AJEMA GLOBAL RESEARCH NETWORK

AMERICAN Journal of Engineering, Mechanics and Architecture
Volume 4, Issue 5, 2026 ISSN (E): 2993-2637

Python-Based Slot Machine Game Simulation using Randomization and Control Structures

S. Ramesh Kumar
Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India
Email: rameshkuma@gmail.com

M. Mohamed Thariq
Department of Computer Science and Engineering, Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India

A. Mohamed Fahadhu
Department of Computer Science and Engineering, Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India

T. Shynu
Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India

M. Mohamed Sameer Ali
Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India

Abstract: The Python Slot Machine project is a simple and interactive console based application that simulates the working of a classic casino slot machine using python programming. The main purpose of this project is to show and use fundamental programming concepts in a fun and practical way. This reinforces the concepts of loops, conditional statements, functions, random number generation, and user input/output handling in a game-like environment, which makes learning fun through interaction. In this game the user has a predefined balance and can make virtual bets before spinning the slot machine. Each spin will create a random combination of symbols through the use of randomisation features in Python. The outcome of each spin is evaluated according to pre-determined rules, where combinations of symbols result in payouts, and other combinations result in the loss of the bet placed. The game is on a loop so the user can play several rounds until he decides to stop or runs out of balance. The system has important features such as input validation to only accept valid bet amounts, a balance tracking mechanism to keep track of the player's credits, and a clear win/lose logic to determine the outcome of each spin. These features make the game play smoothly and prevent invalid inputs from breaking the program flow. Overall, the Python Slot Machine project is a simple entertainment application and an educational tool. It provides hands on experience with basic python concepts and also introduces you to basic game development logic which is a nice stepping stone towards more advanced programming and interactive game development projects.

Keywords: Slot Machine, Advanced Game, Python Programming, Loops, Conditionals, Functions, Randomization, Demonstrate Core Programming.

1. Introduction

This project aims to create and deploy a basic slot machine game in the Python programming language [48]. The goal is to replicate the fundamental aspects of a traditional casino slot

machine in a simplified console form. Slot machines are popular types of gambling whose results are mainly affected by chance and whose results are determined by the random combination of symbols generated every time the reels are spun [57]. Their randomness by nature makes them a perfect model to demonstrate core programming concepts like probability, random number generation, control structures, and game logic. For this project, we leverage Python's native capabilities, particularly its random module, to mimic this erratic behaviour in a controlled digital environment [37]. The motivation of this project is to build an interactive and engaging application, but also to improve your understanding of fundamental programming concepts. Building a working game from scratch allows us to get a glimpse of how logic is put into practice in real applications [51]. This project is built on the importance of loops, conditional statements, functions, and user interaction, which are the basic building blocks in software development. users are subjected to the repetition of logic execution, decision making based on conditions, and changing game states like balance and outcomes while playing continuously while performing the game

In this implementation the user is allowed to place virtual bets and spin the slot machine which produces a random set of symbols on each of the reels [43]. The result of each spin is based on the combination of symbols that appear. The user wins or loses virtual credits according to predetermined rules. The system is based on the concept of reward logic. If the user makes a successful combination, he gets a payout, if not, then his balance is reduced [56]. This type of mechanism helps demonstrate how real-world gaming systems handle probabilities and reward structures, even though the implementation of the mechanisms under the hood in this project is simplified for educational purposes [36]. A basic understanding of the programming constructs is needed to develop this application. game results are decided using control structures such as if-else statements and game play is repetitive with while loops, which continue until the user wishes to exit or the user runs out of money. Functions are implemented to group the code into reusable blocks, making the code more readable and maintainable [47]. Random number generation is essential for simulating the randomness of slot machines, ensuring that each spin produces a unique result, and replicating the behaviour of actual slot machines [67]. Also, part of it is the handling of user input and output, which is an important part, since players will interact with the game through a command line interface.

One of the nicest things about this project is that it is simple [63]. This slot machine is completely inside a console environment, not like modern graphical games that use complex engines and visual interfaces. This design choice means that the emphasis is on logic and functionality, not aesthetics. The game itself is simple but is designed to be fun, adding in elements such as betting, winning conditions and keeping track of balance [40]. These features create a sense of progression and risk, which are important for gambling-style games. The system starts by initialising a virtual balance for the user, which is the number of credits the user has available to bet [66]. Before each spin, the user is then asked to place a bet. Input validation is used to ensure that the user does not enter invalid or excessive bet amounts, thus maintaining the integrity of the game [60]. When a valid bet is placed the slot machine is activated and symbols are randomly generated on each reel. These symbols are usually maintained in a list that is defined beforehand and the spinning behaviour is simulated using Python's random selection functions.

Once the symbols are generated, the program evaluates the result according to the matching conditions [55]. For example, if all symbols are matching, the user may get a high payout, whereas partial matches may result in smaller rewards or no reward at all. The evaluation is made

by using conditional statements to compare the generated results with defined winning patterns. Depending on the result of the spin, the balance is then adjusted to suit, either up or down. This constant updating of state of the game brings out the idea of dynamic variables in programming [44]. Another important feature of the project is the balance tracker, which keeps the record of the user's virtual credits during the gameplay session. This adds another layer of realism, with players needing to carefully manage their resources while deciding how much to bet in each round. The balance system also adds in strategic thinking as players may choose to either bet on the safe side or take the risk of betting big in hope of higher rewards [52]. This is representative of real-world gambling behaviour and enhances the educational value of the project by showing the effect of decision-making on outcomes in probabilistic systems.

The project also includes a loop-based gameplay structure, allowing users to play multiple rounds without having to restart the program [50]. This loop continues until the user decides to quit the game or runs out of balance. The reason for the need of loops in this context is to allow the game logic to be run a number of times and to be able to store the results of previous spins [42]. It also shows how programs can remember state over time and react to user input in a dynamic way. From the point of view of software development this project is an introduction to structured programming and modular design [64]. The game is divided into functions for generating spins, evaluating outcomes, and managing the balance, making the code easier to understand and maintain [59]. This modular approach also allows for future enhancements as additional features can be added without changing the core logic much.

It also illustrates the importance of randomisation in game development [46]. If slot machines were not random, they would be predictable and would lose their purpose. One of the most common ways to generate random results is to use Python's random library, which is simple but effective and critical for simulating games of chance [62]. This follows the idea of stochastic processes in programming, where the outcome is influenced by probability, rather than fixed logic. Besides technical learning the project also gives an insight into basic game design principles. It's a text-based game, but it still has all the core components of things like user engagement, rewards and progression [39]. These are the foundations of modern game development. They can be built upon with graphical interfaces and more complex mechanics. The problem statement of this project is about to make a working simulation of a real slot machine via Python. Players can place bets and spin the reels, winning or losing virtual credits based on the random arrangement of symbols. The trick is you have to do it using basic programming constructs [54]. No frameworks. No graphical tools. The project is educational & practical both. It teaches how to build real-world systems with basic knowledge of programming.

The project is to design a simple but functional slot machine game that accurately simulates the behaviour of real slot machines, but is accessible to beginners [58]. The project is designed to reinforce the basics of programming such as loops, conditionals, functions and random number generation. It also aims to develop an intuitive textual interface which makes it easy for users to interact with the game [41]. Also, it shows how to build interesting applications in Python with simple logic and structure. This project belongs to the field of game development and software simulation. Particularly, it focuses on text-based game design with Python, where real-world gambling mechanics are duplicated in a digital mini form. This domain is very good for beginners in programming as it involves logic building as well as interactive application development [49]. It also serves as a portal to more advanced topics in game development, like graphical user interfaces and complex simulation systems.

The project's scope is intentionally restricted to a simple console application so that it remains

uncomplicated and concentrates on fundamental programming concepts [53]. They can place their bets, spin the reels and get paid for the symbol combinations. It doesn't have more advanced features like graphical interfaces or networked play, but these are listed as potential future improvements [65]. The main goal is to have a working model of a slot machine that is easy to understand and extend. In future development, the project can be extended significantly by adding graphical user interfaces using libraries such as Tkinter or PyGame. You can also add other features such as sound effects, animations, leaderboards, multi level betting systems etc [38]. Even machine learning techniques could be considered to analyse user behaviour and dynamically tune difficulty or reward systems. These enhancements would turn the plain console app into a more complex gaming experience. Overall, the Python Slot Machine project is a successful demonstration of the application of basic programming concepts to create an interactive, functional game [45]. It provides valuable learning experience in logic development, user interaction and system design. This project is a simple and educational example of a slot machine [61]. It combines random numbers, conditionals and structured programming to simulate the main operation of a real slot machine.

Literature Review

Designing and implementing slot-machine simulations is a multi-disciplinary area, involving ideas from computer science, probability theory, human-computer interaction, educational technology, and behavioral psychology [32]. There has been much interest recently in understanding how randomness is generated in digital systems, how users interact with simulated gambling environments and how such systems can be effectively used as educational tools [14]. These fields, combined, provide the theoretical and practical foundation to develop a simplified but functional simulation for this project based on the Python slot machine [24]. These systems are designed not only to mimic the mechanics of traditional slot machines, but also to illustrate the interplay between computational logic, randomness, and decision making within a controlled software environment.

Any slot machine simulation must have a random outcome generator [29]. In digital systems the randomness is not true randomness but is produced by algorithmic processes known as pseudo-random number generators. The algorithms are deterministic under the hood, but are designed to mimic randomness [11]. Programming environments such as Python typically include built-in randomization functions that enable you to simulate unpredictable behavior. These functions are generally based on well established algorithms that yield enough randomness for educational and simulation purposes. But the quality of the randomness can have a huge impact on how fair and unpredictable a game feels. If badly designed patterns can emerge that can be detected by users reducing the effectiveness of the simulation [18]. Hence, the choice of an appropriate randomization method is critical to ensure that the slot machine operates in a realistic and non-biased way.

The built-in random selection functionality of Python is used to generate the symbol combinations for each spin of the slot machine. The approach is a trade-off between simplicity and effectiveness and is appropriate for didactic applications [3]. The randomness guarantees that each spin is independent of the previous results, simulating the unpredictability of real-world slot machines. This idea is at the heart of the development of probabilistic systems in programming, where the result is not fixed but is controlled by the use of randomness [25]. The project demonstrates how these mechanisms can be used for effective incorporation of randomness in software applications to simulate real life systems.

Another important aspect of slot machine design is the structure of the probability distributions

and payout systems. In real casinos, the slot machines are designed to be fun for the player, but also profitable for the casino [7]. This is done by carefully adjusting the probabilities. Different symbols are assigned different frequencies, which directly affect the probability of particular outcomes [21]. The common symbols appear more frequently, and the high value symbols appear less frequently creating a structured imbalance that determines the expected return over time. The project realizes this idea by means of weighted symbol distributions, where some symbols are assigned higher probabilities than others. This not only enables the simulation to mimic realistic gameplay behavior, but also to show how probability can be manipulated in a controlled system.

A big idea in the math behind how slot machines work is expected value. Each time a slot machine is spun, the possible results and the likelihood of each result can be examined. The outcomes are still random, but you can calculate expected returns to see what happens to the system in the long run [1]. For this project we have added simplified payout logic to reflect these principles without adding unnecessary complexity. Winning combinations get better rewards, while less favorable outcomes get worse rewards or no rewards [17]. This framework is useful to demonstrate how mathematical modeling can be used in game design to create balanced and fun systems.

A slot machine simulation is a common educational example for teaching programming [6]. Projects of this kind are interactive and allow learners to directly manipulate the basic programming constructs such as loops, conditionals, functions, and data structures [31]. A game-based system enables learners to see how code becomes behavior, giving a tangible representation of abstract ideas. Loops allow you to play a game again and again. Conditionals tell you who wins and loses the game [26]. Functions help you organize your logic into pieces you can reuse. This practical approach has been proven to enhance understanding and retention of fundamental programming concepts, particularly for novices.

The Python slot machine project's modular design adds to its educational value [20]. This splits the program into different chunks of functionality such as input handling, game logic and output display. This makes the system easier to understand and maintain. This separation of concerns is consistent with best practice in software engineering, where complex systems are decomposed into manageable modules [33]. Each module performs a specific task and interacts with others via well-defined interfaces. This architecture enhances code readability and simplifies debugging and future enhancements [13]. For example, you could change the scoring system without changing the input handling or the user interface logic.

Interaction with the user is also an important part of slot machine simulations. The user interacts with the system through text input and output in a console-based environment [4]. This is fairly easy when compared to graphical interfaces, but it is a great way to demonstrate the basics of programming. Users are invited to bet, start spins and see results, etc. This interaction model reinforces the connection between user input and system output, a fundamental tenet of software design [15]. Furthermore, input validation maintains the stability of the system when users provide unexpected or invalid data

Also important in the design of such systems are game balance and user experience [22]. Even in a simple simulation, you need to make sure the game is interesting and fair. The system loses its intended purpose as a probabilistic simulation if the outcomes are too predictable or the rewards too frequent. However, too infrequent or unpredictable rewards may cause users to lose interest. So finding a good balance between randomness and reward is key. In this project the balance is achieved by carefully defined rules of how symbols can be combined and how much you can win [9]. This way you can win and lose but in a very controlled way.

The notion of continuous play is achieved through looping mechanisms, which make the game run again and again until a stopping condition is reached. This design is the foundation for many interactive applications, especially games. The loop keeps track of the system state (e.g., user's balance and game history) across multiple iterations [2]. Each pass of the loop is one complete round of the game, from input collection to outcome evaluation and result presentation [19]. This structure illustrates how persistent systems function in programming: the state is stored and modified over time as the user interacts with the system and as the system logic directs.

A coherent system that combines randomization, probability and user interaction simulates the experience of a slot machine [12]. The implementation is a simplified version of real gambling systems, but it effectively demonstrates the basic principles that run such machines. It also emphasizes the potential of computational systems to model real world behaviour by means of abstraction and simplification [27]. This is particularly useful to the project as a learning tool, as it provides a bridge between theory and practice.

The project introduces learners to basic design principles from a software engineering perspective such as reusability, maintainability and scalability [34]. The code is organized into functions and logical modules, which makes the system easier to extend in the future. New features like extra symbols, more complicated payout rules or graphical interfaces can be added without having to re-implement the entire system. This points to the importance of planning and architecture in software development, even in small projects [8]. Testing and validation are also key parts of the development process. For reliability it is vital to ensure that the system will operate as expected in different conditions. The tests for this project include checking that input is handled correctly, random outcomes are generated fairly, and balance updates are reflected by correct calculations. We also should think about edge cases like invalid inputs or extreme betting values to make sure the system is robust [28]. This increases the importance of systematic testing in the development of software.

In addition to the technical aspects, the project also raises questions about the wider implications of simulated gambling systems [23]. The application is purely educational, but it is a reflection of real-world systems that involve financial risk and probability-based decision-making. Learning about these systems can offer useful insight into how these digital gambling environments are built and how they affect users' behavior [30]. This awareness is important, not only in a technical sense but also in an ethical sense, especially when designing systems that involve chance-based outcomes. Future developments of the project could include the addition of graphical user interfaces in order to improve user engagement and visual appeal [5]. The current version is console based and focuses on logic and functionality, but graphical improvements could enhance the immersive experience. Also, complexity and replayability can be increased by implementing features like persistent data storage, leaderboards, or adaptive difficulty systems. These additions would turn the simple simulation into a more sophisticated game app while still keeping the educational aspect.

The Python slot machine project is a full example of how simple programming concepts can be used to build an engaging and meaningful simulation [35]. The project has successfully abstracted the main mechanics of a slot machine using randomness, probability, modular design and user interaction [10]. It offers valuable learning experiences in programming logic, system design, and the role of randomness in computational systems [16]. At the same time, it sets the stage for future expansion into more complex and visually rich applications, showing the scalability and adaptability of well-structured code.

2. Methodology

This project creates a Python slot machine game that simulates a simplified version of the classic casino style gambling experience [73]. The primary aim of the system is to mimic the fundamental mechanics of a slot machine using basic programming concepts such as loops, conditionals, functions and randomisation [80]. The game is interactive and user driven, allowing players to participate in multiple rounds of gameplay while maintaining a virtual balance [69]. The starting balance is the initial amount of money the user has at the beginning of the game [86]. This balance is used to place bets in each round of gameplay. Before spinning the slot machine the user is asked to select the number of betting lines they would like to play on, and the bet amount per line [76]. This brings an element of control and strategy, as the total wager will depend on the number of active lines and value of bet selected [82]. The slot machine itself is depicted as a 3x3 grid containing different symbols.

The symbols are randomly generated and placed in columns in order to simulate the spinning reels of a real slot machine. Every spin creates a new random layout and so no two rounds are the same, and all are a surprise [74]. It is that randomness of symbol placement that is one of the key features of the game. Each outcome is independent of the previous results [85]. After each spin the program checks the selected betting lines for any matching symbols. The user wins if matching symbols are found across a line, according to a set of predefined symbol values [71]. The winnings are then calculated accordingly and added to the player's balance while losses are deducted based on unsuccessful bets. This win/loss mechanism is very similar to how slot machines work in the real world [87]. The arcade game continues in a loop, allowing the player to place new bets and spin again as long as the player has sufficient balance and chooses to continue. The session ends when the player decides to quit or runs out of money [78]. This structure provides an engaging and dynamic gameplay experience with multiple opportunities for risk and reward.

Project Description

The design of virtual slot machines has evolved considerably, relying mainly on probabilistic models to simulate randomness and fairness [83]. Modern implementations use statistical principles to distribute the frequency of symbols to ensure that play is not biased (Abeyruwan & Calmet, 2016). Traditional systems are just simple random selection algorithms. The distributions of symbols are fixed and this will define the outcome for the player [70]. The system takes the user input for deposits and bets, the random module functions for spinning reels, and the pre-assigned symbol values for calculating payouts, which all are essential for educational projects like the python slot machine [81]. They are designed to be modular, and teach basic programming concepts such as loops, conditionals, and list manipulation. But they tend to be missing the sophisticated real-time analytics, dynamic payout adjustments, or security validations you find in advanced commercial systems [75]. In general, the current educational systems are designed to make things easy and teach basic concepts about probability, not to replicate the complex rules of actual gambling sites

Proposed System

The proposed system enhances the basic Python-based slot machine by incorporating better randomization logic, user experience improvements, and modular code design [77]. Unlike earlier models that merely simulate spins, this system introduces adjustable symbol frequency settings, scalable reel sizes, and configurable bet limits. It emphasizes input validation at every stage to prevent errors and ensure a smooth user experience [68]. The architecture follows a modular procedural design, promoting maintainability and future upgrades, such as adding new symbols or bonus features [84]. Additionally, the system provides real-time balance tracking and clear visual representation of the slot grid, improving user engagement [79]. While maintaining simplicity for educational purposes, the proposed system also sets the groundwork for future expansions like graphical user interfaces (GUI) using libraries such as Tkinter or PyGame [72]. By focusing on clean code structure and statistical fairness, the system bridges the gap between academic prototypes and real-world gaming applications.

Proposed Work

General Architecture

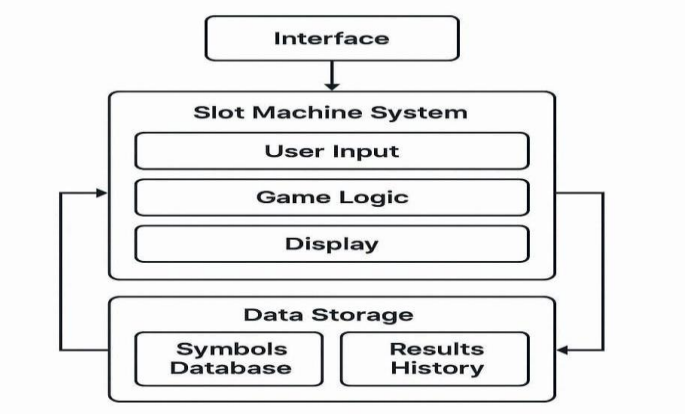

Figure 1. Architecture Diagram**.**

Figure 4.1 shows the system architecture of the Python slot machine project, providing an overview of how the user interface, game logic, and data management modules work together [88]. It shows a clear modular flow of how inputs are processed, spins are generated, results are displayed and balances are updated.

Design Phase: A variety of diagrams and models are created in the design phase to depict different aspects of the system, including its components, interactions, and data flow. UML, Sequence diagrams, Use case diagrams, Data flow diagrams [91]. These types of diagrams are helpful in communicating the design and functionality of a system to stakeholders and development teams [98]. In short, the design phase is crucial for the successful accomplishment of the software solution's objectives in a proficient and effective manner [94].

Data Flow Diagram

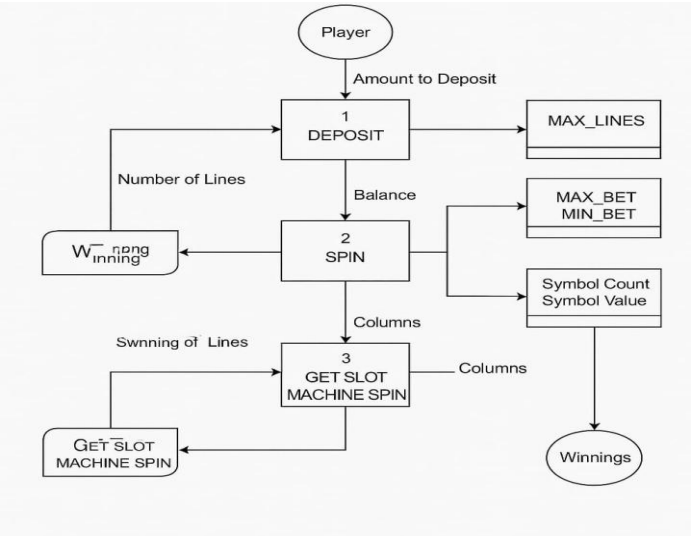

Figure 2. Data Flow Diagram**.**

The flowchart of the slot machine game system is shown in Figure 4.2 [92]. It includes the deposit of the players, the choice of the betting lines, the spinning and the calculation of the winnings based on the values and the number of the symbols [99]. The diagram depicts player, balance, and game mechanic interactions.

Use Case Diagram

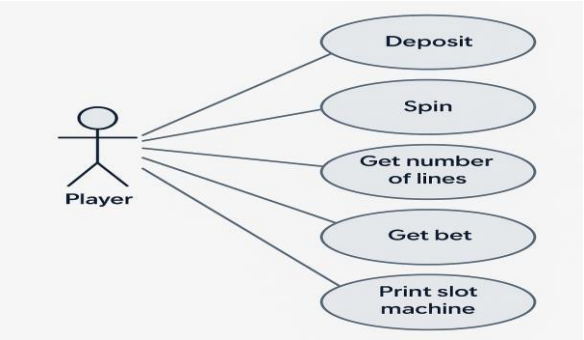

Figure 3. Use Case Diagram**.**

Figure 4.3: Formal Use Case Diagram for a Slot Machine System Figure 4.3 shows the interaction of the actor "Player" and the functionalities of the system [95]. It presents key user actions such as depositing, spinning, getting the number of lines, placing a bet and printing the slot machine outcome in a structured manner [90].

Sequence Diagram

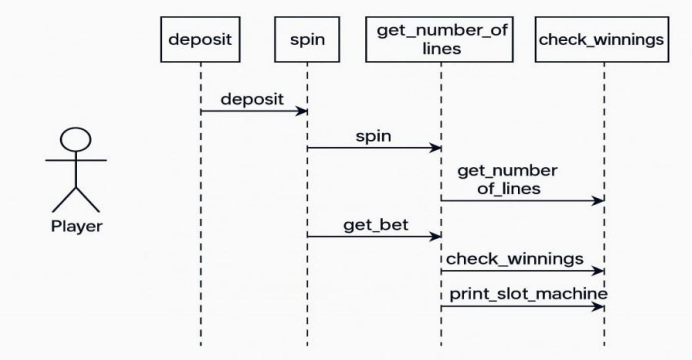

Figure 4. Sequence Diagram**.**

The sequence diagram describes the dynamic behaviour of the slot machine system [89]. It shows the order of interactions between the player and various functions of the system [93]. It captures the flow of messages like deposit, spin, get number of lines, get bet, check winnings and print slot machine [96]. Thus providing clear communication of the system.

3. Results and discussions

The proposed system provides efficiency through modular code design, scalable reel configurations, adjustable symbol frequencies, and robust input validation [103]. This guarantees better gameplay, user engagement and ease of future upgrades. Easy to use with real time balance and clear visual outputs. The system is designed to maximise functionality while being simple enough for educational and future real-world applications with a focus on clean architecture and statistical fairness [105]. The system improves on modularity, user experience and flexibility by allowing variable symbol frequencies, scalable reels and strong validation of user input [101]. It adds real-time balance tracking and clear visual outputs to improve engagement and future expandability. This system provides only basic slot machine simulation with simple randomisation and fixed settings, mainly for teaching basic programming concepts, and it does not include any advanced user-centric features.

Feature	Existing System	Proposed System
Randomization Logic	Basic random spin selection with fixed probabilities	Advanced randomization with adjustable symbol frequency
User Experience	Basic command-line inputs and outputs	Enhanced input validation, real-time balance updates
Code Structure	Simple, linear code: limited modularity	Modular procedural code: supports easyıodates an scalbility
Flexibility (Symbols/Reels)	Fixed number of reels and symbols	Dynamic reel sizes and customizable symbol sets
Error Handling	Minimal error checking; prone to invalid inputs	Rigorous input validation at every step; robust error handling
Future Expandability	Difficult to add new features without major code rewrite	Easy to extend with new features (e.g., bonus rounds, GUI
Educational Value	Focuses on basic programming constructs (loops, lists	Teaches modular programming, input validation, statistical faimess
Real-World Applicability	Simpler academic prototype; not close to industry-grade implements	Bridging academic learning and real-world game design practices

Figure 5. Comparison table between existing and proposed system.

The proposed system is able to improve the overall structure, usability and future adaptation of the Python slot machine project. The system is supposed to be more modular, so different parts like input handling, game logic and result calculation are nicely separated. This makes the program easier to understand, maintain and extend. This new architecture also reduces complexity within individual sections of code, allowing for smoother debugging and more efficient development. Also, strong input validation helps to make the system reliable by ensuring that only valid betting values and user inputs are processed, avoiding errors and unexpected behaviour during gameplay [100]. Another big improvement is the availability of dynamic configurations, which makes it easier to adjust game parameters like bet limits, symbol probabilities and payout values. This means that the system can be more easily adapted to different gameplay scenarios and be more flexible [104]. This results in a better overall user experience with clearer output messages, smoother gameplay flow and better balance management, all of which help to make the interaction more engaging for the player [102]. The overall design makes the game not just functional, but intuitive and fun to use, even in a simple console-based environment.

4. Conclusion

Moreover, the project offers a good basis for scalability, which means that in the future it will be possible to improve it without substantial structural changes. One of the major improvements we plan to do in the future is to add graphical user interfaces using frameworks like Tkinter or PyGame. This would replace the current text-based system with a visually interactive application with animations, images and sound effects, significantly increasing user engagement. Additional symbols and game features such as bonus rounds, jackpot features, and special multipliers could be added to further enhance the gaming experience and make the system more similar to modern slot machine designs. Dynamic reel adjustments that vary symbol distributions based on player performance or game progression may be implemented to further enhance the gameplay environment. Another great improvement is the possibility to see the statistics in real time, so the users can see their win/lose ratio, total spins and performance history overall. This would create an analytical feature to the game and allow users to get a better understanding of their gameplay patterns. More user customisation options with adjustable themes, sound settings, and difficulty levels would make it more personalised and increase replay value. In general, these proposed enhancements aim to evolve the existing system from a simple educational simulation to a more complex and interactive gaming application. Adding graphical interfaces, adaptive mechanics and user orientated features allows the project to slowly reach real-world gaming standards, not losing its educational function and simplicity.

References

[1]. H. AbdulKader, E. ElAbd, and W. Ead, "Protecting online social networks profiles by hiding sensitive data attributes," Procedia Computer Science, vol. 82, pp. 20–27, 2016.

[2]. I. E. Fattoh, F. Kamal Alsheref, W. M. Ead, and A. M. Youssef, "Semantic sentiment classification for COVID-19 tweets using universal sentence encoder," Computational Intelligence and Neuroscience, vol. 2022, pp. 1–8, 2022.

[3]. R. Boina, "Assessing the Increasing Rate of Parkinson's Disease in the US and its Prevention Techniques"," International Journal of Biotechnology Research and Development, vol. 3, no. 1, pp. 1–18, 2022.

[4]. W. M. Ead, W. F. Abdel-Wahed, and H. Abdul-Kader, "Adaptive fuzzy classification-rule algorithm in detection malicious web sites from suspicious URLs," International Arab Journal of e-Technology, vol. 3, pp. 1–9, 2013.

[5]. M. A. Abdelazim, M. M. Nasr, and W. M. Ead, "A survey on classification analysis for cancer genomics: Limitations and novel opportunity in the era of cancer classification and target therapies," Annals of Tropical Medicine and Public Health, vol. 23, no. 24, 2020.

[6]. F. K. Alsheref, I. E. Fattoh, and W. M. Ead, "Automated prediction of employee attrition using ensemble model based on machine learning algorithms," Computational Intelligence and Neuroscience, vol. 2022, pp. 1–9, 2022.

[7]. M. Razeghi, A. Dehzangi, D. Wu, R. McClintock, Y. Zhang, Q. Durlin, et al., "Antimonite-based gap-engineered type-II superlattice materials grown by MBE and MOCVD for the third generation of infrared imagers," in Infrared Technology and Applications XLV, vol. 11002, pp. 108–125, May 2019, SPIE.

[8]. F. Meng, L. Zhang, and Y. Chen, "FEDEMB: An efficient vertical and hybrid federated learning algorithm using partial network embedding," 2023.

[9]. F. Meng, L. Jagadeesan, and M. Thottan, "Model-based reinforcement learning for service mesh fault resiliency in a web application-level," arXiv preprint arXiv:2110.13621, 2021.

[10]. F. Meng, L. Zhang, Y. Chen, and Y. Wang, "Sample-based dynamic hierarchical transformer with layer and head flexibility via contextual bandit," Authorea Preprints, 2023.

[11]. D. K. Sharma and R. Tripathi, "4 Intuitionistic fuzzy trigonometric distance and similarity measure and their properties," in Soft Computing, De Gruyter, Berlin, Germany, pp. 53–66, 2020.

[12]. D. K. Sharma, B. Singh, M. Anam, R. Regin, D. Athikesavan, and M. Kalyan Chakravarthi, "Applications of two separate methods to deal with a small dataset and a high risk of generalization," in 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2021.

[13]. D. K. Sharma, B. Singh, M. Anam, K. O. Villalba-Condori, A. K. Gupta, and G. K. Ali, "Slotting learning rate in deep neural networks to build stronger models," in 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2021.

[14]. K. Kaliyaperumal, A. Rahim, D. K. Sharma, R. Regin, S. Vashisht, and K. Phasinam, "Rainfall prediction using deep mining strategy for detection," in 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2021.

[15]. I. Nallathambi, R. Ramar, D. A. Pustokhin, I. V. Pustokhina, D. K. Sharma, and S. Sengan, "Prediction of influencing atmospheric conditions for explosion Avoidance in fireworks manufacturing Industry-A network approach," Environ. Pollut., vol. 304, no. 7, p. 119182, 2022.

[16]. H. Sharma and D. K. Sharma, "A Study of Trend Growth Rate of Confirmed Cases, Death Cases and Recovery Cases of Covid-19 in Union Territories of India," Turkish Journal of Computer and Mathematics Education, vol. 13, no. 2, pp. 569–582, 2022.

[17]. A. L. Karn et al., "Designing a Deep Learning-based financial decision support system for fintech to support corporate customer's credit extension," Malays. J. Comput. Sci., vol.36, no.s1, pp. 116–131, 2022.

[18]. A. L. Karn et al., "B-lstm-Nb based composite sequence Learning model for detecting fraudulent financial activities," Malays. J. Comput. Sci., vol.32, no.s1, pp. 30–49, 2022.

[19]. P. P. Dwivedi and D. K. Sharma, "Application of Shannon entropy and CoCoSo methods in selection of the most appropriate engineering sustainability components," Cleaner Materials, vol. 5, no. 9, p. 100118, 2022.

[20]. A. Kumar, S. Singh, K. Srivastava, A. Sharma, and D. K. Sharma, "Performance and stability enhancement of mixed dimensional bilayer inverted perovskite (BA2PbI4/MAPbI3) solar cell using drift-diffusion model," Sustain. Chem. Pharm., vol. 29, no. 10, p. 100807, 2022.

[21]. A. Kumar, S. Singh, M. K. A. Mohammed, and D. K. Sharma, "Accelerated innovation in developing high-performance metal halide perovskite solar cell using machine learning," Int. J. Mod. Phys. B, vol. 37, no. 07, p.12, 2023.

[22]. G. A. Ogunmola, M. E. Lourens, A. Chaudhary, V. Tripathi, F. Effendy, and D. K. Sharma, "A holistic and state of the art of understanding the linkages of smart-city healthcare technologies," in 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022.

[23]. P. Sindhuja, A. Kousalya, N. R. R. Paul, B. Pant, P. Kumar, and D. K. Sharma, "A Novel Technique for Ensembled Learning based on Convolution Neural Network," in 2022 International Conference on Edge Computing and Applications (ICECAA), IEEE, Tamil Nadu, India, pp. 1087–1091, 2022.

[24]. A. R. B. M. Saleh, S. Venkatasubramanian, N. R. R. Paul, F. I. Maulana, F. Effendy, and D. K. Sharma, "Real-time monitoring system in IoT for achieving sustainability in the agricultural field," in 2022 International Conference on Edge Computing and Applications (ICECAA), Tamil Nadu, India, 2022.

[25]. Srinivasa, D. Baliga, N. Devi, D. Verma, P. P. Selvam, and D. K. Sharma, "Identifying lung nodules on MRR connected feature streams for tumor segmentation," in 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Tamil Nadu, India, 2022.

[26]. C. Goswami, A. Das, K. I. Ogaili, V. K. Verma, V. Singh, and D. K. Sharma, "Device to device communication in 5G network using device-centric resource allocation algorithm," in 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Tamil Nadu, India , 2022.

[27]. M. Yuvarasu, A. Balaram, S. Chandramohan, and D. K. Sharma, "A Performance Analysis of an Enhanced Graded Precision Localization Algorithm for Wireless Sensor Networks," Cybernetics and Systems, pp. 1–16, 2023, Press.

[28]. P. P. Dwivedi and D. K. Sharma, "Evaluation and ranking of battery electric vehicles by Shannon's entropy and TOPSIS methods," Math. Comput. Simul., vol. 212, no.10, pp. 457–474, 2023.

[29]. P. P. Dwivedi and D. K. Sharma, "Assessment of Appropriate Renewable Energy Resources for India using Entropy and WASPAS Techniques," Renewable Energy Research and Applications, vol. 5, no. 1, pp. 51–61, 2024.

[30]. P. P. Dwivedi and D. K. Sharma, “Selection of combat aircraft by using Shannon entropy and VIKOR method,” Def. Sci. J., vol. 73, no. 4, pp. 411–419, 2023.
[31]. B. Senapati and B. S. Rawal, “Adopting a deep learning split-protocol based predictive maintenance management system for industrial manufacturing operations,” in Lecture Notes in Computer Science, Singapore: Springer Nature Singapore, pp. 22–39, 2023.
[32]. B. Senapati and B. S. Rawal, “Quantum communication with RLP quantum resistant cryptography in industrial manufacturing,” Cyber Security and Applications, vol. 1, no. 12, p. 100019, 2023.
[33]. B. Senapati et al., “Wrist crack classification using deep learning and X-ray imaging,” in Proceedings of the Second International Conference on Advances in Computing Research (ACR’24), Cham: Springer Nature Switzerland, pp. 60–69, 2024.
[34]. A. B. Naeem et al., “Heart disease detection using feature extraction and artificial neural networks: A sensor-based approach,” IEEE Access, vol. 12, no.3, pp. 37349–37362, 2024.
[35]. R. Tsarev et al., “Automatic generation of an algebraic expression for a Boolean function in the basis ∧, ∨, ¬,” in Data Analytics in System Engineering, Cham: Springer International Publishing, Switzerland, pp. 128–136, 2024.
[36]. R. Tsarev, B. Senapati, S. H. Alshahrani, A. Mirzagitova, S. Irgasheva, and J. Ascencio, “Evaluating the effectiveness of flipped classrooms using linear regression,” in Data Analytics in System Engineering, Cham: Springer International Publishing, Switzerland, pp. 418–427, 2024.
[37]. M. A. Yassin et al., “Advancing SDGs : Predicting Future Shifts in Saudi Arabia ’ s Terrestrial Water Storage Using Multi-Step-Ahead Machine Learning Based on GRACE Data,” 2024.
[38]. M. A. Yassin, A. G. Usman, S. I. Abba, D. U. Ozsahin, and I. H. Aljundi, “Intelligent learning algorithms integrated with feature engineering for sustainable groundwater salinization modelling: Eastern Province of Saudi Arabia,” Results Eng., vol. 20, p. 101434, 2023.
[39]. S. I. Abba, A. G. Usman, and S. IŞIK, “Simulation for response surface in the HPLC optimization method development using artificial intelligence models: A data-driven approach,” Chemom. Intell. Lab. Syst., vol. 201, no. April, 2020.
[40]. A. G. Usman et al., “Environmental modelling of CO concentration using AI-based approach supported with filters feature extraction: A direct and inverse chemometrics-based simulation,” Sustain. Chem. Environ., vol. 2, p. 100011, 2023.
[41]. A. Gbadamosi et al., “New-generation machine learning models as prediction tools for modeling interfacial tension of hydrogen-brine system,” Int. J. Hydrogen Energy, vol. 50, pp. 1326–1337, 2024.
[42]. I. Abdulazeez, S. I. Abba, J. Usman, A. G. Usman, and I. H. Aljundi, “Recovery of Brine Resources Through Crown-Passivated Graphene, Silicene, and Boron Nitride Nanosheets Based on Machine-Learning Structural Predictions,” ACS Appl. Nano Mater., 2023.
[43]. B. S. Alotaibi et al., “Sustainable Green Building Awareness: A Case Study of Kano Integrated with a Representative Comparison of Saudi Arabian Green Construction,” Buildings, vol. 13, no. 9, 2023.
[44]. S. I. Abba et al., “Integrated Modeling of Hybrid Nanofiltration/Reverse Osmosis Desalination Plant Using Deep Learning-Based Crow Search Optimization Algorithm,” Water (Switzerland), vol. 15, no. 19, 2023.
[45]. S. I. Abba, J. Usman, and I. Abdulazeez, “Enhancing Li + recovery in brine mining : integrating next-gen emotional AI and explainable ML to predict adsorption energy in

crown ether-based hierarchical nanomaterials," pp. 15129–15142, 2024.
[46]. J. Usman, S. I. Abba, N. Baig, N. Abu-Zahra, S. W. Hasan, and I. H. Aljundi, "Design and Machine Learning Prediction of In Situ Grown PDA-Stabilized MOF (UiO-66-NH2) Membrane for Low-Pressure Separation of Emulsified Oily Wastewater," ACS Appl. Mater. Interfaces, Mar. 2024.
[47]. S. K. Sehrawat, "Transforming Clinical Trials: Harnessing the Power of Generative AI for Innovation and Efficiency," Transactions on Recent Developments in Health Sectors, vol. 6, no. 6, pp. 1-20, 2023.
[48]. S. K. Sehrawat, "Empowering the Patient Journey: The Role of Generative AI in Healthcare," International Journal of Sustainable Development Through AI, ML and IoT, vol. 2, no. 2, pp. 1-18, 2023.
[49]. S. K. Sehrawat, "The Role of Artificial Intelligence in ERP Automation: State-of-the-Art and Future Directions," Transactions on Latest Trends in Artificial Intelligence, vol. 4, no. 4, 2023.
[50]. P. P. Anand, U. K. Kanike, P. Paramasivan, S. S. Rajest, R. Regin, and S. S. Priscila, "Embracing Industry 5.0: Pioneering Next-Generation Technology for a Flourishing Human Experience and Societal Advancement," FMDB Transactions on Sustainable Social Sciences Letters, vol.1, no. 1, pp. 43–55, 2023.
[51]. G. Gnanaguru, S. S. Priscila, M. Sakthivanitha, S. Radhakrishnan, S. S. Rajest, and S. Singh, "Thorough analysis of deep learning methods for diagnosis of COVID-19 CT images," in Advances in Medical Technologies and Clinical Practice, IGI Global, pp. 46–65, 2024.
[52]. G. Gowthami and S. S. Priscila, "Tuna swarm optimisation-based feature selection and deep multimodal-sequential-hierarchical progressive network for network intrusion detection approach," Int. J. Crit. Comput.-based Syst., vol. 10, no. 4, pp. 355–374, 2023.
[53]. A. J. Obaid, S. Suman Rajest, S. Silvia Priscila, T. Shynu, and S. A. Ettyem, "Dense convolution neural network for lung cancer classification and staging of the diseases using NSCLC images," in Proceedings of Data Analytics and Management, Singapore; Singapore: Springer Nature, pp. 361–372, 2023.
[54]. S. S. Priscila and A. Jayanthiladevi, "A study on different hybrid deep learning approaches to forecast air pollution concentration of particulate matter," in 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023.
[55]. S. S. Priscila, S. S. Rajest, R. Regin, and T. Shynu, "Classification of Satellite Photographs Utilizing the K-Nearest Neighbor Algorithm," Central Asian Journal of Mathematical Theory and Computer Sciences, vol. 4, no. 6, pp. 53–71, 2023.
[56]. S. S. Priscila and S. S. Rajest, "An Improvised Virtual Queue Algorithm to Manipulate the Congestion in High-Speed Network"," Central Asian Journal of Medical and Natural Science, vol. 3, no. 6, pp. 343–360, 2022.
[57]. S. S. Priscila, S. S. Rajest, S. N. Tadiboina, R. Regin, and S. András, "Analysis of Machine Learning and Deep Learning Methods for Superstore Sales Prediction," FMDB Transactions on Sustainable Computer Letters, vol. 1, no. 1, pp. 1–11, 2023.
[58]. R. Regin, Shynu, S. R. George, M. Bhattacharya, D. Datta, and S. S. Priscila, "Development of predictive model of diabetic using supervised machine learning classification algorithm of ensemble voting," Int. J. Bioinform. Res. Appl., vol. 19, no. 3, 2023.
[59]. S. Silvia Priscila, S. Rajest, R. Regin, T. Shynu, and R. Steffi, "Classification of Satellite

Photographs Utilizing the K-Nearest Neighbor Algorithm," Central Asian Journal of Mathematical Theory and Computer Sciences, vol. 4, no. 6, pp. 53–71, 2023.

[60]. S. S. Rajest, S. Silvia Priscila, R. Regin, T. Shynu, and R. Steffi, "Application of Machine Learning to the Process of Crop Selection Based on Land Dataset," International Journal on Orange Technologies, vol. 5, no. 6, pp. 91–112, 2023.

[61]. T. Shynu, A. J. Singh, B. Rajest, S. S. Regin, and R. Priscila, "Sustainable intelligent outbreak with self-directed learning system and feature extraction approach in technology," International Journal of Intelligent Engineering Informatics, vol. 10, no. 6, pp.484-503, 2022.

[62]. S. S. Priscila, D. Celin Pappa, M. S. Banu, E. S. Soji, A. T. A. Christus, and V. S. Kumar, "Technological frontier on hybrid deep learning paradigm for global air quality intelligence," in Cross-Industry AI Applications, IGI Global, pp. 144–162, 2024.

[63]. S. S. Priscila, E. S. Soji, N. Hossó, P. Paramasivan, and S. Suman Rajest, "Digital Realms and Mental Health: Examining the Influence of Online Learning Systems on Students," FMDB Transactions on Sustainable Techno Learning, vol. 1, no. 3, pp. 156–164, 2023.

[64]. S. R. S. Steffi, R. Rajest, T. Shynu, and S. S. Priscila, "Analysis of an Interview Based on Emotion Detection Using Convolutional Neural Networks," Central Asian Journal of Theoretical and Applied Science, vol. 4, no. 6, pp. 78–102, 2023.

[65]. S. Temara, "Maximizing Penetration Testing Success with Effective Reconnaissance Techniques Using ChatGPT", Asian Journal of Research in Computer Science, vol. 17, no. 5, pp. 19–29, 2024.

[66]. S. Temara, "The Ransomware Epidemic: Recent Cybersecurity Incidents Demystified", Asian Journal of Advanced Research and Reports, vol. 18, no. 3, pp. 1–16, Feb. 2024.

[67]. S. Temara, "Harnessing the power of artificial intelligence to enhance next-generation cybersecurity," World Journal of Advanced Research and Reviews, vol. 23, no. 2, pp. 797–811,2024.

[68]. B. Senapati and B. S. Rawal, "Adopting a deep learning split-protocol based predictive maintenance management system for industrial manufacturing operations," in Big Data Intelligence and Computing. DataCom 2022, C. Hsu, M. Xu, H. Cao, H. Baghban, and A. B. M. Shawkat Ali, Eds., Lecture Notes in Computer Science, vol. 13864. Singapore: Springer, 2023, pp. 25–38.

[69]. B. Senapati and B. S. Rawal, "Quantum communication with RLP quantum resistant cryptography in industrial manufacturing," Cyber Security and Applications, vol. 1, 2023, Art. no. 100019.

[70]. B. Senapati et al., "Wrist crack classification using deep learning and X-ray imaging," in Proceedings of the Second International Conference on Advances in Computing Research (ACR'24), K. Daimi and A. Al Sadoon, Eds., Lecture Notes in Networks and Systems, vol. 956. Cham: Springer, 2024, pp. 72–85.

[71]. A. Kulkarni, "Generative AI-Driven for SAP Hana Analytics," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 12, no. 2, pp. 438-444, 2024.

[72]. A. Kulkarni, "Natural Language Processing for Text Analytics in SAP HANA," International Journal of Multidisciplinary Innovation and Research Methodology, vol. 3, no. 2, pp. 135-144, 2024.

[73]. A. Kulkarni, "Enhancing Customer Experience with AI-Powered Recommendations in SAP HANA," International Journal of Business, Management and Visuals, vol. 7, no. 1, pp. 1-8, 2024.

[74]. E. Geo Francis and S. Sheeja, “Enhanced intrusion detection in wireless sensor networks using deep reinforcement learning with improved feature extraction and selection,” Multimedia Tools and Applications, 2024.
[75]. E. Geo Francis and S. Sheeja, “Bi-Level Intrusion Detection in IoT Networks Using Ensemble Method and A-GRU-RNN Classifier,” Electric Power Components and Systems, 2024.
[76]. E. G. F., S. Sheeja, John, A., and J. Joseph, “Intrusion detection system with an ensemble DAE and BiLSTM in the fog layer of IoT networks,” Journal of Applied Research and Technology, vol. 22, no.6, pp. 846–862, 2024.
[77]. E. Geo Francis, S. Sheeja, E.F. Antony John and Jismy Joseph, “An Efficient Intrusion Detection System using a Multiscale Deep Bi-Directional GRU Network to Detect Blackhole Attacks in IoT based WSNs,” Journal of Multiscale Modelling, vol. 15, no. 3, 2024.
[78]. S. Chundru, "Harnessing AI's Potential: Transforming Metadata Management with Machine Learning for Enhanced Data Access and Control," International Journal of Advances in Engineering Research, vol. 27, no. 2, pp. 39-49, 2024.
[79]. S. Chundru, "Beyond Rules-Based Systems: AI-Powered Solutions for Ensuring Data Trustworthiness," International Transactions in Artificial Intelligence, vol. 7, no. 7, p. 17, 2023.
[80]. S. Chundru, "Seeing Through Machines: Leveraging AI for Enhanced and Automated Data Storytelling," International Journal of Innovations in Scientific Engineering, vol. 18, no. 1, pp. 47-57, 2023.
[81]. S. Chundru, "Cloud-Enabled Financial Data Integration and Automation: Leveraging Data in the Cloud," International Journal of Innovations in Applied Sciences & Engineering, vol. 8, no. 1, pp. 197-213, 2022.
[82]. S. Chundru, "Leveraging AI for Data Provenance: Enhancing Tracking and Verification of Data Lineage in FATE Assessment," International Journal of Inventions in Engineering & Science Technology, vol. 7, no.1, pp. 87-104, 2021.
[83]. S. Chundru, "Ensuring Data Integrity Through Robustness and Explainability in AI Models," Transactions on Latest Trends in Artificial Intelligence, vol. 1, no. 1, pp. 1-19, 2020.
[84]. Pothu, A. R., "Celery Trap: A Browser and Email-Based Extension for Proactive Phishing, Spearphishing, and Web Threat Detection," SSRN, Oct. 10, 2024. [Online]. Available: https://ssrn.com/abstract=4983399.
[85]. M. A. Raj, M. A. Thinesh, S. S. M. Varmann, A. R. Pothu, and P. Paramasivan, “Ensemble-Based Phishing Website Detection Using Extra Trees Classifier,” AVE Trends In Intelligent Computing Systems, vol. 1, no. 3, pp. 142 –156, 2024.
[86]. R. K. Dahal, "Customer satisfaction in Nepalese cellular networks," Tribhuvan University Journal, vol. 33, no. 2, pp. 59–72, 2019.
[87]. R. K. Dahal, "Contemporary management accounting techniques and organizational performance," Pravaha, vol. 26, no. 1, pp. 177–185, 2020.
[88]. R. K. Dahal, "Assessing social and environmental performance," Academy of Accounting and Financial Studies Journal, vol. 25, no. 6, pp. 1–9, 2021.
[89]. R. K. Dahal, "Performance score as a measure of organizational effectiveness," Pravaha, vol. 27, no. 1, pp. 131–138, 2021.
[90]. R. K. Dahal, "Customers' perspectives on the Nepalese cellular telecommunications services' technological and innovation capabilities," International Journal of Social

Sciences and Management, vol. 9, no. 1, pp. 41–47, 2022.
[91]. S.M. Amoodi, G.S. Kumar, and S.G.A. Hasan, "Design of II Stage Evaporative Cooling System for Residential," International Journal of Engineering and Management Research (IJEMR), vol. 5, no. 3, pp. 810-815, 2015.
[92]. T. Wahidi, S.A.P. Quadri, S.G.A. Hasan, M.G. Sundkey, and P.R. Kumar, "Experimental investigation on performance, emission and combustion analysis of CNG-Diesel enrichment with varying injection operating pressures," IOSR Journal of Mechanical And Civil Engineering, vol. 12,no.2, pp. 23-29, 2015.
[93]. K.S. Goud, K.U. Reddy, P.B. Kumar, and S.G.A. Hasan, "Magnetic Iron Oxide Nanoparticles: Various Preparation Methods and Properties," ," IJSRSET, vol. 3, no. 2, pp. 535-538, 2017.
[94]. M.S. Reddy, R. Kumaraswami, B.K. Reddy, B.A. Sai, and S.G.A. Hasan, "Extraction of Water from Ambient Air by Using Thermoelectric Modules," IJSRSET, vol. 3, no. 2, pp. 733-737, 2017.
[95]. P.C. Kumar, S. Ramakrishna, S.G.A. Hasan, and C. Rakesh, "Find the Performance of Dual Fuel Engine Followed by Waste Cooking Oil Blends with Acetylene," International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2, pp. 127-131, 2019.
[96]. D. B. Acharya, B. Divya, and K. Kuppan, "Explainable and fair AI: Balancing performance in financial and real estate machine learning models," IEEE Access, vol. 12, no.10, pp. 154022–154034, 2024.
[97]. K. Kuppan, D. B. Acharya, and B. Divya, "Foundational AI in insurance and real estate: A survey of applications, challenges, and future directions," IEEE Access, vol. 12, no. 12, pp. 181282–181302, 2024.
[98]. D. B. Acharya, K. Kuppan and B. Divya, "Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey," in IEEE Access, vol. 13, no.1, pp. 18912-18936, 2025.
[99]. E. Geo Francis, S. Sheeja, E. F. Antony John and J. Jismy, "IoT Network Security with PCA and Deep Learning for Unmasking Anomalies," 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT), Jabalpur, India, 2024, pp. 322-328.
[100]. E. Geo Francis and S. Sheeja. “IDSSA: An Intrusion Detection System with Self-adaptive Capabilities for Strengthening the IoT Network Security,” Advances in Computational Intelligence and Informatics (ICACII), Hyderbad, India, 2024, Lecture Notes in Networks and Systems, vol 993, pp. 23-30.
[101]. E. Geo Francis and S. Sheeja. “Chaotic Resilience: Enhancing IoT Security Through Dynamic Data Encryption,” Intelligent Informatics. (ISI), Bangalore, India, 2024, Smart Innovation, Systems and Technologies, vol 389, pp 331–344.
[102]. A. Kulkarni, "Digital Transformation with SAP Hana," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 12, no. 1, pp. 338-344, 2024.
[103]. A. Kulkarni, "Supply Chain Optimization Using AI and SAP HANA: A Review," International Journal of Research Radicals in Multidisciplinary Fields, vol. 2, no. 2, pp. 51-57, 2024.
[104]. A. Kulkarni, "Image Recognition and Processing in SAP HANA Using Deep Learning," International Journal of Research and Review Techniques, vol. 2, no. 4, pp. 50-58, 2024.
[105]. K. Adivigneshwaran, J. Arun, M.S. O. Akash, R. Regin, R.L. Shyja, and R. Balku, “CampusConnect: A Data-Driven Web Framework for Smart Campus Operations

Management," Ale Journal of Sustainable Intelligent System Applications, vol.1, no.1, pp. 1–15, 2026.