Advances in Computer Technology and Applications

Volume – 2

Editor

Dr. V. Sheeja Kumari

Professor / Department of Computational Intelligence, Institute of AI & ML, SIMATS School of Engineering, SIMATS University, Chennai, Tamil Nadu, India

Scripown Publications
New Delhi

Editor: *Dr. V. Sheeja Kumari*

Edition: *1st*

Publication Year: *2024*

Pages: *83*

ISBN: *978-81-19915-98-9*

Scripown Publications
2nd Floor, 304 and 305, Pocket - 4,
Sector - 22, Rohini, North West Delhi,
Delhi, 110086, India

Contents

S. No.	Chapters	Page No.
1.	Expectation Algorithm for an Economic Institution using Instrument Learning *Dr. J. Raja, Dr. S. Meera and Dr. S. Geerthik*	01-14
2.	Computer Science and Game Theory *Dr. Rajesh Patel*	15-25
3.	Introduction about Html and CSS *Raj Kumar Gupta and Sumit Kumar*	26-40
4.	Strategic Decision-Making with Perfect Information: Foundations, Applications and Insights from Game Theory *Lt J. Hajiram Beevi and S. Munawara Banu*	41-72
5.	Generative AI: Exploring its Potential and Challenges *Dr. Monika Gaur, Nidhi Kataria Chawla, Swati Sareen and Chietra Jalota*	73-83

Chapter - 1

Expectation Algorithm for an Economic Institution using Instrument Learning

Dr. J. Raja
Asso. Professor, Department of Computer Science and Engineering, Agni College of Technology, Chennai, India

Dr. S. Meera
Professor, Department of Computer Science and Engineering, Agni College of Technology, Chennai, India.

Dr. S. Geerthik
Asso. Professor, Department of Computer Science and Engineering, Agni College of Technology, Chennai, India.

Abstract

This paper is an effort at developing an Expectation algorithm for an Economic Institution using Instrument Learning. It attentions on providing an easy way of finding out whether a person will get his/her loan request endorsed based on some limitations that are input by thatperson. Based on the already published papers related to the chosen fieldof research, the findings indicated a gap between providing an easy solution for the insolvent to find out whether he/she will be fixed the loan amount and simultaneously, the banks having an easy occasion to find out whether a borrower is capable of loan repayment or not.In most of the recently done work, the papers focused only on considering single field such as considering only Credit score to evaluate Credit worthiness of the borrower or calculating the interest rates or only considering some limitation. However, none of these papers have sheda light on creation of a Web submission that would allow the user to findout the credibility of his/her loan request getting approved by the banks or non- banking economic companies (NBEC). Thus, this paper is an attempt at bridging that existing gap to provide-at user's convenience solution. The dataset used is a combination of different datasets that wereavailable on google, Kaggle and an effort of our colleagues to enter details in a google form that was circulated. Several classifiers were tested and the one which gave the best accuracy was chosen. These include Logistic regression, Support vector machine (SVM), Random Forest, XGBoost, Gradient Boost and Decision Tree classifiers. The

website created takes in the values of various parameters like income, gender, loan amount, etc. and displays whether the loan will get approved or not.This algorithm is most suitable for common people (borrower/loanee) who wish to get a loan approved from Banks, hassle-free. It can also be used by the banks to find out if the borrower is likelyto default or not. It finds most of its applications in real-time scenario wherein borrowers can keep a check on their credibility in the market atregular intervals and from the banks' perspective, this algorithm can be implemented for personal, housing, vehicle and education loans as the parameters considered remain the same.

Keywords: Loan, Prediction Algorithm, Logistic regression, Support vector machine (SVM), Random Forest, XGBoost, Gradient Boost, Decision Tree.

I. Introduction

In a country like India, one of the largest populated countries and the fastest growing economy on the globe, the banking / finance sector holds the key to unlocking the full potential of the industries and businesses. This power comes into their hands by helping in circulating the cash flow in terms of investments and loans. Even while the countryis advancing at a rapid pace, it seems that seamless loan sanctioning still, remains a concern. Many banks and financial institutions rely heavily on the most trusted system of CIBIL score in order to approvea loan amount requested by a loanee.

Although, this is the most trusted and a unanimous system for the banks to verify the loanee and it does get the job done for an average loan requesting crowd, but there are many such population groups that cannot access this facility as a monthly subscription fee is charged to the user in order to calculate the user's score and its merits. Those population categories, that cannot afford the same, get into a cycle of hit and trial which becomes a tedious process for a simple "approved/not approved".

Sometimes age could become a factor or income range could end up causing a drastic change in one's credibility to get a loan approved from banks. Also, there are many such instances wherein just by considering this score, several people in need miss out on their chanceto get a loan amount sanctioned. As it is observed that loan sanctioningis a huge source of cash flow in banks, therefore, a secured and a trustworthy process which would satisfy the borrower as well as the lender, needs to be implemented.

We have decided to build a web application that can be used by the masses to help them figure out whether they will get a loan or not based on their values entered for the various parameters asked as input.It can also be used by the banks to check the credibility of the borrower. The platform will give access

to anyone and everyone and will indicate the user whether his/her loan request will be approved inone go.

In this paper, we have proposed the implementation of six classifiers and have found the best algorithm out of these that returned the best model accuracy. Based on the accuracy obtained from each classifier,the best result was obtained with XGBoost which gave an accuracy of94.13%.

XGBoost was developed in 2014 and since then, has gained a lot of popularity in Machine learning based projects and case studies. It is flexible and works well on different popular Operating Systems like Windows, MacOS and Linux. It is essentially an open-source softwareand supports Distributed gradient boosting library.

This paper is divided into six sections. The Literature survey highlights the recent published papers that have been referred and consulted in this paper, wherein a study of various papers and what have been published under each has been briefly mentioned.

Followed by dataset and algorithms section which is an insight into the creation and cleansing of dataset used for the model creation and various classifiers that were implemented to find out which gave the best accuracy and an in-depth detail of the final classifier: XGBoost, used.

A dedicated section which describes the creation of website, the input parameters considered and the output that gets displayed. The result section shows the comparative study of various models implemented and their accuracy returned for the dataset considered. Finally concluding with a brief conclusion of what the paper has achieved at implementation level and the future scope of the same.

II. Prose Survey

Based on the research done while finalizing this topic of research, there were several papers that were referred and considered to build up this model and to bridge the gap between the already published work and the areas of problem, that needed some attention. Previously done researches were mainly based on calculating either the Credit score to check the credit worthiness of a borrower or to calculate the borrower'scredibility of loan repayment based on some parameters like marital status, gender, income etc. or a scenario where the interest rate on loanborrowed can be calculated. However, none of the previously publishedpapers have proposed a creation of web application while considering aset of several different parameters which are input by the user who is the borrower/ the bank in order to get a knowledge of whether loan

amount will get approved or not based on the entered parameter values in case of the borrower user and whether the borrower will be able to repay the sanctioned loan amount or not, by the banks or NBFCs.

In order to coagulate this paper, several previously published papers were considered which gave an insight about what has already been done in the chosen field of research and what is yet to be done in order to provide feasible solutions to both the borrower and the banks and other institutes. One of the papers considered focused mainly on the Credit Scoring Model which was used to evaluate the Credit Worthiness of an individual [1]. Another focused on finding important dataset features from several parameters so as to get the best possible results [2]. This paper gave a clear idea of machine learning models that can be used and how to effectively use those models for the required applications [3]. This paper essentially focused on Gradient Boosting classifier and calculated the Rate of Interest on the applied loan [4]. Another consulted paper provided a comparison between existing models and a logistic regression model which helped to identify the probability of a loanee to default a sanctioned loan. It also considered parameters like gender and marital status and spread its results to a large crowd instead of just focusing on the Rich customers of the banks [5]. This research was based on creation of a predictive credit model based on loan status which was done for analysis of defaulters and loyal customers. It also provided comparison between various machine learning techniques and classifiers. It aimed at helping the banks for approving loans of customers who can diligently pay the borrowed sum within the given time and thereby helping banks to avoid huge losses [6].

Another paper referred to proposed an application of classical logistic regression and AUC using Nelder-Mead Algorithm, over an existing model which was used by bank of Indonesia in order to further refine the model and compare the two and predicted defaulting by customers and categorized them into two groups in order to provide a threshold value and accurate results for checking the model's efficiency [7].

Among the considered papers, some provided a method/ approach to create models based on a combination of algorithms that could be used for data cleaning and AUC optimization. One such paper aimed at creating a model based on three different training algorithms and compared it with existing models after being trained and then created two different sets at each level of training. It aimed at improving the results obtained from a predictive model by forming an ensemble model which provided satisfactory results in comparison with the Individual models that were initially being used [8]. Another paper used DSVM model to improve prediction of loan defaulters which could be

implemented by the credit risk management system in banking and lending Institutions [9]. One of the papers considered gave us a better idea about the usage of Regression Trees, multiple splices and how to implement them in prediction [10] and another one gave us a better knowledge about ensemble neural networks and if it can be implemented in our prediction algorithm [11].

All of these researches gave a clearer picture about what is lacking in this domain and could be implemented in order to provide the customers as well as the banks and NBFCs an easier approach to the loan and credit system.

III. Dataset And Procedures

This section gives the reader an insight of the dataset used, how it was created and the classifiers that were tested and implemented. It is further divided into four sub-sections starting with Data preparation that highlights the process of how the data was collected and the dataset was created. It also talks about the parameters that we have additionally created that would improve the accuracy of prediction bythe final model. The second sub-section focuses on the data cleaning process wherein the selection of parameters and the actual parametersused and are present in the compiled dataset have been mentioned. The third part is where the six classifiers that have been tried and tested for the curated dataset have been explained in brief. And the final sub-section is an elaborate explanation of the XGBoost classifierwhich gave the best accuracy for the dataset considered.

A. DATASET GROUNDWORK

The dataset used is a collection of data from various sources taking into account the similarities of parameters between various small datasets and also some real input from our colleagues about their own financial parameters [12].

Novel parameters were created by us to aid the model come to a more accurate conclusion. These parameters were namely Loan to income, Bad state, Available line of credit, Interest paid, EMI paid progress percentage and total repayment progress. Loan to income is a ratio showing how big the loan a person has taken with respect to his earnings (i.e., annual income to loan amount ratio).

Bad state gives a magnitude of how much the repayment has gone off course in terms of ratios. It is the sum total of all the ratios of all the delinquent loans / borrowed sums against their respective amounts invested. This particular parameter helps us to analyze the negative credit history of the given candidate with the parameters given by himand cumulate all of it under a single variable.

bad_state = acc_now_delinquent +

(total_rec_late_fee / funded_amnt_inv) +(recoveries / funded_amnt_inv) +

(collection_recovery_fee / funded_amnt_inv) +

(collections_12_mths_ex_med / funded_amnt_inv).

Available line of credit gives the total number of available/unused credit lines. It is calculated by finding the difference between the parameters, total account value and opened account. Interest paid so far is calculated by summing up Total received interest and Total received late fee. EMI paid progress is found by the ratio of the total payment cleared till the last week to the entire amount to be paid throughout the term. We calculated total repayments received so far, in terms of EMI or recoveries after charge off.

B. Dataset Scrubbing

The dataset considered in this project is a combination of readily available datasets on Google and platforms like Kaggle along with some manually entered data values which was done with the help ofa google form which was circulated among known people.

The entire dataset is of 2.3 GB which consists of 50000+ entries which spans across 33 columns [13] and the parameters that are considered as column values are mentioned in Table 1.

The parameters were selected based on the most frequently appearing parameters in the various datasets that were readily available. Theproblem with considering an individual readily available dataset wasthat it was not contributing efficiently to the purpose of the project.As several projects have been published for the same given problemearlier, but all of these partially focused on the proposed problem. Thus, selection of the right parameters that would effectively andefficiently contribute to the successful completion of this project wasthe main goal.

Table 1: Parameters considered in the dataset

0: Member id	16: Months since last majorderogatory
1: Loan amount	17: Last week payment
2: Funded amount	18: Total current
3: Funded amount invested	19: Total revised high limit
4: Sub Grade: based on loangrading system	20: Total amount collected
5: Interest Rate	21: Recoveries
6: Annual income	22: Recovery fee collected

7: Debt to Income Ratio	23: Term of loan
8: Months since lastDelinquency	24: Account now delinquent
9: Months since last record	25: Loan to income
10: Opened account	26: Bad state
11: Revolving balance	27: Available line of credit
12: Revolving utilization	28: Interest paid
13: Total account value	29: EMI paid progresspercentage
14: Total received interest	30: total repayment progress
15: Total received late fee	

C. Classification algorithms for loan Prediction

The term Classification in Machine learning means approximating a mapping function (f) from input variables (x) to the predicted output variables (y). There are several classification algorithms that are available and in a generalized scenario, it is difficult to quote which oneis the best or is better than the other.

Based on the application of the classifiers for specific dataset in use, itcan be determined which one is superior or better in terms of providingthe best model accuracy than the other, and the corresponding algorithmis used for that particular dataset.

For this particular model, we took into consideration six different classifiers:

I. Random Forest:

An ensemble learning method for classification, regression and othertasks that operate by constructing a number of decision trees at training time and outputting the class that is the mode of classes (classification) or mean prediction (regression) of the individualtrees. It corrects for decision trees' habit of over-fitting to their training set.

II. Decision Tree:

It is a well-known classification technique in different pattern recognition issues like, image classification and character recognition. They perform more successfully, specifically for complex classification problems, due to their high adaptability and computationally effective features. Besides, it also exceeds expectations over numerous typical supervised classification methods.

III. Support Vector Machine

In machine learning, support vector machines are supervised learning

models with associated learning algorithms that analyze data used for classification and regression analysis. However, they are mostly used in classification problems. An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximummarginal hyperplane (MMH).

IV. Logistic Regression:

It is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. The dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X and it is one of the simplest ML algorithms that can be used for various classificationproblems.

V. Gradient Boosting:

These are a group of machine learning algorithms that combine many weak learning models together to create a strong predictivemodel. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular due totheir effectiveness at classifying complex datasets.

VI. XGBoost:

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance that is dominative competitive machine learning. It has recently been dominating applied machine learning and Kaggle competitions for structuredor tabular data.

It is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It provides a parallel tree boosting (also known as GBDT, GBM) that solves many datascience problems in a fast and accurate way [14].

D. XGBoost

It is an acronym for eXtreme Gradient Boosting. The name XGBoost, though, actually refers to the engineering goal to push the limit of computational resources for boosted tree algorithms whichis predominantly the reason for its popularity in many Machine learning projects.

The algorithm differentiates itself in various aspects; It can be usedfor a

wide range of applications such as to solve regression, classification, ranking and user-defined prediction problems.

It is portable and compatible with the major Operating Systems like Windows, MacOS, Linux, etc. It is also capable of providing cloudIntegration support with AWS, Azure and Yarn clusters while alsoworking perfectly fine with Flink, Spark and other ecosystems. Themultiple languages supported by XGBoost includes Java, C++, R, Python, Julia and Scala.

XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods and these apply the principle of boosting weak learners (CARTs generally) while utilizing the gradient descent architecture. However, XGBoost improves the base GBM framework through systems optimization and algorithmicenhancements.

The boosting ensemble technique consists of three simple steps:

i) An initial model F0 which is defined to predict the target variable y. This model will be associated with a residual (y– F0)

ii) A new model h1 is defined which is fit to the residuals obtainedin the previous step

iii) F0 and h1 obtained from the first two steps are combined to generate F1, which is the boosted version of F0. The mean squared error from F1 will be lower than that from F0: $F1(x) <- F0(x) + h1(x)$

For improving the performance of F1, model creation could be done after the residuals of F1 and then create a new model F2: $F2(x) <- F1(x) + h2(x)$

This step can be done for 'm' iterations until residuals have been minimized as much as possible: $Fm(x) <- Fm\text{-}1(x) + hm(x)$

In such a scenario, the additive learners do not disturb the functions that are created in the previous steps. Instead, they simply impart information of their own so as to bring down the errors.

IV. User Friendly Website

Machine learning has become the go to solution for many top companies' problems and they are also using it for improving their services and products.

A simple trained machine learning code is of no use. In order to make the most of it and to test out over real-life values, it can be first trained and then deployed using Streamlit on Heroku cloud.

Streamlit is an open-source app framework which is used to deploy ML models easily.

A. Streamlit Library:

It is essentially used for app creation for any ML based projects using simple python scripts. It also supports hot-reloading which enables the app to live edit and save files.

Using Streamlit API, the app can be built in few lines of code and there is no need to write a backend, define different routes or handle HTTP requests. It is also easy to deploy and manage.

B. Heroku:

Heroku is a Platform as a Service (PaaS). It is essentially a cloud platform which allows anyone to build, operate and run his/her applications in the cloud itself. Apart from being a very extensive and helpful platform, it offers many free plans as well for new account creation on the platform.

Once the dataset is split into train and test values and is successfully implemented using one of the classifiers mentioned above, a new file is created and saved in the same directory and this is used for deployment of the model using Streamlit.

Using the streamlit run "streamlit_app_name".py command, where "streamlit_app_name".py is the file name containing the Streamlit code. On execution of the same, the website opens in the browser and is tested. In order to run the same on Heroku cloud, an account is created and then a new GitHub repository is also created wherein the "streamlit_app_name".py file and the pickled ML model file are added in order to run the web application on cloud.

"Procfile" is also added which contains the code that is used to give commands for executing the files when the application is run. "requirements.txt" contains the packages list and dependencies which are needed in order to run the web application.

"setup.sh" is that file which contains the shell script required to set up the shell environment.

Once all of these files are added, the web app is created by giving the suitable name (we have used loan prediction algorithm as file name) and once created, it is connected to the GitHub repository where the created files were initially added.

After connecting the app to GitHub, automatic deployment was executed.

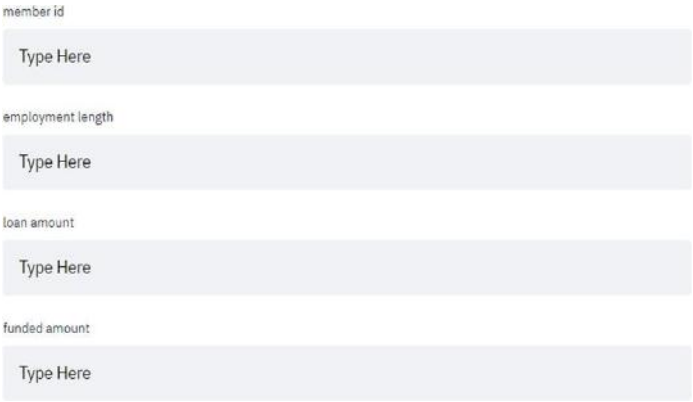


Fig 1: Layout of the created Website Application

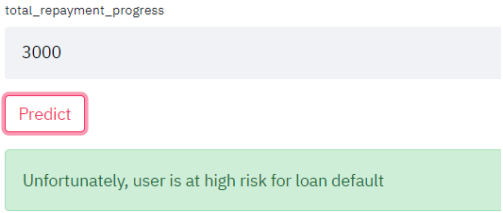


Fig 2: Prediction from the website where user maybe a defaulter

V. Experimental Results

The same dataset was used to check various classifiers and the model accuracy that they returned. The six classifiers that were considered for creation of this model and their resulting model accuracies are mentioned in Table 2.

Table 2: Accuracy% obtained from various classifiers

XGBoost	94.13%
Random forest	92.64%
Decision Tree	89.76%
Support Vector Machine(SVM)	77.03%
Gradient Boosting	66.52%
Logistic Regression	50.93%

The dataset considered was initially split in the ratio 80:20 (train: test)for all the classifiers used. Among the six individually considered classifiers, the highest accuracy was obtained from XGBoost algorithm which was 94.13 % and the lowest was obtained from Logistic Regression classifier which returned an accuracy of 50.93%.Upon obtaining XGBoost as the best classifier

providing the highest model accuracy, different combinations of splitting the dataset into train: test ratio was considered. Table 3 indicates the results obtainedfor various splitting ratios that were tried and tested using XGBoost algorithm.

Table 3: Different Train: Test ratios of the dataset and resulting modelaccuracy (%) and ROC-AUC (%) returned by XGBoost classifier

Train: Test	Model Accuracy	ROC-AUC
80:20	94.13%	92.55%
75:25	94%	92.33%
90:10	93.94%	92.35%
85:15	93.96%	92.31%

VI. Conclusion

This paper proposes a simplified approach at making the loan approval process by the borrower a much easier and less time- consuming process.It majorly focuses at providing a hassle-free and free of cost method thatcan be easily understood and used by the user at his/her convenience.

The research based on previously published papers indicated no such application's development and thus, this paper targeted to bridge that gapin the field of Finance and banking environment.

As there are several classifiers that are available for implementation in order to provide the best model accuracy depending upon the application for which it is being used, this paper majorly focused on tryingout six different classifiers on the cumulated dataset.

A comparison of the model accuracy returned by these six most commonly and popularly used classifiers revealed that the best accuracy obtained was with the implementation of XGBoost classifier which resulted in a model accuracy of 94.13% when the dataset was split in theratio 80:20 (train: test).

Upon the successful training of the model, a website application was created which can be used by any general user.

Keeping in focus that this algorithm was developed to help financial institutions find defaulters, the website will predict if the user whose datais given is a good candidate for loan approval or no.

VII. Future Scope

Perform multiple multivariate analyses to know the important features and to discard the features.

Collect and refine the data to minimize skew columns and skewed variables so that they do not affect the algorithm.

Perform multiple runs with different feature sets and obtain scatter plots for different results algorithm to find the best fit algorithm for this used case.

Collect an extensive dataset to give enough room for the algorithm to get trained and give accurate results.

Collaborate with Banks / TransUnion CIBIL to implement this project at a prolific level with a streamlined dataset and improved accuracy.

Develop an algorithm for financial institutions to check if the loanee is credit worthy or not.

VIII. References

1. Asia Samreen, Farheen Batul Zaidi, Aamir Sarwar: "Design and Development of Credit Scoring Model for Commercial Banks in Pakistan: Forecasting Creditworthiness of Corporate Borrowers"; International Journal of Business and Commerce, Vol 2, No 5, 2013.
2. Kumar Arun, Garg Ishan, Kaur Sanmeet: "Loan Approval Prediction based on Machine Learning Approach"; ISOR Journal ofComputer Engineering (ISOR-JEC), 2016.
3. Kumar, R., Jain, V., Sharma, P. S., Awasthi, S., & Jha, G. (2019). Prediction of Loan Approval using Machine Learning. International Journal of Advanced Science and Technology, 28(7), 455 - 460.
4. Pidikiti Supriya, Myneedi Pavani, Nagarapu Saisushma, Namburi Vimala Kumari, K Vikas"Loan Prediction by using Machine Learning Models" Volume 5 - Issue 2 (144-148) 2019, International Journal of Engineering and Techniques (IJET), ISSN:2395-1303.
5. M. A. Sheikh, A. K. Goel and T. Kumar, "An Approach forPrediction of Loan Approval using Machine Learning Algorithm," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020.
6. G. Arutjothi and C. Senthamarai, "Prediction of loan status in commercial bank using machine learning classifier," 2017 InternationalConference on Intelligent Sustainable Systems (ICISS), Palladam, 2017.
7. H. Sutrisno and S. Halim, "Credit Scoring Refinement Using Optimized Logistic Regression," 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT), Denpasar, 2017.

8. Amira Kamil Ibrahim Hassan, Ajith Abraham: " Modelling Consumer Loan Default using Ensemble Neural Networks "; Department of Computer Science - University of Science and Technology, Sudan, Machine Intelligence Research Labs - WA, USA; 2009.
9. A. Shivanna and D. P. Agrawal, "Prediction of Defaulters using Machine Learning on Azure ML," 2020 11th IEEE Annual InformationTechnology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2020.
10. Tian-Shyug Lee: "Mining the Customer Credit Using Classification and Regression Tree and Multivariate Adaptive Regression Splines"; DBLP, Conference Paper, January 2003.
11. Ibrahim Hassan, Abraham Shaikh: "Modelling Prediction Algorithms using Ensemble Neural Networks"; Department of Computer Science - University of Science and Technology, Sudan, Machine Intelligence Research Labs - LA, USA; 2010.
12. Sihem Khemakhem, Younes Boujelbena: "Credit Risk Prediction - A comparative study between discriminant analysis and theneural network"; Accounting and Management Information Systems, Vol 14, 2015.
13. Arun Iru, Ishanya Sabarval, Kaur Avneet: "Preparation of database based on Machine Learning Approach"; ISOR Journal of Computer Engineering (ISOR-JEC), 2017.
14. Rising Odegua: "Predicting Bank Loan with Extreme Gradient Boosting"; Department of Computer Science - Ambrose Alli University-Nigeria, 2018.

Chapter - 2

Computer Science and Game Theory

Dr. Rajesh Patel
Assistant Professor, Department of Computer Engineering, Sankalchand Patel College of Engineering, Sankalchand Patel University, Visnagar, Gujarat, India

Abstract

Computer Science and Game Theory is an interdisciplinary field that combines concepts and techniques from computer science and game theory to study strategic decision-making in computational systems. This field explores the interactions between computational agents, the analysis of algorithmic strategies, and the design of computational mechanisms to achieve desirable outcomes. This abstract provides an overview of Computer Science and Game Theory, highlighting its key concepts and applications, Machine learning and reinforcement learning techniques, Network Design, Game Theory in Security and Privacy, and Game-Theoretic Approaches to Market Design.

Keywords: Computer Science, Game Theory, Computational Systems, Strategic Decision-Making, Interdisciplinary Field, Computational Agents, Algorithmic Strategies, Computational Mechanisms, Desirable Outcomes, Market Design, Market Mechanisms.

1. Introduction

In recent years, the intersection of computer science and game theory has become an increasingly important field of study. This chapter aims to explore the relationship between these two disciplines and shed light on how computer science techniques and algorithms have influenced game theory and vice versa. By understanding this synergy, we can gain valuable insights into the development of intelligent systems, decision-making processes, and strategic thinking.

1.1. The Foundation of Game Theory

Game theory, pioneered by mathematicians John von Neumann and Oskar Morgenstern in the mid-20th century, is a framework for analyzing strategic interactions between rational decision-makers. It provides a

mathematical model to study conflicts of interest, decision-making strategies, and the consequences of these choices. Game theory has been widely applied in various fields, including economics, politics, biology, and sociology.

Game theory is a branch of mathematics that studies strategic decision-making. It provides a framework for analyzing interactions between rational individuals or entities with conflicting interests. This chapter aims to provide an overview of the foundation of game theory, its key concepts, and applications.

Definition of Game Theory: Game theory is the study of mathematical models of strategic interaction between rational decision-makers. It seeks to understand how individuals or entities make choices and how those choices affect outcomes. It involves analyzing the behavior of players, their strategies, and the possible outcomes of their interactions.

1.2. Key Concepts in Game Theory:

1. Players: In game theory, players are the participants who make decisions. They can be individuals, companies, or even countries. Each player has their own set of possible actions or strategies.
2. Strategies: Strategies are the choices or actions available to players. Players select strategies based on their preferences and the information available to them.
3. Payoffs: Payoffs represent the outcomes or rewards associated with each combination of strategies chosen by the players. Payoffs can be positive or negative, depending on the desired outcome.
4. Nash Equilibrium: Nash equilibrium is a concept in game theory where no player has an incentive to change their strategy, given the strategies of other players. It represents a stable state where each player's strategy is the best response to the strategies chosen by others.

1.3. Applications of Game Theory:

Game theory has diverse applications in various fields, including economics, political science, biology, and computer science. Some notable applications include:

1. **Economics:** Game theory is extensively used in economics to analyze market behavior, pricing strategies, and competition between firms. It helps economists understand strategic interactions

and predict outcomes in different economic scenarios.

2. **Political Science:** Game theory is applied in political science to analyse voting behaviour, coalition formation, and negotiations between political parties or countries. It helps in understanding the strategic decisions made by politicians and their impact on the political landscape.
3. **Biology:** Game theory provides insights into the behaviour of organisms in evolutionary biology. It helps analyze mating strategies, predator-prey interactions, and the evolution of cooperation or competition among species.
4. **Computer Science:** Game theory plays a crucial role in computer science, especially in the field of artificial intelligence and multi-agent systems. It helps in designing intelligent systems that can make optimal decisions in complex and dynamic environments.

1.4. Computer Science and Game Theory: A Symbiotic Relationship

Computer science has greatly impacted game theory, enabling researchers to analyze complex games and optimize decision-making processes. Here are some key areas where computer science has made significant contributions:

Algorithmic Game Theory

Algorithmic game theory is the study of algorithms and computational models for analysing strategic interactions. It combines computer science's algorithmic techniques with game theory's strategic insights. Researchers have developed algorithms for various game-theoretic problems, such as finding Nash equilibria, computing optimal strategies, and analysing the stability of cooperative outcomes. These algorithms have practical applications in designing efficient market mechanisms, resource allocation, and automated negotiation systems.

Algorithmic Game Theory is an interdisciplinary field that combines elements of computer science, game theory, and economics. It focuses on designing algorithms and computational models to analyse strategic interactions and decision-making in complex systems. This chapter aims to provide an overview of Algorithmic Game Theory, its key concepts, and applications.

Definition of Algorithmic Game Theory: Algorithmic Game Theory studies the computational aspects of strategic interactions. It explores how algorithmic techniques can be applied to analyse and optimize decision-

making in game-theoretic settings. The field seeks to understand the computational complexity of finding optimal strategies, mechanisms, and equilibrium in various game-theoretic scenarios.

1.5. Key Concepts in Algorithmic Game Theory:

1. **Algorithmic Mechanism Design:** Algorithmic Mechanism Design aims to design algorithms and mechanisms that incentivize participants to behave in certain desirable ways. It involves creating rules and incentives to elicit truthful information and ensure desirable outcomes in strategic interactions.
2. **Computational Complexity of Games:** Algorithmic Game Theory analyses the computational complexity of finding equilibria, optimal strategies, and solutions in games. It explores the trade-offs between computational efficiency and solution quality.
3. **Price of Anarchy:** The Price of Anarchy measures the inefficiency caused by selfish behaviour in a system. It quantifies the difference between the social welfare achieved in a Nash equilibrium and the optimal social welfare that could be achieved with a centralized decision-maker.
4. **Algorithmic Auctions:** Algorithmic Auctions involve designing algorithms for conducting auctions and optimizing the allocation of resources. It explores different auction formats and mechanisms to ensure fairness, efficiency, and truthful bidding.

1.6. Applications of Algorithmic Game Theory: Algorithmic Game Theory has various applications in different domains, including:

1. **Online Advertising:** Algorithmic Game Theory plays a crucial role in online advertising platforms where advertisers bid for ad placements. It helps in designing auction mechanisms to maximize revenue for the platform while maintaining fairness and efficiency.
2. **Network Routing and Resource Allocation:** Algorithmic Game Theory provides algorithms and models for efficient routing and resource allocation in computer networks. It considers the selfish behaviour of network users and aims to optimize resource utilization.
3. **Market Design:** Algorithmic Game Theory is applied in market design to analyse and improve the functioning of markets. It helps in designing mechanisms that encourage competition, ensure fair pricing, and promote market efficiency.

4. **Social Networks and Influence Maximization:** Algorithmic Game Theory explores strategic interactions in social networks and studies how influence can be maximized. It analyses viral marketing, opinion formation, and the spread of information in social networks.

Algorithmic Game Theory combines the power of algorithms and computational models with strategic decision-making. It provides insights into the computational complexity of games, mechanism design, and the price of anarchy. The field has numerous applications in online advertising, network routing, market design, and social networks. Understanding Algorithmic Game Theory can lead to more efficient and effective decision-making in complex systems.

2. Machine Learning and Reinforcement Learning

Machine learning and reinforcement learning techniques have revolutionized game-playing and strategic decision-making. By using algorithms that learn from data, computers can now surpass human players in complex games like chess, Go, and poker. Reinforcement learning algorithms enable agents to learn optimal strategies through trial and error, often achieving superhuman performance. These advancements have practical implications in autonomous systems, robotics, and strategic planning.

Introduction: Machine Learning (ML) and Reinforcement Learning (RL) are two branches of artificial intelligence that have gained significant attention and have revolutionized various domains. This chapter aims to provide an overview of Machine Learning and Reinforcement Learning, their key concepts, and their applications.

Machine Learning: Machine Learning is the field of study that focuses on developing algorithms and models that allow computers to learn from data and make predictions or decisions without being explicitly programmed. It involves training models on existing data to recognize patterns, generalize from examples, and make accurate predictions on new, unseen data.

2.1. Key Concepts in Machine Learning:

1. **Supervised Learning:** In Supervised Learning, a model learns from labeled examples, where the desired output is provided for each input. The model generalizes from these examples to make predictions or classifications on new, unseen data.
2. **Unsupervised Learning:** Unsupervised Learning involves learning from unlabeled data, where the model identifies patterns, structures,

or clusters in the data without any predefined labels or outputs.

3. **Deep Learning:** Deep Learning is a subset of Machine Learning that focuses on training deep neural networks with multiple layers to learn hierarchical representations of data. It has achieved remarkable success in various domains, including computer vision and natural language processing.

Reinforcement Learning: Reinforcement Learning is a type of Machine Learning where an agent learns to interact with an environment and make sequential decisions to maximize a cumulative reward. The agent explores the environment, receives feedback in the form of rewards or penalties, and learns to take actions that lead to favorable outcomes.

2.2. Key Concepts in Reinforcement Learning:

1. **Agent:** The agent is the entity that interacts with the environment in Reinforcement Learning. It takes actions based on its current state and receives feedback in the form of rewards or penalties.

2. **Environment:** The environment represents the external system with which the agent interacts. It provides the agent with information about its current state and gives feedback in the form of rewards or penalties based on the agent's actions.

3. **Reward:** Rewards are the feedback signals provided to the agent to indicate the desirability of its actions. The agent's objective is to maximize the cumulative reward it receives over time.

2.3. Applications of Machine Learning and Reinforcement Learning: Machine Learning and Reinforcement Learning have a wide range of applications, including:

1. **Natural Language Processing:** Machine Learning techniques, including deep learning, have greatly advanced natural language processing tasks such as speech recognition, machine translation, and sentiment analysis.

2. **Computer Vision:** Machine Learning algorithms, such as convolutional neural networks, have revolutionized computer vision tasks, including image classification, object detection, and image generation.

3. **Autonomous Systems:** Reinforcement Learning is applied to train autonomous systems, such as self-driving cars or robots, to make intelligent decisions in dynamic environments.

4. **Recommendation Systems:** Machine Learning is used to develop personalized recommendation systems that suggest products, movies, or content based on user preferences and behavior.

3. Game Theory in Network Design

Computer networks are fundamental to modern communication systems. Game theory has been instrumental in designing efficient protocols, optimizing network traffic, and preventing congestion. By modeling the interactions between network nodes as strategic games, researchers can analyze the stability of network equilibria, devise incentive mechanisms, and improve overall network performance. These insights have paved the way for scalable and robust network architectures.

Game theory provides a powerful framework for analyzing strategic interactions among rational decision-makers. When applied to network design, game theory offers valuable insights into the behavior of self-interested entities and helps optimize the design and operation of complex networks. This chapter aims to explore the application of game theory in network design, highlighting its key concepts and implications.

Understanding Game Theory in Network Design: Game theory in network design focuses on analyzing the behavior of entities, such as users or service providers, who make strategic decisions that affect the performance and efficiency of a network. Key concepts involved in this domain include:

1. **Players:** In network design, players refer to the entities or stakeholders involved in the decision-making process. They can be users, network operators, or even competitors in the market.

2. **Strategies:** Strategies represent the choices available to the players. In the context of network design, strategies can include selecting specific routes, determining the allocation of resources, or deciding on pricing mechanisms.

3. **Payoffs:** Payoffs reflect the outcomes or rewards associated with the strategies chosen by the players. In network design, payoffs can include factors such as network performance, cost efficiency, or market share.

4. **Equilibrium:** Equilibrium in game theory refers to a state where no player has an incentive to deviate from their chosen strategy, given the strategies of other players. Equilibrium concepts such as Nash equilibrium or Stackelberg equilibrium help analyze stable outcomes in network design.

3.1. Applications of Game Theory in Network Design:

Game theory finds various applications in network design, including:

1. **Routing Optimization:** Game theory helps optimize routing protocols by considering the self-interested behaviour of network nodes. It addresses issues such as congestion, load balancing, and the efficient utilization of network resources.
2. **Spectrum Allocation:** In wireless communication networks, game theory aids in the allocation of limited spectrum resources among multiple users or service providers. It considers factors such as interference, spectrum pricing, and cooperation strategies.
3. **Network Security:** Game theory contributes to network security by modeling the interactions between attackers and defenders. It helps identify optimal defense strategies, understand the dynamics of cyber threats, and design robust security mechanisms.
4. **Internet Economics:** Game theory plays a vital role in analyzing economic aspects of network design, such as pricing models, service differentiation, and competition among service providers. It helps in understanding market dynamics and designing efficient mechanisms.

5. Game Theory in Security and Privacy:

This topic applies game theory to analyze security and privacy challenges in computational systems. It investigates strategic interactions between attackers and defenders, incentives for cooperation in security measures, and privacy-preserving mechanisms.

Game theory has emerged as a valuable tool for analyzing strategic interactions and decision-making processes in the realm of security and privacy. It provides a framework to model and understand the behaviors of different entities, such as attackers, defenders, and users, and offers insights into optimal strategies and outcomes.

In the context of security, game theory helps to analyze the dynamics between attackers and defenders. It models their interactions as a game, where each player selects actions to maximize their payoff or utility. Attackers seek to exploit vulnerabilities and compromise security, while defenders aim to protect systems and prevent unauthorized access.

One application of game theory in security is the study of optimal defense strategies. By considering the potential actions and motivations of

attackers, defenders can strategically allocate resources, implement security measures, and adapt their defense mechanisms. Game theory helps identify equilibrium points, such as Nash equilibria, where both attackers and defenders have no incentive to deviate from their chosen strategies.

Privacy is another domain where game theory plays a crucial role. With the increasing digitization of personal information and the growing concern for data privacy, understanding the interactions between users and data collectors becomes essential. Game theory models can capture the trade-offs users face when deciding whether to disclose personal information or engage in privacy-protecting behaviors.

By employing game-theoretic frameworks, researchers can analyze scenarios such as online advertising, social networks, and data-sharing platforms. These models enable the examination of privacy-preserving mechanisms, incentive structures, and the effects of strategic behaviors on individual privacy and societal outcomes.

Overall, game theory provides a powerful toolkit for understanding strategic decision-making in security and privacy. It assists in formulating effective defense strategies, evaluating risk and payoff trade-offs, and designing mechanisms that encourage privacy-conscious behaviors. Its application in these domains continues to advance our understanding of complex security and privacy challenges in an increasingly interconnected world.

Game theory offers valuable insights and analytical tools for addressing security and privacy challenges. By modeling strategic interactions, analyzing incentives, and designing mechanisms, game theory helps in understanding the dynamics of security and privacy domains. It enables the development of robust defense strategies, incentivizes cooperation among stakeholders, and facilitates the design of privacy-preserving mechanisms. Incorporating game-theoretic approaches can enhance the security and privacy of systems in a world with ever-evolving threats.

6. Game-Theoretic Approaches to Market Design:

This topic applies game theory to study market design, analyzing mechanisms for resource allocation, pricing, and market dynamics. It considers issues such as market efficiency, fairness, and strategic behavior of market participants.

This section discusses game-theoretic models used in market design. It introduces concepts such as auctions, bargaining, and mechanism design,

which provide a foundation for understanding strategic decision-making in markets. It explores how game theory can be applied to model the behavior of buyers, sellers, and intermediaries in diverse market settings.

Market Mechanisms and Allocation: This section explores the design of market mechanisms for resource allocation. It discusses how game theory can inform the design of efficient and fair allocation mechanisms, such as combinatorial auctions, matching markets, and spectrum auctions. It analyzes the strategic considerations of participants and aims to optimize allocation outcomes.

Pricing Strategies and Revenue Maximization: In this section, the focus is on pricing strategies and revenue maximization in markets. Game theory provides insights into pricing decisions, including dynamic pricing, price discrimination, and auction mechanisms that optimize revenue generation. It considers the strategic behavior of participants and the impact of pricing strategies on market dynamics.

Competition and Market Dynamics: This section discusses how game theory helps analyze competition and market dynamics. It explores models of strategic behavior among firms, the impact of market structure on competition, and the design of mechanisms that promote competition and prevent anti-competitive practices. It considers concepts such as market entry, collusion, and strategic positioning.

Case Studies and Applications: This section presents case studies and real-world applications that demonstrate the effectiveness of game-theoretic approaches in market design. It includes examples such as electronic marketplaces, spectrum auctions, and online advertising platforms. The case studies highlight how game theory can aid in understanding market dynamics, designing efficient mechanisms, and optimizing outcomes in diverse market environments.

References

1. Azeem, M., Malik, H. A., Javaid, A., & Chaudhary, M. A. (2020). Algorithmic Game Theory for Optimizing Market Mechanisms. IEEE Access, 8, 112287-112304. DOI: 10.1109/ACCESS.2020.3006694
2. Azeem, M., Khan, A. S., & Hussain, I. (2020). Game theory: a comprehensive introduction. John Wiley & Sons.
3. Russell, S., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach (3rd ed.). Pearson.
4. Nisan, N., Roughgarden, T., Tardos, É., & Vazirani, V. V. (2007).

Algorithmic game theory. Cambridge University Press.

5. Shoham, Y., & Leyton-Brown, K. (2009). Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press.
6. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT Press.
7. Osborne, M. J., & Rubinstein, A. (1994). A Course in Game Theory. MIT Press.
8. Osborne, M. J., & Rubinstein, A. (1994). A course in game theory. MIT Press.
9. Roughgarden, T. (2016). Twenty lectures on algorithmic game theory. Cambridge University Press.
10. Papadimitriou, C. H. (2007). Algorithms, Games, and the Internet. In Algorithms, Games, and the Internet (pp. 1-15). Springer.
11. Cavusoglu, H., & Raghunathan, S. (2002). Economics of information security and privacy. Springer.
12. Tirole, J. (2017). Economics for the common good. Princeton University Press.

Chapter – 3

Introduction about Html and CSS

Raj Kumar Gupta and Sumit Kumar
Assistant Professor, MCA Department, Noida Institute of Engineering and Technology Greater Noida, Uttar Pradesh, India

Introduction to Web Development Strategies

Websites are always developing with novel tools and features for users. All this is built into the web design strategies that were put into effect from the start and are currently being worked on to ensure their continued functioning. These represent a few of the modern web development tactics that a company that develops websites should always be conscious of at the beginning of their project.

A successful website does three things:

- It generates the appropriate attendees.
- Showcase your main services or commodities.
- Collect contact data to build ongoing interactions.

Website Development Strategy Tips for Engineers

These are explained below.-

1. Unimportant visuals and phrases.

Sites are easier to index and crawl when every component are correctly defined in the web creating process. Therefore, HTML tags encompass photo tags, meta descriptions, tag titles, and canonical anchor text. They are pretty accurate and indicative of a website's design and general operation. It results in web pages that are clear, developed, and totally searchable using corresponding search clauses.

2. Utilising Social Media Posting References

The use of social media is now prevalent in practically every area and degree of human effort. This suggests that there is access to a massive market or audience, offering many opportunities and possibilities.

3. Incorporate calls to action (CTAs).

CTAs in web design are crucial in converting visitors to possible and

actual clients. This is why CTAs urge visitors to turn each search or scroll throughout into some sort of proactive act. Online groups and firms using call to action (CTA) buttons have higher average rates of conversion (CR) compared with those without it.

4. Exact Imagery to Serve the Audience of Interest

Customers constantly look for what they are interested in or looking for, and developers of websites ought to keep doing the exact same at all phases of development. Substance photos on a website provide a distinctive message & are easier for readers to relate to than text. That's why appropriate illustrations convey an appropriate data or message. This story is more fully engaging for that specific demographic.

5. Webpage Accessibility

Compared to any other web design technique, website navigation is a crucial one to consider and to enhance at all times. Continuous improvement and creativity are essential inputs that insure sites are always available, accessible and easy to navigate. Better and better site navigation allows visitors and users to look up websites for longer periods of duration, giving them a greater belief in where they are, what they doing, and the overall layout of the sites. Good navigation permits users and visitors to look up information with simplicity.

History of the Internet

The Internet started with studies on what was then referred to as packet switching, in the 1960s. The packet switching method was seen as a better and faster way of transmitting information than the hardware that addressed the problem, i.e., devices. The American Military relied significantly on packet switching devices when building the ARPANET network the ARPANET initiative is widely regarded as the earliest known network of computers that are networked, often known as the internet. The method was used to exchange highly classified data all over the military. This data-sharing method was then made available for educational institutes in the United States, giving them to connect to the government's microprocessor at 56 kbit/s, 1.5 Mbit/s, and 45 Mbit/s. Com Internet service suppliers arose in the late 1980s, but the internet became totally commercialized in the USA by 1995.

History of the Web

British scientist Tim Berners-Lee developed the World Wide Web (WWW) in 1989 while employed at CERN. The Web was designed and built to meet the demand for automated sharing of data among scientists at colleges and universities across the globe.

CERN is not a solitary laboratory, instead being the hub of a large community involving over 17,000 academics from over 100 nations. Even though they spend a period at CERN, researchers usually work at colleges and national laboratories in their native countries. Reliable communication mechanisms are consequently required.

The main goal of the WWW was to connect the growing technologies of computers, data networks, and web into a powerful and easy to use worldwide database.

Protocol governing the Web

1. TCP/IP (Transmission Control Protocol/Internet Protocol).

These are a collection of common regulations that enable many types of computers to interact with one another. The protocol known as IP guarantees that each computer related to the Internet is assigned an individual serial number known as the IP address. TCP defines how data is transferred over the internet as well as how it is divided into IP packets. It ensuring that the packets contain information about the message material's source, destination, and the order in which the information contained in the message should be re-assembled, in addition to checks to see if the message was correctly delivered for the specified destination. TCP is also referred to as a protocol that focusses.

2. SMTP (Simple Email Transfer Protocol)

These standards are crucial for sending or distributing incoming emails. The protocol in question uses the mail headers to obtain the receiver's email handle and places the communication in the incoming mail queue. And right away as the email gets sent to the recipient's mailbox, it is removed out of the out list. The communication or electronic email can include text, video, pictures, etc. It aids to establish connection server settings.

3. PPP (Point-to-Point Protocol)

It is an exchange protocol that creates a direct link among both devices that communicate. This protocol creates the rules for verifying and sharing information between both devices. For example, someone connects his PC with an Internet Service Provider's computer while also using PPP. Similarly, PPP can be utilised to connect multiple routers simultaneously.

4. File Transfer Protocol (FTP)

This type of protocol is employed to send information from one machine to other. This works on an architecture with clients and servers. When a computer requests a file transfer from another machine, the FTO creates a

connection between the two and authenticates it using their username and password. And the requested file transfer happens between each computer.

5. Secure File Transfer Protocol (SFTP)

SFTP, frequently referred to as SSH FTP, relates to the File Transfer Protocol (FTP) over a secure shell (SSH), which encrypts data as well as instructions while transmission. SFTP works like an extension of SSH, encrypts files and data while sending them via a secure shell data stream. This method is used to establish links to other systems while executing command-line operations.

6. HTTP (Hypertext Transfer Protocol)

The HTTP protocol is used for transmitting hypertext across the internet and is defined by the web (World Wide Web) to facilitate data transfer. This protocol regulates how information ought to be formatted and moved. It also defines the different steps that web browsers should take in reply to requests for visiting specific web pages. When a person starts a browser for the web, they will indirectly use HTTP, which is the protocol intended to share text, images, and other multimedia stuff on the net.

7. HTTPS (Hyper Text Transfer Protocol Secure)

HTTPS is a modification to the hypertext transfer protocol (HTTP). It is used for safe communication across an electronic network, utilising the SSL or TLS encryption protocols for encryption and identification. So, in general, a web page uses the HTTP protocol; however, if the website gets private data such as credit card details, debit card specifics, OTP, and many more, an SSL certificate must be placed in order to render the page more secure. So, when entering any private data on a website, we ought to check whether the link is HTTPS or not. If it does not have HTTPS, it could possibly not be secure enough to enter critical information.

8. TELNET (Terminal Network)

ISO defines TELNET as the accepted TCP/IP protocol for virtual terminal services. This allows a single neighbourhood machine to connect to another. The machine that is being connected is known as a distant machine, and the computer connected is known as the local computer. TELNET lets us to display what happening on the remote computer onto the one nearest to us. It relies on the client/server principle. The local computer utilises the telnet client program, while the other computer uses the tcp servers program.

9. POP3 (Post Office Protocol 3)

POP3 stands for Post Office Protocol Version 3. It has two

Communication Access Agents (MAAs), one client MAA (Message Access Agent) and the other service MAA (Message Access Agent) for accessing messages in the mailbox. This protocol allows the retrieval and handling of emails from the recipient's mail server's mailbox to the receiver's laptop. This is an implicit agreement between the receiver and the recipient's mail server. It is referred to as a client-server protocol that only interacts in one of the directions. The POP3 works on two ports: 110 and 995.

What is HTML

HTML (Hypertext Markup Language) is the standard markup language utilised to construct internet pages and content for display on the World Wide Web. It provides an assortment of elements or tags for structuring substance on a web page. These components specify the structure, design, and semantic purpose of the substance in question.

HTML is the base of web pages that web browsers use to render and display content. It creates rich or dynamic web experiences by combining the use of CSS with JavaScript for responsiveness.

Here are a few key points about HTML:

1. **Markup Language**: HTML defines website elements utilising tags found inside brackets with angles ("<" and ">"). These components, which include headings, sections, links, pictures, and lists, show the format and purpose of the written content.
2. **Structure:** HTML allows you to organise content in an ordered way. For example, users can use heading tags like <h1>, <h2>, <h3>, etc., to construct headers of various levels or use <p> tags to sections of content.
3. **Hyperlinks**: HTML contains the <a> (anchor) component to generate hyperlinks. This allows consumers to move among the pages by tapping on link.
4. **Multimedia**: HTML permits you to put multimedia assets that include photographs & movies on an internet page. The <img> tag is used for images, and the <video> and <audio> elements are used to incorporate both audio and video knowledge separately.
5. **Forms**: HTML includes elements to build interactive forms. Use <form>, <input>, <textarea>, and various other form-related elements to collect user input, such text, checkboxes, choices, and more.

HTML TAGS

HTML (Hypertext Markup Language) uses tags to structure web content. Here are some common HTML tags:

1. **<html>:** The root element that wraps all content on a web page.
2. **<head>:** Contains metadata about the document, like the title, character encoding, and linked stylesheets.
3. **<title>:** Sets the title of the web page, displayed in the browser tab.
4. **<meta>:** Provides metadata about the document, such as character encoding or author information.
5. **<link>:** Used to link external resources like stylesheets.
6. **<style>:** Contains inline CSS for styling elements.
7. **<script>:** Embeds JavaScript code or links to external script files.
8. **<body>:** Contains the visible content of the web page.
9. **<p>:** Defines a paragraph.
10. **<a>:** Creates hyperlinks to other web pages or resources.
11. **<img>:** Embeds images.
12. **<ul>:** Creates an unordered (bulleted) list.
13. **<ol>:** Creates an ordered (numbered) list.
14. **<li>:** Defines list items within <ul> or <ol>.
15. ** :** A container for grouping and styling elements.
16. **:** Inline container for applying styles to a specific portion of text.
17. **<table>:** Defines a table.
18. **<tr>:** Defines a table row.
19. **<td>:** Defines a table cell.

HTML List

HTML lists allow web developers to group a set of related items in lists.

Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default.The example is given below:

Book

Copy

Pencil

Ordered HTML List

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default.The example is given below:

Coffee

Tea

Milk

HTML IMAGES

In HTML, you can display images using the <img> element. Here's a basic example of how to do it:

<img src="pic_trulli.jpg" alt="Italian Trulli">

src: This attribute specifies the path to the image file. It can be a relative or absolute URL.

alt: This feature provides take text for the image, it shows up if the image cannot be loaded or for accessible considerations.

HTML Links - Hyperlinks

HTML links are hyperlinks. You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

HTML Links - Syntax

The HTML <a> tag defines a hyperlink. It has the following syntax:

link text

Example

Visit google.com!

HTML TABLE

<table> defines the table.

<tr> defines a table row.

<th> defines table headers (usually displayed in bold).

<td> defines table data cells.

The example is given

<table>

<tr>

<th>Header 1</th>

<th>Header 2</th>

</tr>

<tr>

<td>Data 1</td>

<td>Data 2</td>

</tr>

<tr>

<td>Data 3</td>

<td>Data 4</td>

</tr>

</table>

HTML IFRAME

An HTML <iframe> (short for inline frame) embeds a different website or documents within the current web page. It lets you to show articles from yet another source on your online presence. Here is a straightforward example of how to utilise it:

<iframe src="https://www.example.com"></iframe>

In this example, the content of the example.com website will be displayed within the <iframe> on your web page. You can customize the attributes of the <iframe> to control its size, border, and other properties.

HTML FORMS

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The <form> Element

The HTML <form> element is used to create an HTML form for user input:

<form>

.

form elements

.

</form>

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable

Introduction of CSS

CSS, or Cascading Style Sheets, is an important technology used in the development of websites. It governs the visual display and arrangement of web pages, enabling developers and designers of websites to style elements using HTML.

CSS defines requirements for how elements in HTML should be presented, like colours, fonts, distance, and location. The regulations are made up of selections (which target HTML elements) and declarations (which describe the stylistic variables and expressions).

CSS syntax:

```
selector {
property: value;
another-property: another-value;
}
```

Example:

```
p {
color: red;
text-align: center;
}
```

External style sheet using link

To create an external style sheet using the "link" element in HTML, follow these steps:

1. **Create a New CSS File:**

Open a text editor such as Notepad, Visual Studio Code, or any code editor of your choice.

Create a new file and save it with a .css extension. For example, you can name it "styles.css."

2. **Write Your CSS Rules:**

Inside the CSS file, write your CSS rules to style your HTML elements. For example:

```
/* styles.css */
body {
font-family: Arial, sans-serif;
background-color: #f0f0f0;
}
h1 {
color: #333;
}
/* Add more styles as needed */
```

3. **Link the CSS File to Your HTML Document**:

In your HTML document (e.g., index.html), add the "link" element inside

the "head" section to link the external CSS file. Specify the "rel" attribute as "stylesheet," the "type" attribute as "text/css," and the "href" attribute as the path to your CSS file:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
<h1>Welcome to My Website</h1>
<p>This is a sample paragraph.</p>
<!-- Your HTML content here -->
</body>
</html>
```

4. **Save and Load:**

   Save both your HTML and CSS files.

   Open the HTML file in a web browser, and it will apply the styles from your external CSS file.

Multiple style sheet

To link multiple guidelines in HTML, utilise <link> components in the <head> section. Styles ought to be organised in various files, and remember to cascade in the order that they're linked. Use IDs & classes to choose specific products.

Css Selectors

A CSS selector selects the HTML element(s) you want to style.

Here are some common CSS selectors:

1. **Element Selector**: Selects all instances of a specific HTML element. For example, p selects all <p> elements. Another example is given as:

```
p {
  text-align: center;
  color: red;
```

}

2. **Class Selector**: Selects elements with a specific class attribute. For example, .my-class selects all elements with class="my-class".

   Other example is given:

   .center {

   text-align: center;

   color: red;

   }

3. **ID Selector**: Selects a single element with a specific ID attribute. For example, #my-id selects the element with id="my-id". Another example is given:

   #para1 {

   text-align: center;

   color: red;

   }

4. **Universal Selector**: Selects all elements on the page. It is denoted by an asterisk (*).

   The example is:

   * {

   text-align: center;

   color: blue;

   }

5. **Descendant Selector**: Selects elements that are descendants of a specific element. For example, ul li selects all <li> elements that are descendants of a <ul> element.

6. **Child Selector**: Selects elements that are direct children of a specific element. For example, ul > li selects all <li> elements that are direct children of a <ul> element.

7. **Adjacent Sibling Selector**: Selects an element that is immediately preceded by a specific element. For example, h2 + p selects a <p> element that directly follows an <h2> element.

8. **Attribute Selector**: Selects elements with a specific attribute or attribute value. For example, [type="text"] selects all elements with type="text".

9. **Pseudo-class Selector**: Selects elements based on their state or position. For example, :hover selects an element when the mouse pointer is over it.
10. **Pseudo-element Selector:** Selects a specific part of an element. For example, ::before selects the content before an element.

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

1. **External CSS-**With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.The example is:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

2. **Internal CSS-**An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.The example is given as:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
```

```
color: maroon;
margin-left: 40px;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

3. **Inline CSS-**An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.The example is given as:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
</body>
</html>
```

The CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

Content - The content of the box, where text and images appear.

Padding - Clears an area around the content. The padding is transparent.

Border - A border that goes around the padding and content.

Margin - Clears an area outside the border. The margin is transparent.

The box model allows us to add a border around elements, and to define space between elements. The example is:

```
div {
width: 300px;
border: 15px solid green;
padding: 50px;
margin: 20px;
}
```

Introduction to bootstrap

Bootstrap is a frequently employed open-source front-end and back framework. This is intended for the development of websites. It offers an array of pre-designed, receptive, and adaptable components and styles. This allows developers to quickly and simply construct modern yet uniform interfaces for the web. The community of open-source software currently maintains Bootstrap, which had originally been created by Twitter.

Basic Key features of Bootstrap include:

Responsive Design: Bootstrap's grid system and components are built to adapt to various screen sizes, making it mobile-friendly.

CSS Framework: It offers a collection of CSS classes and styles for typography, buttons, forms, navigation bars, and more. They are useful for large projects.

JavaScript Components: Bootstrap includes interactive elements like modal dialogs, tooltips, carousels, and more, enhanced with JavaScript.

Customization: Developers can customize Bootstrap to match the project's design by modifying variables or using the built-in theming tools.

Browser Compatibility: Bootstrap aims for compatibility with popular web browsers, ensuring a consistent experience for users.

To get started with Bootstrap, you typically include its CSS and JavaScript files in your HTML document, and then you can use its classes and components to build responsive and visually appealing websites. The official Bootstrap documentation provides comprehensive guidance and examples for using the framework effectively.

Reference

1. Ivan Vayroys, www.google.com

Chapter – 4

Strategic Decision-Making with Perfect Information: Foundations, Applications and Insights from Game Theory

Lt J. Hajiram Beevi
Assistant Professor, PG and Research Department of Computer Science, Jamal Mohamed College, Trichy, Tamil Nadu, India

S. Munawara Banu
Assistant Professor, PG and Research Department of Computer Science, Jamal Mohamed College, Trichy, Tamil Nadu, India

Abstract

Through its mathematical structure Game Theory analyzes strategic decision-making between rational agents who bring profound influence to multiple domains. The paper explores the fundamental ideas of game theory with an emphasis on how it applies to strategies, decision-making processes, and perfect information games. Through the analysis of Nash equilibrium together with minimax theorem and backward induction and dominance the paper shows how game theory creates models for understanding real-world competition and cooperation and conflict solutions. Besides the paper explores its interdisciplinary applications in economics, business, politics, biology, and computer science, focusing its role in optimizing strategies and predicting outcomes in interdependent decision-making scenarios. Modern challenges require game theory as an indispensable tool because it perfectly models complex interactions while analyzing behavior patterns in markets along with creating strategic plans and enhancing human behavior understanding.

Keywords: Game Theory, Nash Equilibrium, Minimax Theorem, Backward Induction, Decision-Making, Perfect Information Games.

Introduction

"Players" in the formal study of Game Theory may be humans, animals, computer systems, groups, agencies, and much more. To create, organize, evaluate, and understand unique game scenarios, Game Theory offers a mathematical framework. Due to this, it offers beneficial mathematical models and tools for investigating the strategies that agents may use when engaging or collaborating in games. Basically, it examines situations in which the

outcome for each participant is affected by both their own and other player's decisions. It offers useful facts approximately a way to make the exceptional selections, resolve conflicts, negotiate, and compete. It is critical because it can be used to simulate real-world scenarios involving interdependent decisions, offering strategic insights into a variety of human interactions and economic decisions. It has performed an important role in forming fields namely economics (market strategies, auctions), business (pricing strategies, competitive behavior), and military strategy (optimal tactics). Game theory itself has an underlying application in a wide range of contexts, including entertainment, politics, business competition, geopolitical issues between countries, and so on. This applied mathematics has been well-spread into various areas, such as economics, social sciences, biology, politics, international relations, computer sciences, philosophy, and more.

In game theory, games are labeled into various classes based on different criteria. Two major forms of games are: Cooperative vs Non-cooperative Games and Zero-sum vs non-Zero-sum Games. In cooperative games, players must cooperate to accomplish a shared goal, concentrating on how they might build alliances, share information, and make legally binding agreements to increase their overall rewards. Non-cooperative games, on the other hand, handle situations where participants act in the most beneficial way without enforceable agreements or cooperation. Each player in those games must make strategic decisions while keeping in mind the movements of other players' capabilities, which frequently results in competitive outcomes. Any advantage gained by one player comes at the expense of another in a zero-sum game, which is a particular kind of game in which the gains of one person are entirely offset by the losses of other players, resulting in a constant total payoff [1]. Classic examples of zero-sum video games include poker and chess. On the other hand, in non-zero-sum games, the whole payoff is not fixed, offering the opportunity for all players to either benefit or suffer, depending on their decisions. Many real-world scenarios, such as trade negotiations or environmental agreements, fall into this category, in which cooperation can cause mutual profits. By dividing games into various categories, it offers valuable insights into identifying suitable strategies and solutions in a variety of strategic contexts, providing a thorough understanding of the dynamics at work in every particular situation. This chapter focuses on game theory with perfect information, where each player is fully aware of the game's structure, the viable strategies and each player's payoffs [8].

Perfect Information

In game theory, perfect information refers to a situation in which all

players have complete and accurate knowledge of the game's history at every point in time. This means that each player knows the moves previously made by other players, the available strategies, and the payoffs associated with each strategy. There is no hidden information or uncertainty about the game's state, making it easier to predict the consequences of each move.

Examples of games with perfect information include Chess, Go, Tic-Tac-Toe, and Checkers. In Chess, players can see the entire board, know all possible moves, and have access to all necessary information to strategize effectively. Similarly, Go features an open board where players have full visibility of all stones placed and the game's progression. Tic-Tac-Toe ensures that every player can observe the grid and past moves, leaving no room for hidden information. In Checkers, players are fully aware of the position of all pieces and can predict outcomes based on their strategies. These games exemplify the essence of perfect information, where strategic planning relies entirely on observable facts.

In contrast, games with imperfect information introduce an element of uncertainty or hidden information. For instance, in Poker, players cannot see the cards held by their opponents, requiring probabilistic reasoning and strategic bluffing. In Battleship, the positions of the opponent's ships are unknown, demanding educated guesses and careful deduction. Similarly, in Clue, players must uncover hidden information about the murderer, weapon, and location by piecing together limited clues. These differences illustrate the contrasting dynamics between games with perfect and imperfect information [4].

Normal-Form Games

A normal-form game is a framework in game theory used to represent strategic interactions between players. It is defined by three key components: players, strategies, and payoffs. Players are the individuals or entities participating in the game, each making decisions with the goal of maximizing their respective payoffs. Strategies refer to the complete set of actions or decisions available to each player, representing a player's plan of action for every possible scenario in the game. Payoffs are the rewards or outcomes that players receive based on the combination of strategies chosen by all participants, reflecting the players' preferences or utility for specific outcomes. A normal-form game assumes that all players make their choices simultaneously, without knowledge of the decisions of others [6-7].

Representation of Games in Matrix Form:

Normal-form games are often represented using a payoff matrix, which summarizes the strategies and payoffs for all players. For a two-player game,

the matrix rows represent the strategies of one player, while the columns represent the strategies of the other. Each cell in the matrix contains the payoffs for both players based on the strategies chosen. For example, consider a two-player game where Player 1 can choose Strategy A or B, and Player 2 can choose Strategy X or Y. The payoff matrix might look like this:

	X	**Y**
A	(3, 2)	(1, 4)
B	(0, 1)	(2, 3)

In each cell, the first number represents the payoff for Player 1, and the second number represents the payoff for Player 2. For instance, if Player 1 chooses A and Player 2 chooses X, the payoffs are 3 for Player 1 and 2 for Player 2.

Analysis of Strategic Decisions:

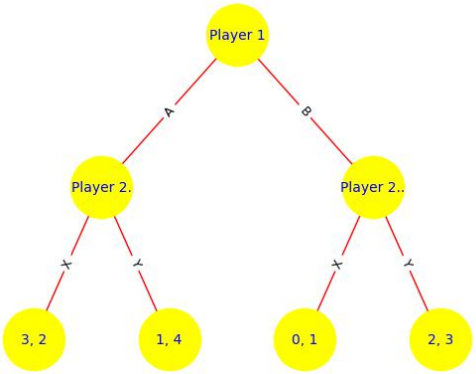
Analyzing strategic decisions in a normal-form game involves identifying the optimal strategies for each player based on their payoffs. Key concepts include:

1. Dominant Strategy: A strategy that yields a higher payoff for a player, regardless of the strategies chosen by others. If a player has a dominant strategy, they will always choose it.
2. Nash Equilibrium: A situation where no player can improve their payoff by unilaterally changing their strategy. In a Nash Equilibrium, every player's strategy is optimal, given the strategies of others.
3. Mixed Strategies: In some games, players may randomize their decisions rather than choosing a single strategy, leading to probabilistic outcomes.

By analyzing the payoff matrix, players can determine the equilibrium strategies and predict the likely outcome of the game. This framework is widely used in economics, business, and decision-making to understand competitive and cooperative interactions.

Extensive-Form Games

Extensive-form games are a way to represent games where players make decisions one after another, instead of simultaneously. These games are often shown using game trees, which look like flowcharts. A game tree starts from an initial point (called the root) and branches out as players make choices. Each branch represents a decision, and the endpoints (leaves) show the outcomes and payoffs. Game trees are especially useful for showing the sequence of moves and how choices at one stage affect future decisions [6-7].



Strategies in Extensive-Form Games:

In extensive-form games, a strategy is a complete plan of action for a player. It describes what the player will do at every possible decision point, even if some of those points are never reached during the actual game. This ensures that the strategy accounts for every possible scenario that could occur.

Backward Induction and Its Applications:

Backward induction is a method used to solve extensive-form games. It starts at the end of the game tree and works backward to find the best decisions at each point. By analyzing the final outcomes and payoffs, players can determine the optimal choices earlier in the game. This method is often used in decision-making scenarios like negotiations, where understanding the likely end result helps guide earlier decisions.

Subgame Perfect Equilibrium:

A subgame perfect equilibrium is a refined solution concept for extensive-form games. It ensures that the strategies chosen are optimal not just for the entire game but also for every smaller part (or "subgame") of the game tree. In simple terms, it guarantees that players make the best decisions at every point, no matter how the game unfolds. This concept helps eliminate unrealistic or non-credible strategies in games with multiple stages.

Extensive-form games are widely used to study scenarios where timing and order of decisions matter, such as negotiations, chess, or investment decisions.

Coalitional Form

The coalitional form (also known as the characteristic function form) in game theory is a framework used to study cooperative games. In this form,

players can form groups, or coalitions, to achieve shared goals and maximize collective payoffs. It focuses on how payoffs are distributed among players when they cooperate and on finding stable and fair allocations.

The coalitional form in game theory introduces several key concepts that focus on how players form groups and share payoffs. A coalition is any subset of players who agree to cooperate, while the grand coalition includes all players working together. The framework is defined by a characteristic function, denoted by v(S), where S represents a coalition. The function v(S) specifies the total payoff that a coalition S can secure for itself, independent of actions taken by players outside the coalition. Additionally, it satisfies $v(\emptyset)=0$, meaning an empty coalition generates no payoff. One of the main goals in coalitional games is to determine how the total payoff v(S) should be divided among the coalition members. Popular solution concepts like the core, Shapley value, and nucleolus ensure fair and stable distributions [6-7].

Examples of coalitional form highlight its applicability in real-world scenarios. In resource sharing, players combine their resources—such as capital, labor, or skills—to generate profits, with the characteristic function defining the achievable profit for any coalition. For instance, companies pooling resources for a joint project exemplify this. In political alliances, parties may form coalitions to pass legislation or win elections, where the characteristic function represents the collective power of each alliance.

The coalitional form plays a crucial role in ensuring fair division of payoffs among coalition members, fostering stability by discouraging subsets of players from breaking away to form separate coalitions. Its wide-ranging applications in fields like economics, political science, and resource management demonstrate its value in analyzing cooperative behavior among entities.

Nash Equilibrium in Perfect Information Games

A Nash Equilibrium is a fundamental concept in game theory where every player in a game chooses a strategy that maximizes their payoff, assuming the strategies of the other players remain unchanged. At this equilibrium, no player can improve their outcome by unilaterally changing their strategy. In essence, Nash Equilibrium reflects a stable state of strategic interaction where no player has an incentive to deviate from their chosen strategy. In perfect information games, all players have complete knowledge of the game's structure and all previous moves. This transparency often simplifies the process of identifying Nash Equilibria compared to games with imperfect information.

	Suspect B: Stay Silent	Suspect B: Confess
Suspect A: Stay Silent	(-1, -1)	(-10, 0)
Suspect A: Confess	(0, -10)	(-5, -5)

The Prisoner's Dilemma is a classic example in game theory that demonstrates the conflict between individual rationality and collective benefit. In this scenario, two suspects are arrested and interrogated separately. Each suspect has two strategies: to confess (betray the other) or to stay silent (cooperate with the other). The outcomes, measured in years of prison, depend on the choices made by both suspects. If Suspect A stays silent, Suspect B can reduce their prison term by confessing, lowering their sentence from -1 to 0. Conversely, if Suspect A confesses, Suspect B minimizes their prison term by also confessing, reducing their sentence from -10 to -5. The same logic applies to Suspect A, creating a situation where both suspects are incentivized to confess [9].

The Nash Equilibrium in this game occurs when both suspects choose to confess, resulting in an outcome of (-5, -5). Neither suspect has an incentive to unilaterally change their strategy, as doing so would increase their own prison term. However, this outcome is suboptimal for both players. The dominant strategy for each player is to confess because it guarantees a better result regardless of the other's choice. Cooperation, where both suspects stay silent, would lead to a better collective payoff of (-1, -1). Despite this, the fear of betrayal and self-interest forces both suspects to choose the Nash Equilibrium.

This example highlights how Nash Equilibrium can lead to stable yet inefficient outcomes, especially in situations where individual incentives are at odds with the group's overall benefit. It illustrates the challenges of achieving cooperation in competitive scenarios.

One of the key limitations of Nash Equilibrium is that it assumes full rationality, whereas in reality, people often have bounded rationality, meaning they make decisions based on emotions, biases, and incomplete information rather than perfectly rational calculations. Additionally, Nash Equilibrium assumes common knowledge of the game structure and player rationality, which may not hold in situations where players have imperfect or asymmetric information. This can make the equilibrium less applicable or useful in real-world scenarios where not all players have access to the same knowledge. Another limitation is that the existence of multiple equilibria can make predicting the outcome difficult, as it is unclear which equilibrium will be selected without further refinements or additional selection criteria. Finally,

Nash Equilibrium does not account for the dynamic nature of real-world games; it fails to consider how players learn, adapt, and change strategies over time in repeated interactions, which is a crucial aspect of decision-making in many strategic settings [2-10].

Dominant Strategies

Dominant strategies play a crucial role in decision-making and strategy analysis in game theory. A dominant strategy is a strategy that yields a better or at least equal payoff for a player, regardless of the strategies chosen by other players. This concept simplifies the analysis of games by focusing on strategies that are inherently optimal for a player. There are three commonly recognized notions of dominance: strong dominance, weak dominance, and very weak dominance. These concepts differ in the conditions under which one strategy is considered better than another. Let's explore each with definitions and examples.

1. Strong Dominance

A strategy strongly dominates another strategy if it provides a strictly better payoff for a player in every possible scenario, regardless of the choices of other players.

Definition: Strategy S1 strongly dominates S2 if Payoff of S1> Payoff of S2 for all possible actions of other players.

Example: Consider a game with two players, A and B, and the following payoff matrix:

	B1	**B2**
A1	5, 3	4, 2
A2	3, 4	2, 1

For Player A, strategy A1 provides a strictly higher payoff compared to A2, regardless of whether Player B chooses B1 or B2. Specifically, the payoffs for A1 (5 vs. 3 and 4 vs. 2) are always greater than those for A2 in all possible scenarios. Therefore, A1 strongly dominates A2.

2. Weak Dominance

A strategy weakly dominates another if it provides a payoff at least as good as the other strategy in all scenarios and strictly better in at least one scenario.

Definition: Strategy S1 weakly dominates S2 if Payoff of S1 ≥ Payoff of S2 for all actions of other players, and strictly better in at least one case.

Example: Consider the following payoff matrix:

	B1	**B2**
A1	3, 3	4, 2
A2	3, 1	2, 3

For Player A, strategy A1 provides a payoff equal to A2 in one scenario (3 vs. 3 when Player B chooses B1). However, A1 provides a strictly better payoff (4 vs. 2) in the other scenario, when Player B chooses B2. Therefore, A1 weakly dominates A2.

3. Very Weak Dominance

A strategy very weakly dominates another if it provides a payoff at least as good as the other strategy in all scenarios. Unlike weak dominance, there is no requirement for a strictly better payoff in any scenario.

Definition: Strategy S1 very weakly dominates S2 if Payoff of S1 $\geq$ Payoff of S2 for all actions of other players.

Example: Consider the following payoff matrix:

	B1	**B2**
A1	3, 3	3, 2
A2	2, 1	1, 3

For Player A, strategy A1 provides a payoff that is always greater than or equal to A2. Specifically, when Player B chooses B1, the payoffs for A1 are 3 $\geq$ 2 and 3 $\geq$ 1. Similarly, when Player B chooses B2, the payoffs for A1 are 3 $\geq$ 1 and 2 $\geq$ 1. However, A1 does not provide a strictly better payoff in any scenario. As a result, A1 very weakly dominates A2.

Minimax Theorem

The Minimax Theorem, introduced by John von Neumann, is a fundamental concept in game theory, particularly in the analysis of two-player zero-sum games. It provides a method to determine the optimal strategies for both players by minimizing potential losses while maximizing potential gains.

In a two-player zero-sum game, the gain of one player is exactly balanced by the loss of the other. The minimax strategy is a method used to ensure that a player minimizes their maximum possible loss, assuming that the opponent plays optimally. Player 1, the maximizer, aims to maximize their minimum guaranteed payoff regardless of Player 2's strategy. Conversely, Player 2, the minimizer, aims to minimize their maximum potential loss, regardless of Player 1's strategy.

The key concept of the minimax strategy involves evaluating the game matrix by considering the worst-case scenario—either the maximum loss or the minimum gain—and selecting a strategy that minimizes potential damage in that scenario. When both players follow their optimal minimax strategies, the outcome is the minimax value of the game.

The mathematical foundation of the minimax strategy is captured by the Minimax Theorem. It states that for every finite two-player zero-sum game, there exists a saddle point (or equilibrium) where the minimax value equals the maximin value. This theorem guarantees the existence of an equilibrium where both players' strategies are balanced, ensuring optimal decision-making under the given conditions [5].

Let us solve a two-player zero-sum game using the minimax strategy. Consider the following payoff matrix, where Player 1's objective is to maximize their payoff (row player), and Player 2's objective is to minimize Player 1's payoff (column player):

Player 2	**Strategy B1**	**Strategy B2**
Player 1 (A1)	2	4
Player 1 (A2)	3	1

Step 1: Identify the Minimax and Maximin Values

Player 1 (Row Player - Maximizer):

Player 1 assumes Player 2 will choose a strategy that minimizes Player 1's payoff.

For each row (strategy), calculate the minimum value of Player 1's payoff:

Row A1: min (3,1) = 1

Row A2: min (2,4) = 2

Player 1 selects the row (strategy) that gives the maximum of these minimum values: max ({1,2}) = 2. Therefore, Player 1's minimax value is 2, and they choose A2.

Player 2 (Column Player - Minimizer):

Player 2 assumes Player 1 will choose a strategy that maximizes Player 1's payoff.

For each column (strategy), calculate the maximum value of Player 1's payoff:

Column B1: max (3,2) = 3

Column B2: max (1,4) = 4

Player 2 selects the column (strategy) that gives the minimum of these maximum values: min ({3,4}) = 3. Therefore, Player 2's maximin value is 3, and they choose B1.

Step 2: Check for a Saddle Point

The saddle point occurs where the minimax value equals the maximin value. In this case:

Player 1's minimax value: 2

Player 2's maximin value: 3

Since $2 \neq 3$, no saddle point exists in pure strategies.

Step 3: Solve Using Mixed Strategies

If there is no saddle point in pure strategies, players can adopt mixed strategies to randomize their choices. To find the optimal probabilities for each player's strategies:

Let Player 1 mix between A1 and A2 with probabilities p and $1-p$, respectively.

Let Player 2 mix between B1 and B2 with probabilities q and $1-q$, respectively.

The expected payoff for Player 1 can be calculated as:

$$E(A, B) = p[q(3) + (1 - q)(1)] + (1 - p)[q(2) + (1 - q)(4)]$$

By solving this equation to equalize expected payoffs across strategies, players can determine the optimal mixed strategy probabilities.

Backward Induction

Backward induction is a systematic method used to solve sequential games, particularly those represented in extensive form. It involves analyzing a game starting from the final possible outcomes and working backward step by step to determine the optimal strategies for all players. This approach assumes that all players act rationally and anticipate the rational decisions of others at every stage of the game [3].

Steps to perform Backward Induction

1. **Start at the End:** Look at the final outcomes (payoffs) at the end of the game tree. Identify the payoffs for each player at these points.
2. **Work Backwards:** Move one step back and analyze the decisions

available to the player at that stage. Assume they will pick the option that gives them the best payoff.

3. **Replace with Best Choices:** Replace the decision at each step with the payoff from the best choice and move further back in the tree.
4. **Repeat to the Start:** Continue this process until you reach the beginning of the game tree. This will show the best sequence of actions for all players.

Let's consider a simple sequential game involving two players: Player 1 and Player 2. The scenario begins with Player 1 deciding whether to Invest or Not Invest. If Player 1 chooses to Invest, Player 2 then decides whether to Support or Reject Player 1's investment. The payoffs for each choice are as follows: If Player 1 invests and Player 2 supports, the payoff is (5, 3), meaning Player 1 gets 5 and Player 2 gets 3. If Player 1 invests and Player 2 rejects, the payoff is (-1, 1). However, if Player 1 decides not to invest, the payoff is (0, 0) for both players.

Backward induction helps solve this game by starting from the end. First, consider Player 2's decision. If Player 1 has chosen to invest, Player 2 can either Support, which gives them a payoff of 3, or reject, which gives them a payoff of 1. Since Player 2 prefers a payoff of 3 over 1, they will choose to Support if Player 1 invests. Next, we move to Player 1's decision. Knowing that Player 2 will choose to Support, Player 1 evaluates their options: investing leads to a payoff of 5, while not investing leads to a payoff of 0. Since 5 is greater than 0, Player 1 will choose to Invest.

Using backward induction, the solution shows that Player 1 chooses to Invest, knowing that Player 2 will choose to Support. The final outcome of the game is (5, 3).

Applications of Game Theory with Perfect Information

Game theory with perfect information applies to situations where all players have complete knowledge of the game state and prior moves. One prominent application is in board games and strategy games such as Chess, Checkers, and Go. These classic examples of perfect information games allow players to see the entire board and all moves, enabling strategic planning. Game theory helps analyze optimal strategies by predicting opponents' best moves. In economic decision-making, it is used for pricing strategies in transparent markets where firms aim to maximize profits and in auctions where bidders make informed decisions based on clear rules, such as open ascending auctions. Negotiation and bargaining benefit from perfect information as it allows parties to make decisions with a full understanding of

each other's preferences and payoffs, leading to fair and stable agreements in business or political contexts.

Another application is in resource allocation, where game theory helps efficiently distribute shared resources, such as land, water, or finances, among entities with complete knowledge of the resources and priorities. In legal and contractual scenarios, such as litigation, where all evidence is known to both parties, game theory guides strategies for settlements or trials. Artificial Intelligence also leverages game theory in applications like chess engines or decision-making bots, optimizing strategies in perfect information environments. Similarly, in supply chain management, companies use game theory to determine production, pricing, and inventory decisions when full information about demand, supply, and costs is available. In education and testing, teachers or test designers can utilize game theory to create fair grading strategies when all students' performance data is accessible. These applications highlight how game theory with perfect information can optimize strategies and outcomes across various fields where transparency and foresight are critical.

Conclusion

Game theory with perfect information offers a robust framework for analyzing and optimizing decision-making in scenarios where all players have complete knowledge of the game state and prior moves. Its applications are vast and diverse, spanning from board games like Chess and Go, where strategic planning thrives on transparency, to complex economic and business decisions, where firms and individuals strive to maximize outcomes based on known variables. The principles of game theory also provide valuable insights in negotiations, resource allocation, legal disputes, artificial intelligence, supply chain management, and education. By leveraging the transparency and foresight inherent in perfect information scenarios, game theory empowers players, organizations, and systems to achieve fair, efficient, and optimal results. This chapter has highlighted the importance of game theory as a powerful tool for understanding strategic interactions, demonstrating its relevance across multiple disciplines and real-world applications.

References

1. Camerer, C. F. (2003). Behavioral Game Theory: Experiments in strategic interaction.
2. Carmona, G. (2012). Existence and stability of Nash equilibrium.
3. Colman, A. M. (2000). Rationality assumptions of game theory and the

backward induction paradox. In Oxford University Press eBooks (pp. 353–371).

4. Garcia, S. C., Tamayo, J. E. I., Carbonell-Olivares, J., & Cabrera, Y. P. (2013). Application of the Game Theory with Perfect Information to an agricultural company. Agricultural Economics (Zemedelska Ekonomika), 59(1), 1–7.
5. Marchi, E. (1967). On the minimax theorem of the theory of game. Annali Di Matematica Pura Ed Applicata (1923 -), 77(1), 207–282.
6. Norozpour, S., & Safaei, M. (2020). An overview on game theory and its application. IOP Conference Series Materials Science and Engineering, 993(1), 012114.
7. Osborne, M. J. (2013). An introduction to game theory. In SAGE Publications, Inc. eBooks (pp. 33–42).
8. Rasmusen, E. (2006). Games and Information: An Introduction to Game Theory.
9. Shubik, M. (1967). Prisoner's Dilemma: A Study in Conflict and Cooperation. By Anatol Rapoport and Albert M. Chammah. American Political Science Review, 61(1), 173–175.
10. Van Damme, E. (1983). Refinements of the Nash equilibrium concept.

Chapter – 5

Generative AI: Exploring its Potential and Challenges

Dr. Monika Gaur
Professor & Principal, Anangpuria School of Management and Technology, Faridabad, Haryana, India

Nidhi Kataria Chawla
Assistant Professor, B.S. Anangpuria Institute of Technology and Management, Haryana, India

Swati Sareen
Computer Educator, Aravali International School, Faridabad, Haryana, India

Chietra Jalota
Computer Educator, Aravali International School, Faridabad, Haryana, India

Abstract:

For a better and prolific life, education plays a vital role. With the advent of technology like artificial intelligence, higher education institutions are integrating technology with traditional teaching methods. For several educational tasks like content generation, adaptation of learning experiences and in-depth educational support, there is an immense potential in Artificial Intelligence (AI). In the current situation, a paradigm shift is started in the study of artificial intelligence (AI) due to innovative abilities of generative models in both supervised and unsupervised learning scenarios. Generative AI has shown advanced performance to solve confounding challenges of real-world such as medical diagnostics, textual imagery fusion, natural language processing, image translation etc. In this chapter, the systematic review and analysis of recent advancements and techniques in Generative AI with a detailed discussion of their applications is discussed in depth. Currently, many industries including consumer, healthcare, finance, retail, and telecom are renovating by using this technology. Apart from these industries, personal lives and society is also altering by using this technology. This article deliberates the working methodology, advantages, limitations, and potential of generative artificial intelligence (AI).

Keywords: Artificial Intelligence, Generative Adversarial Networks, Variational Autoencoders, Transformers, Diffusion Models, Segmentation

Introduction

The architype shifts of AI applications from discriminative to generative is leading to unique use cases and promising opportunities in various domains, including those traditionally resistant to automation. Therefore, researchers and practitioners need to understand the inherent properties of generative AI to effectively leverage its potential while also mitigating associated risks. Generative AI comprises artificial intelligence systems with the capability to generate typescript and metaphors and synthetic data by utilizing generative models. The main task of these models is to comprehend the structures and underlying patterns within the training data, afterward produce new data which has similar characters and features as of training data. The technology is not entirely new. In 1960, generative AI was used by Chatbots. But at the same time, generative AI was not able to generate realistically photos, videos, or voices of real people because of absence of generative adversarial networks, or GANs. GANs are a class of machine learning and projecting framework for approaching generative AI [1]. A primary focus of generative AI's impact is evident in the domains of NLP and Video Translation, where advanced models have emerged with the capacity to tackle a wide array of human-centric challenges. These include tasks like question answering, code generation, language translation, image transformation, and more interdisciplinary applications [2]. In this manuscript, a systematic review of generative AI is done to highlight the substantial applications of Generative Artificial Intelligence models and their performance. Generative AI examples include:

- Text generation Models like GPT (Generative Pre-Trained Transformer)
- Image synthesis Model like DALL-E
- Music Composition Tools

The main aim of this review is:

(a) To comprehend the techniques of generative AI which comprises algorithms and summary of key methodologies.

(b) To do systematic literature review related to the recurring patterns in the development and application of generative AI techniques, emerging trends and common challenges of generative AI.

(c) For the comparative analysis of various approaches of generative AI such as Autoencoders, Generative Adversarial Networks, Transformers, and Diffusion models

(d) To examine successful applications of generative AI such as

knowledge graph generation, natural language processing, video synthesis and generation and image translation.

(e) To recognize ethical challenges and their suggestive solutions for responsible AI development.

2. Working of generative AI

Text, image, video, design, musical notation or any other input can be sent in the form of signal to Generative AI which is understood by AI system. After this, fresh content can be generated by different AI systems. For example, text inputs can be transformed into an image, an image can be altered into a song, or a video can be converted into text. This may include problem-solving techniques, essays and accurate imitations that is made from human voice or images. By using generative AI, a user-friendly interface can be generated which allow users to direct their requests in plain English. The request can also be customized even after getting result by commenting on the content, style, and other aspects.

3. Contemporary Generative Models

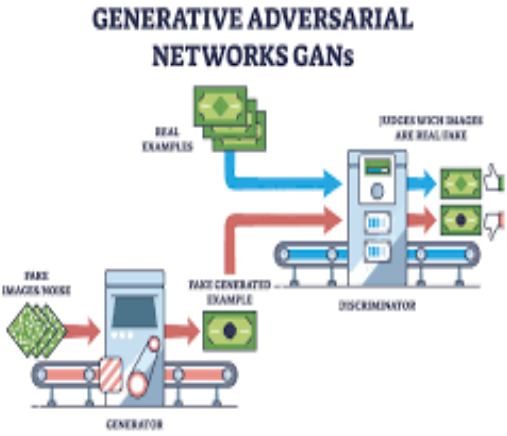
[3]The creation of intelligent machines requires more than the manual tinkering by humans. Generative systems can be created by applying day to day machine learning technology. In the present era, researchers have shifted their emphasis of research from discriminative learning to generative learning. Multiple generative models have developed with the competence of generating new data points like the training data inputs on the basis of their learning and distribution. To epitomize and analyse contents, various AI techniques are combined by generative AI models. For example, to generate text sentences, numerous natural language processing algorithms are used for transformation of parts of speech, entities, and actions from raw characters (such as letters, punctuation, and words). Afterward, several encoding techniques are used to represent such transformed characters as vectors. On the basis of vectors movement, images are transmuted into other visual elements. Bias, racism, fraud, and bloat in the training data can be encoded. After encoding, a specific neural network is used to produce new content for a question or prompt given by the user. GAN and variational autoencoder (VAE), which are neural networks with encoder and decoder, are methods that can create realistic human faces, artificial data for AI training, or even duplicates of specific individuals. Recent developments in encoders such as Google Bidirectional Encoders (BERT), OpenAI GPT, and Google Alpha Fold have also led to neural networks capable of encoding language, images, and proteins, as well as creating original materials.

Contemporary generative AI models have revolutionized content creation across various domains. These models, trained on vast datasets, can generate text, images, audio, and even 3D models with remarkable fidelity. Among the prominent architectures are Generative Adversarial Networks (GANs), which employ a generator and discriminator network to iteratively refine the generated output, and diffusion models, which progressively remove noise from a random input to produce coherent samples. Transformer-based models, such as GPT and its variants, have demonstrated exceptional capabilities in natural language processing, enabling tasks like text generation, translation, and summarization. These advancements have led to a wide range of applications, from creative content generation and data augmentation to drug discovery and material design, showcasing the transformative potential of generative AI.

4. Models of Generative AI

[4] This chapter extensively explores the central themes and topics addressed in previous research concerning GAI using topic modelling techniques covering the period from 1985 to 2023) within the field. Generative AI boasts a diverse range of models, each with unique strengths and applications. Here are some of the main ones:

- **Generative Adversarial Networks (GANs)**: [5] A generative adversarial network (GAN) is a deep learning architecture. It trains two neural networks to compete against each other to generate more authentic new data from a given training dataset. GANs consist of two neural networks, a generator and a discriminator, locked in a competitive dance. The generator creates synthetic data, while the discriminator tries to distinguish between real and fake data. This adversarial process leads to the generation of highly realistic content, particularly in image synthesis and manipulation. [5] This chapter proposed a framework that is the combination of two distinct generative models, an encoder–decoder network and a generative adversarial network. Because of the use of the information noise in the GAN, there were very small chances of mode collapse, and it generated realistic-looking brain tumor images Results proved that the proposed framework could be used as a clinical tool for neurologists and various other medical experts, as the proposed method can be used to generate medical images in other domains, not only for brain tumors.



Source: https://pg-p.ctme.caltech.edu/blog/ai-ml/what-is-generative-adversarial-network-types [6]

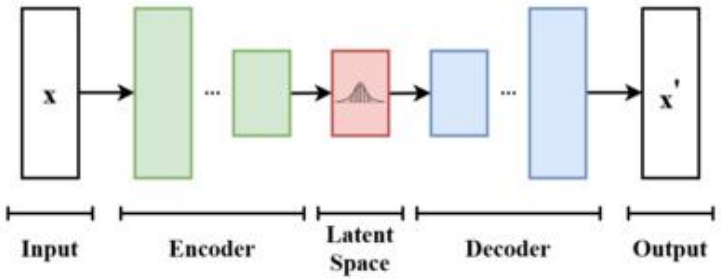
Advantages

1. **Realistic Data Generation:** GANs excel at generating synthetic data that is remarkably similar to real data.
2. **Unsupervised Learning:** GANs can learn from unlabelled data, which is a significant advantage in situations where labelled data is scarce or expensive to obtain.
3. **Novelty:** GANs can generate novel data samples that are not present in the training data.

Limitations

1. **Mode Collapse:** In some cases, the generator may get stuck producing a limited variety of outputs
2. **Computational Cost:** Training GANs is computationally expensive, requiring significant resources and time.
3. **Training Instability:** GANs are notoriously difficult to train.

Variational Autoencoders (VAEs): Variational Autoencoders learns how to describe the nature of data. VAEs learn to encode data into a lower-dimensional latent space, capturing the essential features. By sampling from this latent space and decoding, VAEs can generate new data instances. They are known for their ability to produce diverse and novel outputs, often used in image generation and data compression [7].



Source: https://en.wikipedia.org/wiki/Variational_autoencoder [8]

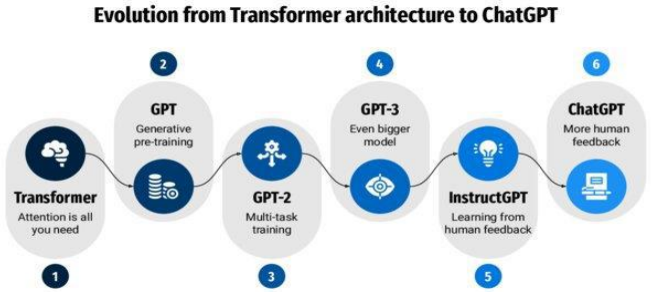
Advantages

1. **Generative Capabilities:** VAEs can generate new data samples by sampling from the latent space and passing them through the decoder.
2. **Learning Latent Representations:** VAEs learn a structured and meaningful latent space, which can be useful for tasks like dimensionality reduction, data visualization, and anomaly detection.
3. **Probabilistic Framework:** The probabilistic nature of VAEs allows for handling uncertainty and noise in the data.

Limitations

1. **Blurry Samples:** VAEs can sometimes generate blurry or less sharp samples compared to other generative models like GANs.
2. **Training Complexity:** Training VAEs can require careful tuning of hyperparameters and network architectures.

Autoregressive Models: This model is a class of statistical models which are used to predict future values of a time series based on the past values. These models generate data sequentially, predicting each element on the basis of same distribution as the training data. They excel at capturing long-range dependencies in data, making them suitable for tasks like text generation and music composition. Transformer-based models, like GPT, are a prominent example of autoregressive models [9].



Source:
https://www.researchgate.net/publication/368688549_ChatGPT_Jack_of_all_trades_master_of_none/figures?lo=1 [10].

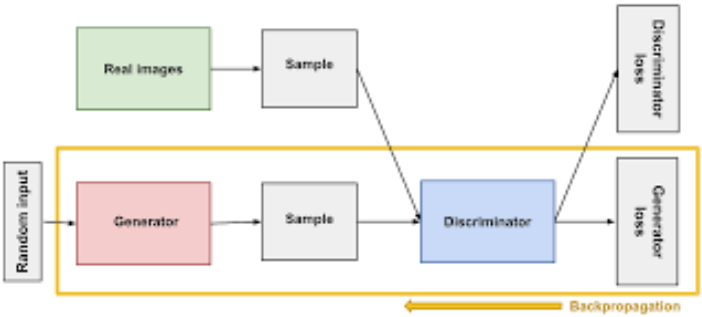
Advantages

1. **Captures Temporal Dependencies:** Autoregressive models excel at capturing the relationships between consecutive data points in a sequence. This makes them particularly well-suited for time series data, where the order of observations matters.
2. **Simple and Interpretable:** Compared to some other complex models, autoregressive models are relatively straightforward to understand and implement. The model's coefficients often have clear interpretations, providing insights into the relationships between past and future values.

Limitations

1. **Limited to Short-Term Dependencies:** While autoregressive models can capture temporal dependencies, they are often better suited for modelling short-term relationships.
2. **Sensitivity to Noise:** Autoregressive models can be sensitive to noise in the data. Outliers or irregularities in the time series can negatively impact the model's performance.

Diffusion Models: The diffusion model is a generative model that creates new data, such as images or sounds, by mimicking the data on which it was trained. Inspired by the physics of diffusion, these models gradually add noise to data until it becomes pure noise. Then, they learn to reverse this process, denoising to generate new samples. Diffusion models have shown remarkable success in high-quality image generation, surpassing GANs in some aspects [11].



Source: https://medium.com/@kemalpiro/step-by-step-visual-introduction-to-diffusion-models-235942d2f15c [12].

Advantages

1. **High-Quality Outputs:** Diffusion models excel at generating high-fidelity data, particularly images.
2. **Stable Training:** Unlike GANs, which can be notoriously difficult to train, diffusion models have a more stable training process.
3. **Flexibility:** Diffusion models can be adapted to various data types and tasks, including image generation, text-to-image synthesis, image editing, and even audio generation.

Limitations

1. **Computational Cost:** Diffusion models can be computationally expensive to train and sample from.
2. **Slow Generation:** Compared to other generative models like GANs, diffusion models can be slow at generating new samples.
3. **Memory Intensive:** Diffusion models often require significant memory to store the intermediate states of the diffusion process.

Benefits of Generative AI

1. **Enhanced Creativity and Innovation:**

- **Content Creation:** Generative AI can assist in creating various types of content, including text, images, music, and even videos. This can boost creativity for artists, writers, and content creators by providing new ideas and inspiration.
- **Design and Development:** In fields like architecture, product design, and software development, generative AI can help explore new design possibilities and generate innovative solutions.

2. **Increased Efficiency and Productivity:**
    - **Automation:** Generative AI can automate repetitive tasks, such as writing emails, summarizing documents, or generating code. This frees up human time for more complex and strategic work.
    - **Accelerated R&D:** In fields like drug discovery and materials science, generative AI can accelerate research and development by generating and testing numerous potential solutions much faster than traditional methods.
3. **Advancements in Science and Research:**
    - **Data Analysis:** Generative AI can analyze vast amounts of data and identify patterns that would be difficult for humans to detect. This can lead to breakthroughs in various scientific fields.
    - **Simulation and Modeling:** Generative AI can create realistic simulations and models, helping researchers understand complex phenomena and make predictions.

Limitations of Generative AI

While generative AI offers a wealth of potential, it's important to be aware of its limitations:

1. **Lack of True Understanding:**
    - **Surface Level:** Generative AI models are excellent at recognizing patterns and relationships in data, but they don't truly understand the meaning behind the data.
    - **Contextual Errors:** They can struggle with nuances in language, such as sarcasm, humour, or complex reasoning, leading to outputs that are contextually inappropriate or nonsensical.
2. **Dependence on Data:**
    - **Bias Amplification:** Generative AI models are trained on vast amounts of data, which can contain biases present in society.
    - **Data Quality:** The quality of the generated output is heavily dependent on the quality of the training data. If the data is incomplete, inaccurate, or biased, the model's performance will suffer.
3. **Technical Challenges:**
    - **Computational Cost:** Training and running large generative AI models can be computationally expensive, requiring significant resources and energy.

- **Evaluation Difficulty:** Evaluating the quality and creativity of generated outputs is challenging, as there are no universally agreed-upon metrics.
- **Lack of Control:** Sometimes, it can be difficult to control the specific features or content of the generated output.

Future of Generative AI

The future of generative AI is brimming with potential, promising to revolutionize how we create, interact with technology, and solve complex problems. Following are some exciting prospects of this technology:

1. **Interactive AI:** Future AI systems may be able to generate content dynamically during live conversations, create personalized visuals on the spot, and respond to changing situations in real-time, making interactions more engaging and personalized.
2. **Personalized Content:** This trend will enable more personalized experiences, where AI can adapt to individual preferences and generate content tailored to specific needs and interests.
3. **Responsible AI:** As generative AI becomes more powerful, there will be a growing focus on addressing ethical concerns and mitigating biases in generated outputs.
4. **Synergistic Applications:** Generative AI will be increasingly integrated with other technologies, such as cloud computing, IoT, and robotics, leading to synergistic applications that can solve complex problems and create new possibilities.

Conclusion

The fast change of generative AI has changed many areas causing both benefits and concerns for workers, scholars and decision makers. The advanced system of generative AI is proficient to generate text, image and music, we need to carefully look at their effects. The future of generative AI is full of promise, with the potential to transform various aspects of our lives. By addressing the current limitations and focusing on ethical development, we can unlock the full potential of this technology and create a future where AI empowers us to achieve more than ever before [13].

References

1. Leonardo B., Gero S. “Generative Artificial Intelligence”, Electronic Markets, 2023, https://doi.org/10.1007/s12525-023-00680-1
2. Sandeep S.S, Affan B.H, Sanjay K, Fiona C, Generative Artificial

Intelligence: A systematic review and applications, *Multimedia Tools Appl*, 2024. https://doi.org/10.1007/s11042-024-20016-1

3. Zant T.V.D., Kouw M, Schomaker L, Philosophy and Theory of Artificial Intelligence, SAPERE 5, Springer Verlag, Berlin, 2012, pp. 107-120
4. Priyanka G., Bosheng D., Chang G., Ding D. "Generative AI: A systematic review using topic modelling techniques", Data and Information Management, June 2024, 8(2), 100066
5. https://aws.amazon.com/what-is/gan/.
6. https://pg-p.ctme.caltech.edu/blog/ai-ml/what-is-generative-adversarial-network-types
7. Desta H. Rick B. "Brain Tumor Classification Using a Combination of Variational Autoencoders and Generative Adversarial Networks", MDPI, 2022, 10(2), https://doi.org/10.3390/biomedicines10020223
8. https://en.wikipedia.org/wiki/Variational_autoencoder
9. https://botpenguin.com/glossary/autoregressive-models
10. https://www.researchgate.net/publication/368688549_ChatGPT_Jack_of_all_ trades_master_of_none/figures?lo=1
11. https://www.altexsoft.com/blog/generative-ai/. 5 Sep. 2024.
12. https://medium.com/@kemalpiro/step-by-step-visual-introduction-to-diffusion-models-235942d2f15c
13. Mudasir A, "Generative AI: Challenges and the Road Ahead", International Journal of Science and Research, Oct. 2024, 13(10), 716-725