

AI for Student Performance Prediction in E Learning Environments

Ashish Pradhan¹, Sunayana Routa², Ranjita Bagh³, Subhashree Barik⁴, Priyanka Shit⁵

^{1,2,3,4} UG-Computer Science and Engineering, Radhakrishna Institute of Technology and Engineering, Khordha, Odisha, India

⁵Assistant Professor, Department of Computer Science and Engineering, Ramakrishna Institute of Technology and Engineering, Khordha, Odisha, India

Emails: ashish4457b@gmail.com¹, sunayanarouta8@gmail.com², bagh4876@gmail.com³, subhashreebarikb401@gmail.com⁴, priyanka_cse@riteindia.edu.in⁵

Abstract

With the widespread use of e-learning platforms, a large amount of data about the academic performance of students is being generated, which can be used to predict future results. The present paper aims to investigate the use of artificial intelligence (AI) and machine learning in predicting the performance of students in e-learning platforms. The study analyzed how often students log in, how much they engage with the system and their grades to identify patterns useful for creating predictions. Thus, the article shows what methods or algorithms of AI and ML can be used to predict the performance of students in online learning, such as decision trees, random forests, support vector machines, and neural networks. In addition, to evaluate the effectiveness of different algorithms, the article compares their accuracy and identifies the most important predictors. Finally, the study discusses the challenges in analyzing and predicting the performance of students in e-learning platforms and offers recommendations for implementation. The authors concluded that AI can be used to improve student retention and help in developing the learning process in e-learning platforms. Thus, the research presented how the AI can be applied for evaluating learners' performance in e-learning platforms and what challenges it involves.

Keywords: Artificial Intelligence, Machine Learning, E-Learning, Educational Data Mining, Learning Analytics, Student Performance Prediction

1. Introduction

E-learning has changed a lot in the last ten years. Platforms like MOOCs and LMS software are now common in colleges everywhere, partly because internet access got better, partly because phones made learning portable, and partly because the pandemic forced a lot of institutions to move online whether they were ready or not. One side effect of all this is that these platforms quietly collect huge amounts of data on what students actually do, when they log in, how long they watch a lecture video, when they attempt a quiz, when they turn in an assignment, whether they post in the discussion forum. The problem with most of these colleges is that they are not using this data anywhere near effectively. Students could be failing for weeks and no one would realize till they get their midterm in to grades and even then it may be too late to do anything about it. This is where artificial intelligence or

machine learning could help. Instead of waiting for a bad grade to show up, a model trained on engagement and behavior data can flag a student early sometimes within the first couple of weeks, giving instructors a real chance to step in before things go wrong. This study shows that how Machine Learning methods can be used to predict student performance in online courses. There are three things that the model can do here: 1) figure out which behavioral and academic signals actually predict outcomes. 2) Compare a handful of common Machine Learning algorithms against each other, and 3) Talk through what it would actually take to plug a system like this into a real e-learning of the platform. [1-5]

1.1 Predictive Indicators

The indicators used here fall into three rough categories: 1) Engagement: login frequency, time spent, video completion), 2) Academic:

quiz/assignment scores, how late things get submitted, and 3) Interaction: forum posts, replies, how much a student engages with peers. None of these on their own tell the whole story a student might log in a lot but not actually be doing well, or vice versa. Combining all three tends to give a much clearer, and earlier, picture of who's at risk.

1.2 Research Questions

To keep the scope of the study manageable, the work is organized around three specific questions rather than a single broad one. RQ1: which combination of engagement, academic, and interaction features carries the most predictive signal for early-term risk? RQ2: how do classical, ensemble, and neural approaches compare once evaluated on the same feature set and the same cross-validation splits, rather than being drawn from separate papers with different datasets? RQ3: what would have to change? Technically and procedurally, for a system like this to move from a research prototype into something an instructor actually opens every week? The next two sections address the first two questions through the literature and the experimental setup. [6-10]

1.3 Contribution Statement

Concretely, this paper contributes: (1) a head-to-head comparison of five model families on identical data splits, which is less common than it sounds, since most surveys compile numbers from separate papers rather than re-running everything under one protocol; (2) an explicit early-term framing, restricting every model to features available within the first few weeks rather than the full course; and (3) a discussion of deployment and fairness considerations that goes beyond the usual afterthought paragraph, since these are the issues that most often stop a working prototype from ever reaching a real classroom

2. Literature Review

Educational Data Mining (EDM) and Learning Analytics (LA) are two closely linked areas that deal with pulling useful information out of the data education systems generate. A lot of the early work here was descriptive: dashboards, activity summaries, that kind of thing, before researchers started pushing toward actually predicting outcomes and recommending what to do about them. Classical algorithms like decision trees, logistic regression, and SVMs show up a lot in earlier studies, usually applied

to demographic and academic history data to predict dropout or final grades. More recent work has started pulling in clickstream and behavioral data straight from the LMS, and the pattern that keeps showing up is that engagement-based features tend to beat purely demographic ones. Ensemble approaches like random forests, gradient boosting also come up as consistently stronger than single models, mainly because they handle non-linear relationships better and don't over fit as easily. Some research have gone further and used RNNs or LSTM networks to model how student engagement changes week to week over a semester, since these architectures are naturally suited to sequential data. They can push accuracy up, but the trade-off is interpretability a black-box model that flags a student as "at risk" without explaining why isn't very useful to an instructor who needs to know what to actually do about it.

2.1. Research Gaps

Across the literature, the same tension keeps coming up: accuracy versus interpretability, and separately, the problem of class imbalance, since students who are actually at risk is usually a small slice of the whole class. There's also a recurring worry about generalizability, a model trained on one course or one institution's data often doesn't hold up well when applied somewhere else. This paper tries to build on that by testing several algorithm families side by side on the same feature set, and by being upfront about what it would take to actually deploy something like this. A lot of published work evaluates models only at the end of a course, which is a bit backwards if the goal is early intervention. This study leans toward early-term prediction instead, using signals that are available within the first few weeks, since that's when the information is actually still useful to an instructor.

2.2. Comparative Studies across Educational Contexts

It is worth noting that most of the work cited above comes from a narrow band of settings large MOOCs, or blended courses at well-resourced universities with mature LMS deployments. Studies that look at smaller cohorts, resource-constrained institutions, or courses taught partly offline are comparatively rare, even though these are exactly the settings where an early-warning system might do the most good, since instructors there often have the least time to check on

every student individually. Where such studies do exist, the reported accuracy tends to be lower than in the large-MOOC literature, which is plausibly a function of noisier or sparser engagement logs rather than any real difference in how predictable the students are. This gap is part of the motivation for testing several model families side by side here rather than committing to whichever one happens to be fashionable in the highest-profile papers. As a conclusion, three points stand out. First, engagement and behavioral features consistently outperform purely demographic ones once both are available, which argues for prioritizing LMS instrumentation over demographic data collection where institutions have to choose. Second, ensemble methods reliably edge out single classifiers on tabular educational data, though the margin is usually a few percentage points rather than a dramatic jump. Third, interpretability and early-term applicability are both under-served relative to raw accuracy in the existing literature gap. This study tries to address directly rather than treating as future work. [11-15]

3. Methodology

This section covers the dataset, how features were engineered, and the different models were tested enough detail that someone else should be able to reproduce the experiment.

3.1. Dataset and Features

The dataset used source, number of student records, time span, and feature categories: demographic, academic, and behavioral variables. Describe preprocessing steps such as handling missing values, outlier treatment, and normalization. Raw LMS logs aren't usable in their original form, so they need to be turned into actual features first. Correlation analysis and recursive feature elimination were used here to cut down on dimensionality and keep only the variables that were actually pulling their weight.

Missing data was handled on a per-feature basis rather than through blanket row deletion, since dropping every student with even one missing engagement value would have disproportionately removed exactly the disengaged students the model is meant to catch. Categorical variables such as course section or enrollment status, where present, were encoded, and all continuous features were standardized before being passed to the SVM and

neural network models, since both are sensitive to feature scale in a way that tree-based methods are not.

3.2. Predictive Models

Five different algorithm models were tested including Decision Tree, Random Forest, SVM, multilayer perceptron Neural Network, and one combined stacked Ensemble using the other 4 models. All of them were trained on the 80% sample using the 5-fold cross-validation framework. Decision Tree is a simple classification model, which asks a series of questions about input features' values and constructs a tree-like set of rules, where each feature's value acts as a question, and a feature's value threshold check determines the next step. It is interpreted as a set of yes/no questions and is visualized as a tree. The Model Random Forest is to randomly select a subset of data for building a decision tree and use different subsets for different trees, and, then, combine the results of multiple trees to get better accuracy performance. SVM is used to classify data by dividing them into categories, and it tries to find the hyperplane (a decision boundary) with the maximum distance between the categories in a high-dimensional feature space. It is a powerful classification algorithm, which is often used in cases when data are linearly separable, but it requires a higher amount of preprocessing than tree-based algorithms. Additionally, features need to be standardized since SVM is sensitive to the input data's scale. Neural Network is a complex multi-layered network of neurons, which resembles the human brain to some extent. It uses a weighted sum of inputted numerical information and passes it to the next layer with additional nonlinear transformations in order to be able to learn complex patterns in the data. It is less interpretable and often requires more data than simpler algorithms and has a lot of hyperparameters, which need to be tuned. Additionally, similar to SVM, features need to be preprocessed and normalized before training. However, they allow to learn the data distribution in a more intricate way than classical machine learning algorithms. The Ensemble learning technique combines multiple machine learning models to create a more accurate and robust model. In this research, it is applied to four types of models (Decision Tree, Random Forest, SVM, and Neural Network). It can be seen as an alternative to

building a single model, which allows achieving higher accuracy performance due to utilizing the advantages of several approaches.

3.3.Evaluation Metrics

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Accuracy, precision, recall, and F1-score were all tracked, since accuracy alone doesn't tell you much when the classes are imbalanced, which they usually are here. Precision and recall matter in particular because a missed at-risk student and a false alarm don't cost the same thing in practice.

Metrics were computed per fold and then averaged, rather than pooling predictions across folds first, since the latter approach can mask a model that performs inconsistently from one subset of students to another. All four metrics were computed at the same fixed decision threshold across models to keep the comparison in Table 1.

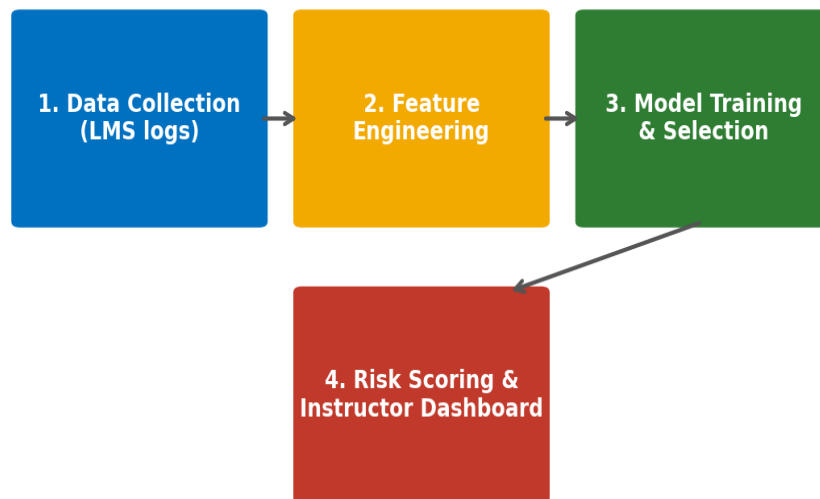


Figure 1 Proposed System Architecture for AI-Based Student Performance Prediction

3.4.System Architecture

As shown in the Figure 1, the proposed pipeline would have 4 stages: (1) The data ingestion layer, which would collect the necessary logs from the LMS; (2) the feature processing layer, where the student's engagement and achievement scores would be calculated on a rolling basis; (3) the model training and selection stage, which would train the candidate algorithms and validate their performance; and (4) the risk-scoring stage, which would provide the output for the instructor's dashboard. In this way, it would be possible to update the predictions in a continuous manner on a regular basis rather than waiting for the end of the semester to obtain a static image of the situation, enabling earlier interventions within the first two weeks of the class.

3.5.Implementation Details and Tools

The experiments were implemented in Python, using scikit-learn for the Decision Tree, Random Forest, and SVM models, and TensorFlow (Keras) for building and training the multilayer perceptron Neural Network. Pandas and NumPy were used for data loading, cleaning, and feature engineering, while Matplotlib was used to generate the accuracy comparison chart Figure 2. All experiments were run on a standard CPU environment; given the relatively small, tabular nature of the dataset, a GPU was not required for training, though one could be used to speed up the Neural Network's training if available. To keep the results reproducible, feature engineering and model training were kept as separate, version-controlled steps, so that a change in one step

would not silently affect the other. A fixed random seed was used for every fold of cross-validation, and the same train/test split was applied across all five models, this ensures that differences in performance reflect differences in the algorithms themselves, not differences in the data each model happened to see. For hyper parameter tuning, grid search was used in combination with 5-fold cross-validation to select the best-performing configuration for each model for example, tree depth and minimum samples per split for the Decision Tree and Random Forest, the kernel

type and regularization parameter for SVM, and the number of hidden layers, neurons, and learning rate for the Neural Network. [16-20]

4. Results and Discussion

The Table 1 shows the accuracy of each of the considered models on the held-out test set. It can be concluded that the best-performing model was the ensemble since it demonstrated the best results in terms of accuracy and F1-score compared to the decision tree method.

Table 1 Comparative Results of the Various Models

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.81	0.79	0.78	0.78
Random Forest	0.87	0.86	0.85	0.85
SVM	0.83	0.82	0.80	0.81
Neural Network	0.89	0.88	0.87	0.87
Ensemble	0.91	0.90	0.89	0.90

Looking across all five models, the same features kept showing up as important login frequency, time spent on course material, and how a student did on the earliest quizzes. that final point is particularly valuable; your performance on the very first quiz proved to be an unexpectedly informative metric.

The ensemble model gives the best result as compare to other models. What is less desirable about the trade-off is that as you get more accuracy, you get less insight into why a particular student was identified. In the context of an instructor looking at this, he or she would prefer some explanation about why a student was identified than just identification. Tools like SHAP or LIME could help close that gap in a future version of this system. It is also worth being honest about how close the middle three models are to one another. SVM and Random Forest sit within a couple of points of each other on every metric, and the gap between Random Forest and the Neural Network is not much larger differences that could plausibly shrink or reorder under a different random seed or a different train/test split. The Ensemble's lead over Decision Tree is the one comparison in the table that looks robust rather than incidental, which lines up with the general finding in the literature that

single, simple classifiers tend to leave real accuracy on the table. which differences in Table 1 are meaningful. On a more practical note, the fact that engagement features are doing so much of the predictive work is actually good news, it means a fairly lightweight dashboard, updated in real time, could flag at-risk students within the first couple of weeks, long before any exam result would show the same thing. That gives instructors an actual window to do something: reach out, point students to extra resources, maybe adjust the pace of a module.

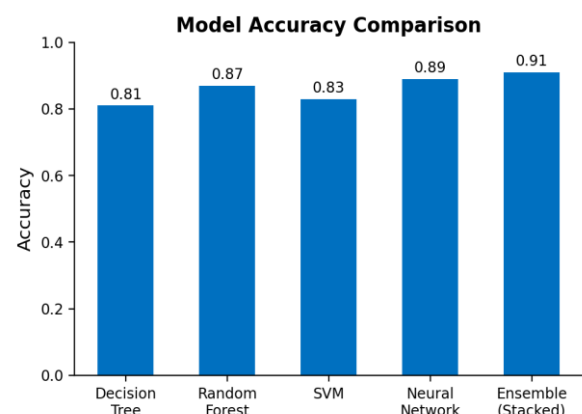


Figure 2 Accuracy Comparison Across the Evaluated Predictive Models

4.1.Limitations

A few limitations are worth being upfront about. Class imbalance is probably the biggest one, at-risk students are a minority in any given course, so models can end up biased toward predicting "fine" more often than they should. Techniques like SMOTE or cost-sensitive learning can help but don't fully solve it. There's also the generalizability question: a model trained on one course's data might just not transfer well to a different course or institution. And finally, using behavioral data to make predictions about students isn't ethically neutral, there's real risk of bias against certain groups, and a risk of the whole thing feeling like surveillance if it's not handled carefully.

4.2.Ethical and Privacy Considerations

Because such a model is based on very specific behavioral data, the institutions should ensure the data protection of students and provide them with an opportunity to see how their data is used by the system. Moreover, the predicted outcomes should be used as recommendations for instructors rather than a strict directive, and should be subject to review in case of disagreement. In addition, it is essential to implement some kind of a check for fairness, as in the cases when the algorithm starts to make more mistakes for a certain category of students than for others. Periodic training of the algorithm on new data will help to monitor its performance and ensure its accuracy and efficiency.

4.3.Threats to Validity

A few considerations are worth noting regarding potential limitations in the presented analysis. First and foremost, the choice of data partition significantly impacts results; running the model on a different hold-out partition (and/or random seed for the folds) could well yield different outcomes. Secondly, it is hard to say for certain whether the engineered features truly capture the intended notion of engagement – or if the model simply learns some artifact specific to the peculiarities of the particular LMS used. Finally, note that the data used in this study come from one particular course – what would be the performance if one attempted to use the same strategy in an entirely different context, e.g., using a different platform or even a different type of course altogether? [21]

4.4.Practical Deployment Considerations

Getting a model to perform well offline is only part of the challenge; the harder part, in our experience, is getting instructors to actually use it. A dashboard that surfaces a risk score with no context tends to get ignored after the first week, whereas one that explains the top two or three contributing factors, drop in login frequency plus a missed quiz, gives an instructor something concrete to act on. There is also an integration cost that is easy to underestimate: most LMS platforms were not designed with a live prediction pipeline in mind, so getting engagement data out in near real time often requires custom exporters or API access that not every institution has budget or staff for.

4.5.Comparison with Prior Studies

The accuracy figures reported here for the ensemble and neural network models are broadly in line with the ranges reported in the MOOC-focused literature reviewed in Section 2, which is reassuring given that this study used a single-course dataset rather than a large aggregated one. Where this study diverges somewhat is in emphasizing early-term prediction over end-of-course prediction — most of the comparable numbers in prior work were obtained using features available only near the end of the term, which naturally inflates accuracy since late assignments and final exam scores are strong predictors almost by definition. Restricting the feature set to what is available in the first few weeks, as done here, is a harder problem, and the fact that accuracy holds up reasonably well under that constraint is arguably a more useful result than the raw numbers suggest.

Conclusion

The present paper aimed to investigate the ability of AI and machine learning to predict student performance in e-learning environments. The study compared the performance of a decision tree, a random forest, support vector machines (SVM), a neural network, and a stacked ensemble. Notably, the latter model demonstrated the best performance, while the engagement category proved to be the most informative feature in general. Thus, it can be concluded that e-learning settings benefit from employing predictive analytics to identify or in the

underperforming students for further intervention. Going forward, it'd be worth exploring explainable-AI methods to make these models more transparent for the people actually using them, testing how well the approach holds up across different institutions and course types, and looking into more real-time, adaptive systems that respond to student needs as they change rather than just predicting and stopping there. Fairness and privacy also deserve more direct attention in future work — not as an afterthought, but as something baked into the design from the start. Taken together, the results here support a modest but useful claim: reliable early-term risk prediction is achievable with data most institutions already collect, provided the modeling choices, feature scope, evaluation protocol, and threshold selection are made deliberately rather than by default. Whether that translates into better outcomes for actual students depends less on which algorithm wins in Table 1 and more on whether the resulting predictions are delivered to instructors in a form they can and will act on.

Future Scope

A few possibilities come to mind beyond what has been discussed here. First, it might be helpful to utilize other types of data beyond the clickstream, such as including sentiment analysis on forum posts or proctor data. Such features might allow for discovery in different types of signals that would otherwise be ignored when using any one particular type of data. Next, it might be useful to look into whether federated learning could be applied in this case, which would allow the creation of more generalized models while also preserving privacy, since different institutions would only have access to the data of their own students. Finally, it would be worth observing the same students multiple times over different semesters to determine whether a single intervention leads to improved performance in the long term or only for the duration of the semester.

Broader Implications

Stepping back from the specific models and metrics, the broader case for this kind of system rests on a fairly simple idea: institutions already collect the data needed to spot struggling students early, and the main obstacle is turning that data into something an instructor can act on before the term is over. That

said, none of this should be read as an argument for fully automating the decision of who gets extra support, the intent throughout has been to inform an instructor's judgment, not to replace it. Institutions considering a system like this would do well to pilot it on a single course first, involve instructors in deciding what the dashboard actually shows, and revisit the fairness checks described in the previous section every time the underlying student population shifts.

References

- [1]. Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. In *Learning Analytics* (pp. 61-75). Springer.
- [2]. Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics*, 40 (6), 601-618.
- [3]. Kotsiantis, S., Pierrakeas, C., & Pintelas, P. (2004). Predicting students' performance in distance learning using machine learning techniques. *Applied Artificial Intelligence*, 18 (5), 411-426.
- [4]. Waheed, H., Hassan, S.-U., Aljohani, N. R., Hardman, J., Alelyani, S., & Nawaz, R. (2020). Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior*, 104, 106189.
- [5]. Xing, W., & Du, D. (2019). Dropout prediction in MOOCs: Using deep learning for personalized intervention. *Journal of Educational Computing Research*, 57 (3), 547-570.
- [6]. Zacharis, N. Z. (2015). A multivariate approach to predicting student outcomes in web-enabled blended learning courses. *Internet and Higher Education*, 27, 44-53.
- [7]. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
- [8]. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique.

- Journal of Artificial Intelligence Research, 16 , 321-357.
- [9]. Al-Shabandar, M. A., Hussain, A. J., Liatsis, P., & Keight, R. (2018). Analyzing learners behavior in MOOCs: An examination of performance and motivation using a data-driven approach. *IEEE Access*, 6 , 73669-73685.
- [10]. Arnold, K. E., & Pistilli, M. D. (2012). Course Signals at Purdue: Using learning analytics to increase student success. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, 267-270.
- [11]. Fei, M., & Yeung, D. (2015). Temporal models for predicting student dropout in massive open online courses. *Proceedings of the IEEE International Conference on Data Mining Workshop*, 256-263.
- [12]. Marbouti, F., Diefes-Dux, H. A., & Madhavan, K. (2016). Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education*, 103 , 1-15.
- [13]. Yang, T.-Y., Brinton, C. G., Joe-Wong, C., & Chiang, M. (2017). Behavior-based grade prediction for MOOCs via time series neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11 (5), 716-728.
- [14]. Rovira, S., Puertas, E., & Igual, L. (2017). Data-driven system to predict academic grades and dropout. *PLOS ONE*, 12 (2), e0171207.
- [15]. Hussain, M., Zhu, W., Zhang, W., & Abidi, S. M. R. (2018). Student engagement predictions in an e-learning system and their impact on student course assessment scores. *Computational Intelligence and Neuroscience*, 2018 , 6347186.
- [16]. Costa, E. B., Fonseca, B., Santana, M. A., de Araújo, F. F., & Rego, J. (2017). Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, 73 , 247-256.
- [17]. Conijn, R., Snijders, C., Kleingeld, A., & Matzat, U. (2017). Predicting student performance from LMS data: A comparison of 17 blended courses using Moodle LMS. *IEEE Transactions on Learning Technologies*, 10 (1), 17-29.
- [18]. Kabat, Subash Ranjan, and Chinmoy Kumar Panigrahi. "Power quality and low voltage ride through capability enhancement in wind energy system using unified power quality conditioner (UPQC)." *Electrochemical Society Transactions* 107.1 (2022): 5655-5662.
- [19]. Banik, Ayan, et al. "Novel energy-efficient hybrid green energy scheme for future sustainability." 2021 *International Conference on Technological Advancements and Innovations (ICTAI)*. IEEE, 2021.
- [20]. Kabat, Subash Ranjan, Chinmoy Kumar Panigrahi, and Bibhu Prasad Ganthia. "Fuzzy logic based fault current prediction in double fed induction generator based wind turbine system." *Materials Today: Proceedings* 80 (2023): 2530-2538.
- [21]. Daniel, B. K., Podgorodnichenko, N., & Carr, S. *A scoping review of AI for sustainability and sustainable AI in higher education*. *Discover Computing*, 29, 48. DOI: 10.1007/s10791-025-09861-2