CG image generation of flower arrangement” IEEE, 2002.
[15] Neeta Sharma et al., “Predicting Academic Performance of Students Using Machine Learning Models” IEEE,2023.
[16] Tismy Devasia et al., “Prediction of students performance using Educational Data Mining” IEEE,2016.
[17] Gomathy Suganya Ramaswami et al., “Predicting Students Final Academic Performance using Feature Selection Approaches” IEEE,2020.
[18] Aws Khudhur et al., “Students' Performance Prediction Using Machine Learning Based on Generative Adversarial Network” IEEE,2023.
[19] Essa Alhazmi et al., “Early Predicting of Students Performance in Higher Education” IEEE, Volume 18,2023. 563 J INFORM SYSTEMS ENG, 10(7s)
[20] Roselyne Ayienda et al., “Predicting Students Academic Performance using a Hybrid of Machine Learning Algorithms” IEEE,2021.
[21] Mohamed Mohsen Elsaid Khoudier et al., “Prediction of student performance using machine learning techniques” IEEE, 2023.
[22] Muslihah Binti Wook et al., “Predicting NDUM Student's Academic Performance Using Data Mining Techniques” IEEE,2009.
[23] Isreal M. Ogundele et al., “Prediction of Student Academic Performance Based on Machine Learning Model” IEEE,2024.
[24] Guiyun Feng et al., “Analysis and Prediction of Students’ Academic Performance Based on Educational Data Mining” IEEE, Volume 10, 2022.
[25] Wentong Liu et al., “Student Performance Prediction by LMS Data and Classroom Videos” IEEE,2022.
[26] Poonam S. Pawar et al., “A review on Student Performance Prediction using Educational Data mining and Artificial Intelligence” IEEE, 2021.
[27] Behrouz Minaei-Bidgoli et al., “Predicting student performance: an application of data mining methods with an educational Web-based system” IEEE, 2004.
[28] Sujith Jayaprakash et al., “Predicting Students Academic Performance using an Improved Random Forest Classifier” IEEE, 2020.
[29] Ching-Chieh Kiu et al., “Data Mining Analysis on Student’s Academic Performance through Exploration of Student’s Background and Social Activities” IEEE,2018.
[30] Xiaochen Lai et al., “Predicting the academic performance of students with GPcSAGE” IEEE,2023.
[31] Rashmi Jha et al., “A Systematic Study on Student Performance Prediction from the Perspective of Machine Learning and Data Mining Approaches” IEEE, 2023.
[32] K. Deepika et al., “Analyze and Predicting the Student Academic Performance Using Data Mining Tools” IEEE,2018.
[33] Shoukath Tk et al., “Academic Performance Prediction of At-Risk Students using Machine Learning Techniques” IEEE, 2023.
[34] Md Fahim Sikder et al., “Predicting students yearly performance using neural network: A case study of BSMRSTU” IEEE,2016.
[35] Jacob Molina-Astorayme et al., “Predicting academic performance using automatic learning techniques: a review of the scientific literature” IEEE,2020.

ISSN: 2178-7727

[36] S.N. van den Boom-Muilenburg et al., ELSEVIER,2023. “Sustaining data use professional learning communities in schools: The role of leadership practices”
[37] ZILING CHEN et al., IEEE,2023. “Student Performance Prediction Approach Based on Educational Data Mining”
[38] Evandro B. Costa et al., ELSEVIER , “Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses” 2017.
[39] Eyman Alyahyan et al., IJETHE, “Predicting academic success in higher education: literature review and best practices” 2020.
[41] Yi Ren et al., "Long-term student performance prediction using learning ability self-adaptive algorithm," Springer, 2024.
[42] Shah Hussain et al., "Student-Performulator: Predicting Students’ Academic Performance at Secondary and Intermediate Level Using Machine Learning," Springer, 2023.
[43] Chunpeng Zhai et al., "The effects of over-reliance on AI dialogue systems on students’ cognitive abilities: a systematic review," Springer, 2024.
[44] Mustafa Yağc., "Educational data mining: prediction of students’ academic performance using machine learning algorithms," Springer, 2022.
[45] Yudish Teshal Badal et al., "Predictive modelling and analytics of students’ grades using machine learning algorithms," Springer, 2023.
[46] Alberto Rivas et al., "Artificial neural network analysis of the academic performance of students in virtual learning environments," Elsevier, 2020.
[47] Pallavi Asthana et al., "Prediction of Student’s Performance with Learning Coefficients Using Regression Based Machine Learning Models," IEEE, 2023.
[48] Luis Vives et al., "Prediction of Students’ Academic Performance in the Programming Fundamentals Course Using Long Short-Term Memory Neural Networks," IEEE, 2023.
[49] Barrett MA, Humblet O, Hiatt RA, Adler NE. Big data and disease prevention: From quantified self to quantified communities. Big Data 2013;1(3):168–75. http://doi.org/10.1089/big.2013.0027
[50] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential", Health InfSciSyst, vol. 2, no. 1, p. 3, 2014.
[51] Big Data Analytics in Health,© Canada Health Infoway Inc. 2013―, https://www.infoway-inforoute.ca/en/component/edocman/1246-big-dataanalyticsinhealth-white-paper fullreport/viewdocumenthttps://med.stanford.edu/content/dam/sm/smnews/documents/StanfodMedicineHealthTrendsWhitePaper2017.
[52] J. Soni, U. Ansari, D. Sharma and S. Soni, "Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction", International Journal of Computer Applications, vol. 17, no. 8, pp. 43-48, 2011.
[53] D. C. Bindushree and V. U. Rani, "A review on using various DM techniques for evaluation of performance and analysis of heart disease prediction," 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), Bangalore, 2017, pp. 686-690. doi: 10.1109/SmartTechCon.2017.8358459
[54] C. S. Dangare and S. S. Apte, "Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques", International Journal of Computer Applications, vol. 47, no. 10, pp. 44-48, 2012.
[55] Prachi Paliwal, Mahesh Malviya, An Efficient Method for Predicting Heart Disease Problem Using Fitness Value‖, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (2), 2015, 1290-1293
[56] J. Soni, U. Ansari, D. Sharma and S. Soni, "Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction", International Journal of Computer Applications, vol. 17, no. 8, pp. 43-48, 2011.
[57] M. Akhil Jabbar, B.L. Deekshatulub, Priti Chandra, ―Classification of Heart Disease Using K-Nearest Neighbour and Genetic Algorithm‖, First International Conference on Computational Intelligence: Modelling Techniques and Applications (CIMTA) 2013 Volume 10, 2013, Pages 85–94

ISSN: 2178-7727

[58] M. Akhil Jabbar, B. L Deekshatulu and Priti Chandra, ―Heart Disease Prediction System using Associative Classification and Genetic Algorithm‖, International Conference on Emerging Trends in Electrical, Electronics and Communication Technologies-ICECIT, 2012 [59] A. Makwana and J. Patel, "Decision Support System for Heart Disease Prediction using Data Mining Techniques", International Journal of Computer Applications, vol. 117, no. 22, pp. 1-5, 2015.

[60] K. Kavitha, K. Ramakrishnan and M. Singh, "Modelling and design of evolutionary neural network for heart disease detection", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010, vol. 7, no. 5, p. 272, 2010.

[61] Nitin Kumari, Sunita, Smita, ‖Comparison of ANNs, Fuzzy Logic and NeuroFuzzy Integrated Approach for Diagnosis of Coronary Heart Disease: A Survey‖, International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 2, Issue. 6, June 2013, pg.216 – 224

[62] R. Chitra and V. Seenivasagam, ―Review of heart disease prediction system using data mining and hybrid intelligent techniques", ICTACT journal on soft computing, July 2013, VOLUME: 03, ISSUE: 04, ISSN: 2229-6956(ONLINE)

[63] Shamsher Bahadur Patel, Pramod Kumar Yadav, Dr. D. P. Shukla, ―Predict the Diagnosis of Heart Disease Patients Using Classification Mining Techniques‖, IOSR Journal of Agriculture and Veterinary Science (IOSR-JAVS) e-ISSN: 2319- 2380, p-ISSN: 2319-2372. Volume 4, Issue 2 (Jul. - Aug. 2013), PP 61- 64 www.iosrjournals.org]

[64] M.A. Nishara Banu, B Gomathy, ― Disease Predicting System Using Data Mining Technique's‖ International Journal of Technical Research and Applications eISSN: 2320-8163, www.ijtra.com Volume 1, Issue 5 (Nov-Dec 2013), PP. 41-45]

[65] R. Chitra, Dr. V. Sreenivasagam, ―Heart disease Prediction System Using Supervised Learning Classifier‖Bonfring International Journal of Software Engineering and Soft Computing, Vol. 3, No. 1, March2013

[66] S. U. Amin, K. Agarwal and R. Beg, "Genetic neural network based data mining in prediction of heart disease using risk factors," Information & Communication Technologies (ICT), 2013 IEEE Conference on, JeJu Island, 2013, pp. 1227-1231. doi: 10.1109/CICT.2013.6558288

[67] Jimson Mathew, Ashutosh Kumar Singh, Ilayaraja M., Meyyappan T., ―Efficient Data Mining Method to Predict the Risk of Heart Diseases Through Frequent Itemsets, Procedia Computer Science‖, Proceedings of the 4th International Conference on Eco-friendly Computing and Communication Systems Volume 70, 2015, Pages 586-592, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2015.10.040.

[68] K. Shailaja, B. Seetharamulu, M. A. Jabbar, ―Machine Learning in Healthcare: A Review‖, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2018, pp.910-914. DOI: 10.1109/ICECA.2018.8474918

[69] Himanshu Sharma, M ARizvi, ―Prediction of Heart Disease using Machine Learning Algorithms: A Survey‖, 2017 International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169

[70] AditiGavhane, GouthamiKokkula, IshaPandy, Prof. Kailas Devadkar, ―Prediction of Heart Disease Using Machine Learning‖, 2018 Second Internatinal Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2018, pp.1275-1275. DOI:10.1109/ICECA.20188474922

[71] AH Chen, SY Huang, PS Hong, CH Cheng, EJ Lin, ―HDPS: Heart Disease Prediction System‖ 2011 Computing in Cardiology, Hangzhou, 2011, pp.557-560.

[72] Theresa Princy. R, J. Thomas, ―Human Heart Disease Prediction System using Data Mining Techniques‖, 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, 2016, pp.1-5 DOI:10.1109/ICCPCT.2016.7530265

[73] C. Sowmiya, Dr. P. Sumitra, Analytical Study of Heart Disease Diagnosis Using Classification Techniques‖, 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Srivilliputhur, 2017, pp.1-5. DOI:10.1109/ITCOSP.2017.8303115

ISSN: 2178-7727

[74] Purushottam, Prof. KanakSaxena, Richa Sharma, ―Efficient Heart Disease Prediction System using Decision Tree‖, 2015 International Conference on Computing, Communication & Automation, Noida, 2015 pp.72-77. DOI:10.1109/CCAA.2015.7148346
[75] M.A. Jabbar and S. Samreen, Heart disease prediction system based on hidden naïve bayes classifier, 2016 International Conference on Circuits, Controls, Communications and Computing(I4C), Bangalore, 2016, pp.1-5. DOI: 10.1109/CIMCA.2016.8053261
[76] S.M.M. Hasaan, M.A. Manun, M.P. Uddin and M.A. Hossain, Comparative Analysis of Classification Approaches for Heart Disease Prediction,2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IMEC), Rajshahi, 2018, pp.1-4 DOI: 10.1109/IC4ME2.2018.8465594
[78] S. Bharti and S, N. Singh, ―Analytical Study of heart disease prediction comparing with different algorithms, Internatinal Conference on Computing, Communication & Automation, Noida, 2015, pp.78-82 DOI: 10.1109/CCAA.2015.7148347
[79] J. Singh, A. Kamra and H. Singh, Prediction of heart diseases using associative classification, 2016 5th International Conference on Wireless Networks and Embedded Systems (WECON), Rajpura, 2016, pp.1-7. DOI: 10.1109/WECON.2016.7993480
[80] K. G. Dinesh, K. Arumugaraj, K. D. Santhosh and V. Mareeswari, "Prediction of Cardiovascular Disease Using Machine Learning Algorithms," 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, 2018, pp.1-7. DOI: 10.1109/ICCTCT.2018.8550857
[81] V.V. Ramalingam, Ayantan Dandapath, M Karthik Raja ―Heart Disease Prediction Using Machine Learning Techniques: A Survey‖,2018 2018 International Journal of Engineering & Technology DOI: 10.14419/ijet.v7i2.8.10557
[82] Rajesh N, T Maneesha, Shaik Hafeez, Hari Krishna ―Prediction of Heart Disease Using Machine Learning Algorithms‖, 2018 International Journal of Engineering & TechnologyDOI:10.14419/ijet.v7i2.32.15714
[83] B. D. Kanchan and M. M. Kishor, "Study of machine learning algorithms for special disease prediction using principal of component analysis," 2016 International Conference on Global Trends in Single Processing, Information Computing and Communication (ICTGSPICC), Jalgaon, 2016, pp.5-10. DOI: 10.1109/ICGTSPICC.20167955260
[84] S. Pouriyeh, S. Vahid, G. Sannino, G. De Pietro, H. Arabnia and J. Gutierrez, "A comprehensive investigation and comparison of Machine Learning Techniques in the domain of heart disease," 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, 2017, pp.204-207. DOI: 10.1109/ISCC.2017.8024530
[85] S. Kumar, P. Singh and S. Rani, "Sentimental analysis of social media using R language and Hadoop: Rhadoop," 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, 2016, pp. 207-213.doi: 10.1109/ICRITO.2016.7784953
[86] Hager Ahmed, Eman M.G. Younis, AbdeltawabHendawi, Abdelmgeid A. Ali, Heart disease identification from patients' social posts, machine learning solution on Spark, Future Generation Computer Systems, 2019, ISSN 0167-739X,https://doi.org/10.1016/j.future.2019.09.056.arXiv:1610.02445 [cs.HC]
[87] Yan M., Deng Z., Sang J., Xu C. (2013) User-Oriented Social Analysis across Social Media Sites. In: Petrosino A., Maddalena L., Pala P. (eds) New Trends in Image Analysis and Processing – ICIAP 2013. ICIAP 2013. Lecture Notes in Computer Science, vol 8158. Springer, Berlin, Heidelberg
[88] N. AZMINA M. ZAMANI, SITI Z. Z. ABIDIN, NASIROH OMAR, M. Z. Z. ABIDEN, ―Sentiment Analysis: Determining People's Emotions in Facebook, Applied Computational Science, ISBN: 978-960-474-368-1
[89] B. Gokulakrishnan, P. Priyanthan, T. Ragavan, N. Prasath and A. Perera, "Opinion mining and sentiment analysis on a Twitter data stream," International Conference on Advances in ICT for Emerging Regions (ICTer2012), Colombo, 2012, pp. 182- 188. doi: 10.1109/ICTer.2012.6423033

ISSN: 2178-7727

[90] L. Jiang and C. C. Yang, "Using Co-occurrence Analysis to Expand Consumer Health Vocabularies from Social Media Data," 2013 IEEE International Conference on Healthcare Informatics, Philadelphia, PA, 2013, pp. 74-81. doi: 10.1109/ICHI.2013.16
[91] Kotov, Alexander. (2015). Social Media Analytics for Healthcare. 10.13140/RG.2.1.2301.8409.
[92] Safeek, Ilham & Kalideeen, Muhammad. (2017). PREPROCESSING ON FACEBOOK DATA FOR SENTIMENT ANALYSIS. 7th International Symposium 2017 (IntSym2017)- SEUSL - 07th & 08th December 2017 At: Oluvil, Sri Lanka
[93] Cordos, Ariana & Bolboaca, Sorana & Drugan, Tudor. (2016) Social Media as Source of Medical Information for Healthcare Students. Applied Medical Informatics. 38. 2016.
[94] Wisdom, Vivek. (2016). An introduction to Twitter Data Analysis in Python. 10.13140/RG.2.2.12803.30243.
[95] Gamache-O'Leary, Valerie & Grant, Gerald. (2017). Social Media in Healthcare. http://hdl.handle.net/10125/41614
[96] Sri Khetwat, Saritha & Devshriroy, Devshriroy. (2012). Semantic Orientation of Sentiment Analysis on Social Media. INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY. 11. 2401-2409. 10.24297/ijct.v11i4.3124.
[97] Giriş, Fatih. (2014). SENTIMEDIA: OPINION MINING AND SENTIMENT ANALYSIS ON SOCIAL MEDIA. 10.13140/RG.2.2.35785.03681.
[98] AmanKarn, AnishShrestha, Anil Pudasaini, BinayMahara, AnkuJaiswal, ―Statistic-Based Sentiment Analysis of Social Media Data‖, International Research Journal of Innovations in Engineering and Technology (IRJIET), Volume 2, Issue 5, pp 28-32, July 2018.
[99] Jayasanka, Sachira & Madhushani, Thilina & Marcus, E. & Aberathne, I. & Premaratne, Saminda. (2013). Sentiment Analysis for Social Media.
[100] Wegrzyn-Wolska, Katarzyna & Bougueroua, Lamine & Dziczkowski, Grzegorz. (2011). Social media analysis for e-health and medical purposes. Proceedings of the 2011 International Conference on Computational Aspects of Social Networks, CASoN'11. 278-283. 10.1109/CASON.2011.6085958.
[101] M. S. Haydar, M. Al Helal and S. A. Hossain, "Sentiment Extraction From Bangla Text: A Character Level Supervised Recurrent Neural Network Approach," 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Rajshahi, 2018, pp. 1-4. doi: 10.1109/IC4ME2.2018.8465606
[102] K. Baktha and B. K. Tripathy, "Investigation of recurrent neural networks in the field of sentiment analysis," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017, pp. 2047-2050. doi: 10.1109/ICCSP.2017.8286763
[103] D. Mahajan and D. Kumar Chaudhary, "Sentiment Analysis Using Rnn and Google Translator," 2018 8th International Conference on Cloud Computing, Data Science &Engineering (Confluence), Noida, 2018, pp. 798-802. doi: 10.1109/CONFLUENCE.2018.8442924
[104] Mukherjee, S. Mukhopadhyay, P. K. Panigrahi and S. Goswami, "Utilization of Oversampling for multiclass sentiment analysis on Amazon Review Dataset," 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), Morioka, Japan, 2019, pp. 1-6. doi: 10.1109/ICAwST.2019.8923260
[105] D. R. Pant, P. Neupane, A. Poudel, A. K. Pokhrel and B. K. Lama, "Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis," 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, 2018, pp. 128-132.doi: 10.1109/CCCS.2018.8586824
[106] D. Vishwanath and S. Gupta, "Adding CNNs to the Mix: Stacking models for sentiment classification," 2016 IEEE Annual India Conference (INDICON), Bangalore, 2016, pp. 1-4. doi: 10.1109/INDICON.2016.7839062
[107] G. Xu, Y. Meng, X. Qiu, Z. Yu and X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," in IEEE Access, vol. 7, pp. 51522-51532, 2019. doi: 10.1109/ACCESS.2019.2909919
[108] M. Day and Y. Lin, "Deep Learning for Sentiment Analysis on Google Play Consumer Review," 2017 IEEE International Conference on Information Reuse and Integration (IRI), San Diego, CA, 2017, pp. 382-388. doi: 10.1109/IRI.2017.79

ISSN: 2178-7727

[109] C. Li, B. Xu, G. Wu, S. He, G. Tian and H. Hao, "Recursive Deep Learning for Sentiment Analysis over Social Data," 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Warsaw, 2014, pp. 180-185. doi: 10.1109/WI-IAT.2014.96

[110] V. Goel, A. K. Gupta and N. Kumar, "Sentiment Analysis of Multilingual Twitter Data using Natural Language Processing," 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2018, pp. 208-212. doi: 10.1109/CSNT.2018.8820254

[111] D. Li and J. Qian, "Text sentiment analysis based on long short-term memory," 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI), Wuhan, 2016, pp. 471-475. doi: 10.1109/CCI.2016.7778967

[112] B. Ciftci and M. S. Apaydin, "A Deep Learning Approach to Sentiment Analysis in Turkish," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 2018, pp. 1-5. doi: 10.1109/IDAP.2018.8620751

[113] G. Preethi, P. V. Krishna, M. S. Obaidat, V. Saritha and S. Yenduri, "Application of Deep Learning to Sentiment Analysis for recommender system on cloud," 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, 2017, pp.93-97.doi: 10.1109/CITS.2017.8035341

[114] D. Goularas and S. Kamis, "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), Istanbul, Turkey, 2019, pp. 12-17. doi: 10.1109/Deep-ML.2019.00011

[115] Aziz Sharfuddin, M. NafisTihami and M. Saiful Islam, "A Deep Recurrent Neural Network with BiLSTM model for Sentiment Classification," 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, 2018, pp. 1-4. doi: 10.1109/ICBSLP.2018.8554396

[116] S. SeshathriAathithyan, M. V. Sriram, S. Prasanna and R. Venkatesan, "Affective — Hierarchical classification of text — An approach using NLP toolkit," 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, 2016, pp. 1-6. doi: 10.1109/ICCPCT.2016.7530228

[117] C. Li, G. Zhan and Z. Li, "News Text Classification Based on Improved BiLSTM-CNN," 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, 2018, pp. 890-893. doi: 10.1109/ITME.2018.00199

[118] S. K. Srivasatava, R. Kumari and S. K. Singh, "An ensemblebased NLP feature assessment in binary classification," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017, pp. 345-349. doi: 10.1109/CCAA.2017.8229840

[119] H. Liu, L. Li, Y. Chen, S. Liu and X. Zhang, "Multi-label Classification for Marine Science and Technology Literature," 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, 2018, pp. 1063-1066. doi: 10.1109/IMCEC.2018.8469717

[120] M Kanakaraj and R. M. R. Guddeti, "Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques," Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), Anaheim, CA, 2015, pp. 169-170. doi: 10.1109/ICOSC.2015.7050801

[121] T. Ghorpade and L. Ragha, "Featured based sentiment classification for hotel reviews using NLP and Bayesian classification," 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, 2012, pp. 1- 5. doi: 10.1109/ICCICT.2012.6398136

[122] P. Song, C. Geng and Z. Li, "Research on Text Classification Based on Convolutional Neural Network," 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 2019, pp. 229-232. doi: 10.1109/ICCNEA.2019.00052 [123] V. K. Soni and S. Pawar, "Emotion based social media text classification using optimized improved ID3 classifier," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 1500-1505. doi: 10.1109/ICECDS.2017.8389696

ISSN: 2178-7727

[124] K. Saranya and S. Jayanthy, "Onto-based sentiment classification using machine learning techniques," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2017, pp. 1-5. doi: 10.1109/ICIIECS.2017.8276047
[125] A. A. Helmy, Y. M. K. Omar and R. Hodhod, "An Innovative Word Encoding Method For Text Classification Using Convolutional Neural Network," 2018 14th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 2018, pp. 42- 47. doi: 10.1109/ICENCO.2018.8636143
[126] S. Yuenyong, N. Hnoohom, K. Wongpatikaseree and T. P. N. Ayutthaya, "Classification of Tweets Related to Illegal Activities in Thai Language," 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Pattaya, Thailand, 2018, pp. 1-6. doi: 10.1109/iSAI-NLP.2018.8692858
[127] K. Sundus, F. Al-Haj and B. Hammo, "A Deep Learning Approach for Arabic Text Classification," 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, Jordan, 2019, pp. 1-7. doi: 10.1109/ICTCS.2019.8923083
[128] L. Zhu, G. Wang and X. Zou, "A Study of Chinese Document Representation and Classification with Word2vec," 2016 9th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, 2016, pp. 298-302. doi: 10.1109/ISCID.2016.1075
[129] J. Savigny and A. Purwarianti, "Emotion classification on youtube comments using word embedding," 2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), Denpasar, 2017, pp. 1-5. doi: 10.1109/ICAICTA.2017.8090986
[130] M. G. Thushara, T. Mownika and R. Mangamuru, "A Comparative Study on different Keyword Extraction Algorithms," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 969- 973.doi: 10.1109/ICCMC.2019.8819630
[131] P. Wongchaisuwat, "Automatic Keyword Extraction Using TextRank," 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), Tokyo, Japan, 2019, pp. 377-381. doi: 10.1109/IEA.2019.8714976
[132] Y. Tao, Z. Cui and Z. Jiazhe, "Research on Keyword Extraction Algorithm Using PMI and TextRank," 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 2019, pp. 5-9.doi: 10.1109/INFOCT.2019.8711099 [133] X. Zhang, Y. Wang and L. Wu, "Research on Cross Language Text Keyword Extraction Based on Information Entropy and TextRank," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, pp. 16-19. doi: 10.1109/ITNEC.2019.8728993
[134] S. Ganiger and K. M. M. Rajashekharaiah, "Comparative Study on Keyword Extraction Algorithms for Single Extractive Document," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 1284-1287.doi: 10.1109/ICCONS.2018.8663040
[135] P. Sun, L. Wang and Q. Xia, "The Keyword Extraction of Chinese Medical Web Page Based on WF-TF-IDF Algorithm," 2017 International Conference on CyberEnabled Distributed Computing and Knowledge Discovery (CyberC), Nanjing, 2017, pp. 193-198. doi: 10.1109/CyberC.2017.40
[136] X. Cai and S. Cao, "A Keyword Extraction Method Based on Learning to Rank," 2017 13th International Conference on Semantics, Knowledge and Grids (SKG), Beijing, 2017, pp. 194-197.doi: 10.1109/SKG.2017.00040
[137] P. Zeng, Q. Tan, Y. Yan, Q. Xie, J. Xu and W. Cao, "Automatic Keyword Extraction Using Word Embedding and Clustering," 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), Dalian, 2017, pp. 1402-1408. doi: 10.1109/ICCSEC.2017.8447033
[138] F. Bayatmakou, A. Ahmadi and A. Mohebi, "Automatic query-based keyword and keyphrase extraction," 2017 Artificial Intelligence and Signal Processing Conference (AISP), Shiraz, 2017, pp. 325-330.doi: 100.1109/AISP.2017.8515121
[139] M. Zhao, W. Yu, W. Lu, Q. Liu and J. Li, "Chinese Document Keyword Extraction Algorithm Based on FP-growth," 2016 International Conference on Smart City and Systems Engineering (ICSCSE), Hunan, 2016, pp. 202-205. doi: 10.1109/ICSCSE.2016.0062

ISSN: 2178-7727

[140] W. Li and J. Zhao, "TextRank Algorithm by Exploiting Wikipedia for Short Text Keywords Extraction," 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), Beijing, 2016, pp. 683-686. doi: 10.1109/ICISCE.2016.151

[141] S. Bing and Fu-Lei, "Keyword Extraction Algorithm Based on," 2013 International Conference on Computational and Information Sciences, Shiyang, 2013, pp. 664-665. doi: 10.1109/ICCIS.2013.180

[142] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[143] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[144] Rousseauw, J., du Plessis, J., Benade, A., Jordaan, P., Kotze, J. and Ferreira, J. (1983). Coronary risk factor screening in three rural communities, South African Medical Journal 64: 430–436.

[145] R. Alizadehsani et al., ―A data mining approach for diagnosis of coronary artery disease, Computer Methods and Programs in Biomedicine, vol.111, no.1, pp.52- 61, Jul. 2013. https://en.wikipedia.org/wiki/Facebook

[146] R. He, J. McAuley, ―Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering‖ WWW 2016, April 11–15, 2016, Montréal, Québec, Canada. ACM 978-1-4503-4143-1/16/04. DOI: http://dx.doi.org/10.1145/2872427.2883037. [147] J. McAuley, C. Targett, J. Shi, A. van den Hengel. Image-based recommendations on styles and substitutes. SIGIR, 2015https://en.wikipedia.org/wiki/Snapchat https://en.wikipedia.org/wiki/Instagram

[148] Ashwin Belle, Raghuram Thiagarajan, S. M. Reza Soroushmehr, FatemehNavidi, Daniel A. Beard, and KayvanNajarian, ―Big Data Analytics in Healthcare, BioMed Research International, vol. 2015, Article ID 370194, 16 pages, 2015. https://doi.org/10.1155/2015/370194.

[149] M. Chen, Y. Hao, K. Hwang, L. Wang, L. Wang, ―Disease prediction by machine learning over big data from healthcare communities‖, IEEE Access. 5 (2017)8869–8879, http://dx.doi.org/10.1109/access.2017.2694446.

[150] N. Downing, A. Cloninger, A. Venkatesh, A. Hsieh, E. Drye, R. Coifman, et al., ―Describing the performance of U. S. hospitals by applying big data analytics‖, PLoSOne 12 (2017) e0179603, http://dx.doi.org/10.1371/journal.pone.0179603.

[151] Y. Wang, L.A. Kung, T.A. Byrd, ―Big data analytics: understanding its capabilities and potential benefits for healthcare organizations‖, Technol. Forecast. Soc. Change (2015), http://dx.doi.org/10.1016/j.techfore.2015.12.019.

[152] J. Roski, G.W. Bo-Linn, T.A. Andrews, ―Creating value in health care through big data: opportunities and policy implications‖