A Personalised Investment Portfolio Recommendation System Using a Genetic Algorithm Optimised Neural Network

Aaron Success Adaego*, Nathaniel Ojekudo and Kingsley Igulu

Department of Computer Science, Faculty of Natural and Applied Sciences Ignatius Ajuru University of Education, Rumuolumeni, Port Harcourt, Rivers State, Nigeria

***Corresponding Author**

DOI: https://doi.org/10.51244/IJRSI.2026.1311000012

Received: 07 November 2026; Accepted: 12 November 2026; Published: 19 November 2026

ABSTRACT

Personal financial advisory tools need to connect users' financial circumstances with understandable recommendations. Static decision rules offer consistency but may not capture interactions among income, expenditure, savings and risk preferences. This study presents a web-based personal financial advisory prototype that combines an artificial neural network with genetic algorithm optimisation. An object-oriented approach separates user management, financial profiling, model processing, recommendation generation and reporting. The implementation is described using Python for analytical processing, PHP and Laravel for server-side functions, React and JavaScript for the interface, and MySQL for persistent storage. Financial inputs include income, expenses, savings and self-reported risk appetite. Demonstration cases show the movement of these inputs through the application to advice categories covering savings and bonds, balanced funds, and stocks and exchange-traded funds. The development evaluation reports classification accuracies of 91.8% for the hybrid model and 84.3% for a conventional neural network, together with a mean squared error of 0.032. These figures remain preliminary because dataset counts are inconsistent and the exact data source, test partition, model configuration and prediction-level results are not fully documented. Illustrative distributions and functional examples therefore support an account of prototype behaviour rather than a verified claim of investment suitability or financial benefit. The study contributes a modular advisory architecture and identifies the evidence needed for reproducible evaluation. Further work should reconcile data provenance, establish independent reference labels and evaluate performance, usability and financial suitability under a documented protocol.

Keywords: personal financial advice; genetic algorithm; neural network; recommendation system; object-oriented design

INTRODUCTION

Financial advice for an individual must have a link between the individual's assets, finances, objectives, and risk tolerance. While an app may have information on the individual's income and expenses, it is unlikely to provide much help if the advice provided does not take into consideration the circumstances under which funds are required. This makes the relationship between the information about the individual and the advice provided an important design feature of automated advice services [1] [[10] [11].

Financial literacy is also pertinent to the construction of such systems. In linking financial literacy with decision making, Lusardi emphasizes the need for financial education that caters to the diversity of its users [2]. This implies that the collection of numeric input is not the only objective for a financial advice system. It also involves users being able to comprehend the advice and its underlying information constraints.

This paper deals with the development of a personal finance advisory system for budgeting, saving and general investment advice. It is motivated by the need to advise users whose incomes and expenses may vary, such as in the Nigerian scenario mentioned in the project. But the existing evaluation does not prove whether the

development data belong to the Nigerian population. Hence, the setting where the system is supposed to be used needs to be distinguished from the validated population.

Neural networks are capable of modeling non-linear relationships between financial variables and output classes. Evolutionary optimization represents one method for searching for good parameter settings of the neural network using a population of potential solutions. The broader concept of neuroevolution is explained by Stanley et al., where evolutionary algorithms are capable of optimizing both the weights of the neural network and other aspects of the design [3][12], [13], [14]. Genetic algorithms provide the means for selecting, crossing over and mutating solutions in a search space, although their success relies on certain features [4].

The aim is to build a web-based advisory system prototype based on a neural network optimized by a genetic algorithm. Objectives include organizing financial information in an object-oriented approach, linking the analytical model to a useful web application, making recommendations based on user profiles, and evaluating the results provided. The contribution comes in the form of combining financial profiling, hybrid model processing, and recommendations into a single framework. This is an application contribution and not an innovation in evolutionary learning algorithms.

The proven output of the system is categories of advice and brief statements of recommendations. These are not portfolio weights at the level of individual assets. It is the difference between the scope of the article and its prototype that enables the generation of personal financial advice recommendations, but not portfolio optimisation and investment performance.

LITERATURE REVIEW

Automated advice and user interaction

According to Jung et al., robo-advising represents the digitalization and automation of financial advising services [1]. This definition gives a good basis for considering an advice system as an interaction between the user's data and a decision support process. It is not enough to consider every automated advice system as personalised enough for one particular person.

Another study by Jung et al., which focuses on the design of a robo-advisor for risk-averse people with low budgets, has come up with principles of design [5]. The controlled lab experiment highlights the importance of usability, transparency, and the willingness of users to be engaged in investment decisions. This can be linked to the current prototype in terms of its questionnaire and history module. Nevertheless, the presence of the interface does not mean that the user understands its recommendations.

Evolutionary learning and financial recommendation

Another study carried out by Jung et al. and concerning the development of a robo-advisor for risk-averse individuals with small budgets has produced some design principles [5]. The experiment conducted in laboratory conditions proves the significance of usability, transparency, and users' readiness to participate in making investment decisions. It is possible to relate these principles to the existing prototype in regard to the questionnaire and history modules. However, the existence of an interface does not imply the users' understanding of the recommendations provided.

The method described by Flores-Fernandez et al. consists of combining long short-term memory neural networks with a genetic algorithm in order to optimize investment portfolios [6]. The research divides the stock price modeling problem from the portfolio allocation search and mentions the configurations tested in their study. It provides an example of how the prediction of neural networks and the optimization of evolutionary algorithms can be applied in finance, but the data and goal of their research are not the same as in this prototype.

Genetic algorithms, their operators, and fields of application are reviewed by Katoch et al. [4]. This review justifies the usage of evolutionary search as one of the options for design, but makes the selection of the

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XI November 2026

representation and fitness function important as well. The applicability at the application level should be proved by showing that the search adds value to the task performed sufficiently to cover its computational expenses.

Ojekudo [15] developed a neuro-genetic fuzzy system for predicting students' academic performance, integrating neural-network learning with genetic optimisation and fuzzy reasoning. Although the study addresses educational prediction rather than personal finance, it provides a related example of hybrid computational intelligence applied to decision support. Its reported assessment includes comparative measures of user satisfaction, but does not establish investment suitability or demonstrate predictive gains for financial advice. Accordingly, the present study adopts the hybridisation principle while recognising the need for separate, reproducible evaluation using financial recommendation data.

Transparency and the evaluation gap

The financial dashboard can present the reported numbers and recommendation history without providing any reasoning as to why the model decided on a specific category. Arrieta et al. identify various approaches for explainable artificial intelligence and link explainability to responsible design [7]. In this case, what matters is that an explanation is faithful to the decision of the model and is useful to the user in evaluating its relevance. The graph of advice frequencies is not enough for that.

Classification is not the only aspect that defines the problem of designing a system. The design requires consistent input, target for recommendations, analysis procedure and output presentation. This research brings all these aspects to prototype stage. However, its empirical value should be evaluated in light of the completeness of data and training/testing described below.

MATERIALS AND METHODS

Study design and system scope

Object-oriented analysis and design techniques were used for the project. The process included requirement analysis, entity-relationship identification, interface/database design, analysis processing, and functional demonstration. The key responsibilities were assigned to user management, financial profile, neuro-genetic model, saved parameters of the model, advisory processing, and output generation. This classification provides a structure for future maintenance of the system and for the way a financial proposal goes through to the output interface.

The system includes elements that enable registration and login, the financial questionnaire, the presentation of advice, and a history of previous submissions. Detailed descriptions include functionalities like transaction analysis, debt management, and live market information; but the recorded demonstrations deal only with questionnaire data entry and types of recommendations. The latter functionalities are mentioned as design objectives rather than performance achievements. No claim is made concerning ongoing retraining, live market adjustments, or transaction execution.

Architecture and implementation

Figure 1 shows the object-oriented design. Input financial data from a user is used to create a financial profile. The analysis model uses a combination of the artificial neural network and genetic optimization tool. The parameters obtained from the modeling process form the basis of recommendation generation, while the reporting component relays the recommendations.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XI November 2026

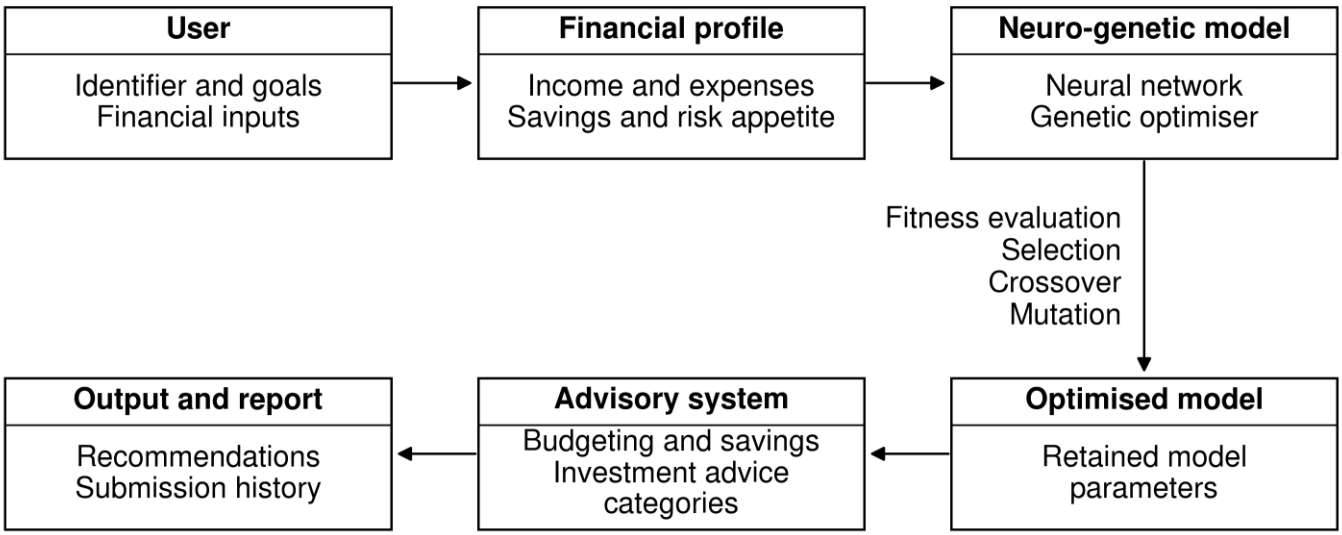

Figure 1. Object-oriented structure of the advisory system.

The implementation is described as a locally deployed web application. Python performs analytical processing; PHP with Laravel handles server-side application functions; HTML, CSS, JavaScript and React support the interface; and MySQL stores structured records. Apache and XAMPP provide the local server environment. These are the reported development technologies. Exact dependency versions and a complete executable training environment are not established by the available implementation records.

Table 1 identifies the principal data entities. The user identifier links personal records, transactions and financial profiles. In the demonstrated workflow, the user enters monthly income, monthly expenses, current savings and a risk level. Risk appetite is represented by low, medium and high categories. These labels capture self-reported preference; they do not independently measure financial capacity to absorb loss.

Table 1. Principal data entities and their roles

Entity	**Representative fields**	**Role**
User	User identifier, name, email, risk appetite, financial goal	Identifies the user and records profile preferences.
Transaction	Transaction identifier, user identifier, type, amount, category, date	Records income and expenditure entries in the system design.
Financial profile	Profile identifier, user identifier, total income, total expenses, savings, disposable income	Organises financial attributes for advisory processing.

Financial data and provenance

The development dataset consists of accounting data collected from Kaggle and supplemented with synthetic or augmented data. The variables recorded include income, expenses, savings, debts, investments, financial objectives, and risk appetite. It is important to note that the name of the dataset, its owner, URL, version, license, and collection date are unknown. Furthermore, the extent of contribution of the collected data versus the synthetic data is not specified either. Therefore, Kaggle is classified as a source citation in the context of the current research.

The preprocessing procedures include duplication elimination or inconsistency correction, missing value treatment, normalization of numeric variables, and categorical variables coding. In one particular

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XI November 2026

implementation, numerical attributes are scaled to the range of [0, 1], while risk level is coded as low = 0, medium = 1, and high = 2. However, details about the technique used for missing value replacement, the fitted scale parameters, and the order of preprocessing operations relative to data splitting are not reported. This is important since leaking information between training and testing stages results in biased evaluation results; Kapoor and Narayanan address this problem in general terms in their study on machine learning [8].

In this case, there is no way to confirm or deny leakage from the data provided. Specifically, generated or augmented examples created using the same profile cannot be considered independent. A repeatable evaluation will require identifying the original examples, grouping related examples into the same partition, and keeping any final test set separate from the model selection process. This is related to the next stage of validation and not something that has already been done in this work.

Hybrid model and recommendation workflow

The implementation section defines the genetic algorithm in terms of the search on potential configurations of weights of the neural network. Every configuration is assessed based on the prediction fitness function, and the chosen configurations are subject to crossover and mutation until a termination criterion is met. The obtained weights are saved to produce advice. This represents the weight optimization approach adopted in the article.

Elsewhere in the design, chromosomes are represented as financial strategies or allocations where fitness is measured by profitability, risk, and financial viability. This design cannot be said to be similar to weight optimization and does not have enough details for replicating a constrained portfolio optimizer. There are no details about the network structure, activation functions, output representation, fitness formula, population size, number of generations, selection strategy, crossover probability, mutation probability, random number generator, and stopping criteria. Therefore, the design represents the intended processing mechanism and not an algorithm.

At application level, the process starts with authentication and completion of the questionnaire. The server receives the financial information provided, keeps the submission, and sends the relevant variables to the analysis process. The recommendation system converts the results of the analysis into a text message which is presented through the interface. Past submissions can be obtained from the history table. It is important to note that this process is different from offline training since no information is provided about the training of the neural network for each request.

Evaluation measures and evidence boundaries

The functional demonstration shows how well the submission is stored, appropriately referenced to the correct user, displayed as guidance, and retrieved from the record history. The recorded demonstrations show input-output examples and not an automated log of tests conducted. These examples demonstrate the expected performance of the program without proving any error rates when used concurrently and in abnormal situations.

The predictive evaluation is described by classification accuracy and mean squared error. In the case of the three-class advisory scheme, classification accuracy is defined as the ratio between correctly classified test data and the total number of test data, in percent form. The consistent multiclass evaluation will also include class support, confusion matrix, and per-class precision, recall, and F1-score measures. Saito and Rehmsmeier demonstrate the importance of the precision-recall measure in the case of imbalanced binary evaluation, and their results warn us against the interpretation of one aggregated measure; however, their binary setting is different from our three-class problem.

The information regarding the test fraction, the numbers of training and test samples, the numbers of repeated runs, the independent labeling process, and the level of predictions outputs is missing. Likewise, there is no information about the target scale or output form in connection with the mean squared error reported. Thus, no confidence interval, significance test, or replicable comparison can be performed. Even though the timing information is clearly stated to be an estimate, it is not regarded as a performance measurement.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XI November 2026

Security and research governance

These include authentication, password protection, input validation, data transfer security, restricted database access, and backups. These are architectural requirements. No security assessment has been conducted to prove that these are implemented effectively. As a result, no framework or database is considered to be deployed securely based on the findings of this paper.

The methodology mentions interviews and questionnaires during requirements gathering, but participant numbers, consent procedures and any ethics determination are not reported. The questionnaires shown in the application also serve as financial input forms and should not automatically be interpreted as a separate research survey. The distinction between requirements participants, demonstration records and externally sourced data requires clarification before conclusions about human-subject data collection or population coverage can be made.

RESULTS

Prototype functions and illustrative advice

The stated prototype provides a financial questionnaire with inputs of monthly income, monthly expenditure, savings, and risk appetite. A page recommending suggestions is provided while another page shows the number of past submissions. Table 2 contains three demonstrated cases in question. The cases lack a currency specification, meaning that the values in terms of finance cannot be referred to as naira.

Table 2. Documented illustrative input–output cases

Case	Income	Expenses	Risk	Displayed advice category
1	4,000	2,500	Medium	Mutual funds
2	3,000	2,800	Low	Savings accounts and bonds
3	6,000	5,000	High	Stocks and exchange-traded funds

Note: Monetary units are unspecified. Savings values are not reported in this set of demonstration cases.

These cases show the intended connection between the submitted profile and the general category of recommendation, as well as the structure needed to be able to connect the output to its input. Nevertheless, the advice provided in the references does not have any independent expert assessment. Therefore, there are three matching cases that demonstrate the behavior of the workflow, but not an estimation of generalization ability or the suitability of the advice for a real investor.

Illustrative savings and advice summaries

The distribution of savings is provided in Table 3. There are 100 savings in total. Sixty are between 500 and 2,000 units, while eight are above 4,000 units. This distribution applies only to the one described in the summary, and cannot be used to decide the savings distribution in a sample population or the class distribution in the test dataset.

Table 3. Illustrative savings distribution

Savings range in unspecified units	Records	Percentage
500–1,000	32	32%
1,001–2,000	28	28%

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XI November 2026

2,001–3,000	18	18%
3,001–4,000	14	14%
Above 4,000	8	8%
Total	**100**	**100%**

Note: Percentages are calculated from the illustrative table total of 100. This is not a verified training or test sample size.

Table 4 contains the discrete frequency distribution of advice, where the total adds up to 100. There are 40 observations for savings and bonds, 35 for balanced mutual funds, and 25 for stocks and ETFs. These categories refer to the outputs contained in the table and must not be interpreted as actual investments made by the user or as products being offered.

Table 4. Illustrative frequency of advice categories

Advice category	**Records**	**Percentage**
Savings accounts and bonds	40	40%
Balanced mutual funds	35	35%
Stocks and exchange-traded funds	25	25%
Total	**100**	**100%**

Note: This is a descriptive output summary. Its relationship to the final model evaluation dataset is not established.

These summary tables have been retained for purposes of illustration, since the figures that appear in them can be independently verified. They are not included in the risk by advice crosstabulation, which has a total of 110, nor are they used in order to estimate missing test results. In order to determine whether these different summaries come from different runs or transcriptions, access to the records themselves would be necessary.

Reported predictive scores

The overall results for the development evaluation are displayed in Table 5 below. The hybrid model is able to achieve an accuracy of 91.8%, while the conventional artificial neural network attains an accuracy of 84.3%, hence a difference of 7.5%. The mean squared error for the hybrid model is 0.032, but there is no base line value for comparison..

Table 5. Preliminary reported model scores

Measure	**Hybrid model**	**Conventional neural network**	**Interpretation**
Classification accuracy	91.8%	84.3%	Test counts, split membership and prediction-level outputs are not documented.
Mean squared error	0.032	Not reported	Target scale and output representation are not specified.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XI November 2026

Accuracy difference	7.5 percentage points	Reference comparator	Arithmetic difference only; uncertainty and significance cannot be determined.

Note: Values are retained as reported; they were not reproduced through new training. Approximate timing comparisons are excluded from measured results.

The available information does not support the assertion that both approaches used the same independent testing set or a similar budget for tuning their parameters. The denominator of the percent values presented is not known. Therefore, the difference of 7.5 percent cannot be considered a statistically significant gain. In addition, the value of 0.032 of the mean squared error has no meaning without a proper definition of the objective function.

There is no proof from generation-by-generation logging or a detailed baseline for the assertions of rapid convergence and low error. The independent measurement of speed is described as approximate and not measured using a standard benchmark; therefore, it does not prove low latency or response time or scalability.

DISCUSSION

What the prototype demonstrates

The most well-documented contribution relates to the design of an advisory application based on a financial profile and the history of recommendations. The separation of functionalities, including user management, data analysis processing, recommendation generation, and reporting, creates a well-defined structure for the application. The separation also clearly defines the inputs, the output of the model, and the display of recommendations. This type of system design is beneficial for implementation and evaluation purposes, although modularity alone does not define the quality of the model.

This research has shown the reason why recommendation and portfolio optimization should be evaluated separately. The system that generates the label “balanced mutual funds” only chooses a category without defining the asset allocation, applying allocation restrictions, or showing a good risk/return tradeoff. Flores-Fernandez et al. solve a different problem using prediction and then portfolio optimization [6]. The current prototype is unable to achieve the performance of a portfolio matching the questionnaire profile and category label.

Interpreting accuracy and generated labels

Accuracy is informative only once the prediction problem and labeling criteria have been specified. In the case of deterministic labeling, the model might be able to learn the rules but add little value besides their straightforward implementation. This represents a valid evaluation of rule emulation, but one that answers a different question than whether the recommendation improves financial decision-making. Therefore, the rule-based benchmark is especially relevant for this use case.

The size of the sample will determine how the accuracy percentage is to be interpreted and its level of uncertainty. On the other hand, the partitioning process determines whether the test contains any unseen cases. The generated data can increase the size of the database without adding any additional information. Therefore, the scores must be seen as a starting point for validation.

Kapoor and Narayanan show that leakage and non-reproducibility may undermine robust results in machine learning [8]. There is no proof at this point of leakage. What is known is that an external reviewer cannot assess some of the relevant measures due to lack of information. To address this problem, it is necessary to know the true lineage of the data, the splits, and predictions; changing the way the scores are presented will not solve the problem.

Recommendation suitability and user understanding

Low, medium, or high risk appetite is merely a partial characterization of the financial suitability of a user. For example, a user’s ability to withstand volatility is not necessarily the deciding factor as to whether money can

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XI November 2026

be kept invested when there are payments that need to be made. The current examples do not take into account the investment horizon, liquidity requirements, product costs, or the consequences of giving bad advice.

The history module provides traceability; but the way previous guidance is presented does not show whether the user understands it. The lab study carried out by Jung et al. shows how usability can be studied as a standalone design goal [5]. In the case of this prototype, some of the relevant issues would include whether the user understands the input fields, whether the user understands the categories, and whether the user understands when a change in financial situation makes previous guidance obsolete. No participant-based usability findings are provided here.

Explanation of models requires more than just graphical representation alone. Arrieta et al. study explanation approaches and their connection with responsible artificial intelligence [7]. Here, explanation must be able to reflect the real decision process and help a user make an assessment of the recommendation. The current graphs show distributions, and the short recommendations are outputs. Both are not a form of explanation.

Limitations and further evaluation

The key limitations include issues related to provenance, reproducibility, and the difference between demonstrating and measuring. There is no specific dataset used, there is no final number of samples that was reconciled, there is no configuration for the hybrid model, and there is no proof at the prediction level regarding the metrics. The code provided is meant to demonstrate how generated data and visualization can be done, not the whole process of training.

The next evaluation should begin by recording the separate sources and generated data, the permissions required, the feature schema of the last step, and the justification for each reference designation. The next step would be to formulate the prediction goal and determine if the genetic algorithm is optimizing model parameters or portfolio allocations. If it is optimizing model parameters, replicability will require an executable training procedure, fixed partition assignment, search configuration documentation, and comparable baseline procedures.

The comparison must involve the majority class baseline, deterministic advice rules, a regular neural network, and the hybrid classifier, all evaluated against the same independent test set. Repeating the process multiple times will reveal the variance associated with the evolutionary optimization process. The class-specific errors need special attention because the average accuracy can hide the poor treatment of a less frequent class. Independent financial audit will establish the validity of the recommendation, whereas user testing will measure understanding and usability. Neither of these characteristics can be deduced from the classification accuracy.

The latency and response time have to be measured directly in a certain environment and under a certain workload. An assertion of adaptability needs to be tested through observation in different input or time conditions, whereas the assertion of cost-effectiveness requires an appropriate outcome study. These are two separate issues. The current research provides a basis for designing such evaluations.

CONCLUSION

The proposed design of a personal financial advisory system comprises the following elements: user information, financial profile, neural network architecture learned by genetic algorithms, advice creation, and reporting. The user interface and the example data show how financial data are turned into general advice and historical data are retrieved. The system can be viewed as a prototype of a financial advice system and not as an optimizer of an asset portfolio.

The accuracy levels of 91.8% and 84.3% point towards a possible discrepancy between the hybrid approach and its neural network counterpart. The mean squared error also needs a clear target for proper analysis. The most urgent task for further research is to resolve the data problem, determine the chosen algorithm, and perform the same experiment again. The claims related to suitability, scalability, adaptability, and cost-effectiveness need further confirmation.

REFERENCES

1. D. Jung, V. Dorner, F. Glaser, and S. Morana, "Robo-advisory: Digitalization and automation of financial advisory," Business & Information Systems Engineering, vol. 60, no. 1, pp. 81–86, 2018, doi: 10.1007/s12599-018-0521-9. [Online]. Available: https://doi.org/10.1007/s12599-018-0521-9
2. A. Lusardi, "Financial literacy and the need for financial education: Evidence and implications," Swiss Journal of Economics and Statistics, vol. 155, Art. no. 1, 2019, doi: 10.1186/s41937-019-0027-5. [Online]. Available: https://doi.org/10.1186/s41937-019-0027-5
3. K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," Nature Machine Intelligence, vol. 1, no. 1, pp. 24–35, 2019, doi: 10.1038/s42256-018-0006-z. [Online]. Available: https://doi.org/10.1038/s42256-018-0006-z
4. S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," Multimedia Tools and Applications, vol. 80, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6. [Online]. Available: https://doi.org/10.1007/s11042-020-10139-6
5. D. Jung, V. Dorner, C. Weinhardt, and H. Pusmaz, "Designing a robo-advisor for risk-averse, low-budget consumers," Electronic Markets, vol. 28, no. 3, pp. 367–380, 2018, doi: 10.1007/s12525-017-0279-9. [Online]. Available: https://doi.org/10.1007/s12525-017-0279-9
6. G. A. Flores-Fernandez, M. Jimenez-Carrion, F. Gutierrez, and R. A. Sanchez-Ancajima, "Genetic algorithm and LSTM artificial neural network for investment portfolio optimization," Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 15, no. 2, pp. 27–46, 2024, doi: 10.58346/JOWUA.2024.I2.003. [Online]. Available: https://doi.org/10.58346/JOWUA.2024.I2.003
7. A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information Fusion, vol. 58, pp. 82–115, 2020, doi: 10.1016/j.inffus.2019.12.012. [Online]. Available: https://doi.org/10.1016/j.inffus.2019.12.012
8. S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," Patterns, vol. 4, no. 9, Art. no. 100804, 2023, doi: 10.1016/j.patter.2023.100804. [Online]. Available: https://doi.org/10.1016/j.patter.2023.100804
9. T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," PLOS ONE, vol. 10, no. 3, Art. no. e0118432, 2015, doi: 10.1371/journal.pone.0118432. [Online]. Available: https://doi.org/10.1371/journal.pone.0118432
10. K. T. Igulu, T. P. Singh, F. E. Onuodu, and N. S. Agbeb, "A marketing approach to recommender systems," Recommender systems, pp. 105–120, 2023.
11. N. R. Saturday, K. T. Igulu, T. P. Singh, and F. E. Onuodu, "The Role of Machine Learning/AI in Recommender Systems," in Recommender Systems: A Multi-Disciplinary Approach, 2023. doi: 10.1201/9781003319122-5.
12. K. T. Igulu, D. D. Wisdom, M. O. Arowolo, and T. P. Singh, "Computational intelligence in big data analytics," in Computational Intelligence in Industry 4.0 and 5.0 Applications, Auerbach Publications, 2025, pp. 265–328.
13. K. T. Igulu, Z. P. Piah, and P. O. Asagba, "An Investigation into the Conceptual Controversies between Artificial Intelligence and Computational Intelligence," Afr. J. Comput. Ict, vol. 8, no. 2, 2015.
14. E. Okorafor et al., "Revolutionizing Multi-agent Systems: The Role of Agentic RAG in Dynamic Data Ingestion and Real-Time Reasoning," in International Conference on Computational Science and Computational Intelligence, 2024, pp. 158–174.
15. N. A. Ojekudo, "A neuro-genetic fuzzy system for the prediction of student's academic performance," Asian Journal of Multidimensional Research, vol. 11, no. 10, pp. 214–232, 2022, doi: 10.5958/2278-4853.2022.00269.5. [Online]. Available: https://doi.org/10.5958/2278-4853.2022.00269.5