**“STUDENT PERFORMANCE ANAYZING USING K NEAREST NEIGHBOR ALGORITHM”**



Department of Computer Science and Engineering(CSE)

**Bangladesh University of Business and Technology (BUBT)**

**DECEMBER , 2019**



*i*

**BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)**



**STUDENT PERFORMANCE ANAYZING USING**

**K NEAREST NEIGHBOR ALGORITHM**

*A Thesis*

*Submitted to the Department of Computer Science and*

*Engineering*

*in partial fulfillment of the requirements*

*for the degree of*

**Bachelor of Science in Computer Science and Engineering**

**By**

|  |  |  |
| --- | --- | --- |
| **Md Ashraful Islam** | **SK Tanvir Rahman** | **Shafiur Rahman Chowdhury** |
| 32th intake | 32th intake | 32th intake |
| ID: 15162103018 | ID: 15162103041 | ID: 15162103043 |
| Section: 01 | Section: 01 | Section: 01 |

**Supervised by**

**Md. Saifur Rahman**

Assistant Professor, Department of CSE, BUBT

Rupnagar, Mirpur-2, Dhaka-1216, Bangladesh





*ii*

**ABSTRACT**

Time series forecasting is the use of a model to predict future values based on previously observed values. Time series are widely used for non-stationary data, like economic, weather, stock price, student’s academic result and retail sales etc. The achievement of this academic qualification enables him to build his future career and determine the course of his life through joining a bachelor program in a university based on the marks obtained in this exam. On another hand, the universities determine the admission rates for each discipline and program depending on the general high school exam results in each year. A smart prediction model of exam results in next year will help the universities to determine the level from the beginning of the acceptance process……………



*iii*

**DECLARATION**

We hereby declare that the thesis entitled “**STUDENT PERFORMANCE ANAYZING USING K NEAREST NEIGHBOR ALGORITHM**” submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering in the Faculty of Computer Science and Engineering of Bangladesh University of Business and Technology, is our own work and that it contains no material which has been accepted for the award to the candidate(s) of any other degree or diploma, except where due reference is made in the text of the thesis. To the best of our knowledge, it contains no materials previously published or written by any other person except where due reference is made in the thesis.

*Signature* *Signature* *Signature*

|  |  |  |
| --- | --- | --- |
| **-----------------------------** | **----------------------------** | **--------------------------------** |
| **Md Ashraful Islam** | **SK Tanvir Rahman** | **Shafiur Rahman Chowdhury** |
| ID: 15162103018 | ID: 15162103041 | ID: 15162103043 |



*iv*

**CERTIFICATE**

This is to certify that Md Ashraful Islam, SK. Tanvir Rahman and Shafiur Rahman Chowdhury of B.Sc. in CSE has completed their thesis work titled **“STUDENT PERFORMANCE ANAYZING USING K NEAREST NEIGHBOR ALGORITHM”** in partial fulfillment for the requirement of B.Sc. in Computer Science andEngineering from Bangladesh University of Business and Technology in the year 2020

|  |  |  |
| --- | --- | --- |
| *--------------------------* | *------------------------------* | *--------------------------* |
| **Md Ashraful Islam** | **SK. Tanvir Rahman** | **Shafiur Rahman Chowdhury** |
| ID: 15162103018 | ID: 15162103041 | ID: 15162103043 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Thesis Supervisor

**(Md. Saifur Rahman)**

Assistant Professor

Department of Computer Science and Engineering (CSE)

Bangladesh University of Business and Technology (BUBT)



*v*

**ACKNOWLEDGEMENTS**

We would like to pay our gratitude to the **Almighty Allah** who created us with all the abilities to understand analysis and develop the process with patience.

We are thankful to our thesis supervisor **Md. Saifur Rahman,** Assistant Professor, Computer Science & Engineering Department, Bangladesh University of Business & Technology for his professional guidance and motivation during the work of this thesis which is a major part of it. Without his valuable support and guidance, this thesis could not reach this level of development from our point of view.

We would like to thank all the Faculty members, Department of CSE, Bangladesh University of Business & Technology for their valuable time spend in requirements analysis and evaluation of the thesis work.

We would like to express our sincere and warm gratitude to all those who have encouraged us directly, provided mental encouragement and criticized our work in several phases during the development of this thesis and for preparing this thesis indirectly.

**Md Ashraful Islam (**15162103018**)**

**SK. Tanvir Rahman (**15162103041**)**

**Shafiur Rahman Chowdhury (**15162103043)



*vi*

**DEDICATION**

Dedicated to our parents, teachers, friends, relatives and all who loved us

for all their love and inspirations.



*vii*

**APPROVAL**

This thesis is **“STUDENT PERFORMANCE ANAYZING USING K NEAREST NEIGHBOR ALGORITHM”** This report is submitted by **Md Ashraful Islam** (15162103018); **SK. Tanvir Rahman** (15162103041); **Shafiur Rahman Chowdhury** (15162103043), Department of Computer Science and Engineering, Bangladesh University of Business and Technology under the supervision of **Md. Saifur Rahman**, Assistant Professor, Department of Computer Science and Engineering has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc. Engg.) in Computer Science and Engineering.

**Supervisor:**

**Md. Saifur Rahman**

Assistant Professor

Department of CSE, BUBT



*viii*

**Abbreviations**

KNN

CNN

DBNs

LSTM

LIFO

RMS

PDF

K-Nerest Neighbor

Convolutional Neural Network

Deep Belief Networks

Long Short-Term Memory

Last In First Out

Root Mean Square

Probability Density Function



*ix*

**Table of Contents**

|  |  |
| --- | --- |
| **ABSTRACT** | **ii** |
| **ACKNOWLEDGEMENT** | **iii** |
| **DECLARATION** | **iv** |
| **DEDICATION** | **vi** |
| **CERTIFICATE** | **vii** |
| **APPROVAL** | **viii** |
| **ABBRIVIATIONS** | **ix** |

|  |  |  |
| --- | --- | --- |
| **Chapter 01: Introduction** | | **1** |
| 1.1 | Introduction | 1 |
| 1.2 | Existing systems | 3 |
| 1.3 | Motivation | 5 |
| 1.4 | Aims and Objectives of the thesis | 7 |
| 1.5 | Contribution | 9 |
| 1.6 | Organization of thesis | 10 |
| 1.7 | Conclusions | 11 |

|  |  |  |
| --- | --- | --- |
| **Chapter 2: Existing Systems** | | **13** |
| 2.1 | Is classification technique appropriate? | 13 |
| 2.2 | Every ML algorithm: three keys | 12 |
|  | 2.2.1 Representation | 14 |
|  | 2.2.2 Evaluation | 18 |
|  |  | 19 |
|  |  | 22 |
|  |  | 23 |



*x*

|  |  |  |  |
| --- | --- | --- | --- |
| 2.3 | Learning: three types | | 25 |
|  | 2.3.1 | Supervised learning | 25 |
|  | 2.3.2 | Unsupervised learning | 29 |
|  | 2.3.3 | Reinforcement learning | 32 |
| 2.4 The learning problem: an overview | | | 36 |
| 2.5 | Nearest Neighbor | |  |
| 2.6 | Importance of the factor K | |  |
| 2.7 | Conclusions | | 32 |

|  |  |  |
| --- | --- | --- |
| **Chapter 3: Proposed Model** | | **38** |
| 3.1 | Introduction | 38 |
| 3.2 | Procedure | 39 |
| 3.3 | Algorithm | 44 |
| 3.4 | Flowchart | 64 |
| 3.5 Advantages | | 65 |
| 3.6 | Conclusions | 65 |

|  |  |  |
| --- | --- | --- |
| **Chapter 4: Experimental Results and analysis** | |  |
|  | | **56** |
| 4.1 | Introduction | 56 |
| 4.2 | Result Analysis | 57 |
| 4.3 | Applications | 73 |
| 4.4 | Conclusions | 74 |



*xi*

|  |  |
| --- | --- |
| **List of Figures** | **xiv** |
| **List of Tables** | **xvi** |
| **References** | **78** |

|  |  |  |
| --- | --- | --- |
| **Chapter 5: Conclusions** | | **75** |
| 5.1 | Conclusions | 75 |
| 5.2 | Limitation | 76 |
| 5.3 | Future Works | 77 |



*xii*

**List of Figures**

|  |  |
| --- | --- |
| Figure 2.1: Traditional programming vs machine |  |
| learning | 15 |
| Figure 2.4: The Learning Problem | 26 |
| Figure 3.6: Flowchart of ID3 Algorithm | 30 |
| Figure 4.2: First Decision Tree | 35 |
| Figure 4.3: Final Decision Tree | 36 |



*xiii*

**List of Tables**

|  |  |
| --- | --- |
| Table 2.5: An unknown Boolean function with 7 examples, |  |
| each comprising of four Boolean inputs | 27 |
| Table 3.1: Observations of having diabetes | 29 |



*xiv*

**Chapter- 1**

**Introduction**

**1.1 Introduction**

In order to motivate the concept of machine learning (ML), consider three seemingly distinct situations: 1. While uploading a new picture onto the Face book app on our phone, the app automatically ”suggests” tagging people that are in our friend list, with their respective names hovering over their faces. 2. While browsing an e-commerce website such as Amazon or Netflix for products, the system in turn”recommends” new, unseen products based on a persons’ preferences. 3. Self-driving cars being able to drive autonomously along highways, and is able to adapt according to an unpredictable environment. What do all these three have in common? They all incorporate ML algorithms that learn from data. The notion of a computer learning, i.e. the ability to program itself, has rapidly become attractive that in addition to the above-mentioned examples, applications such as financial forecasting, medical diagnosis, search engines, robotics, gaming, credit evaluation etc. have now become tremendously dependent on ML. In Chapter 1, we establish a general overview of the foundations of ML; by stating the learning problem, the three key components to every ML algorithm, and the three types of learning. As there are many different types of ML, in this report we will focus on the most popular and extensively studied: classification [3, 2]. In Chapter 2, we introduce one of the ways to represent a ML program, i.e. a classifier known as Decision Trees. We dive deeper into Decision Tree Learning by developing and analyzing the ID3 algorithm, in particular key concepts such as Shannon’s Entropy and Information Gain, in Chapter 3. As part of this, we have also survey the ID3 algorithm, and obtained a correctly working Decision Tree model that is able to predict if a person has diabetes or not, using a data set [5]. In Chapter 4, we discuss two methods: Gain ratio. Finally, we conclude this report by discussing future goals that we would like to pursue with regard to Decision Tree Learning and ML, in general.



*15*

**1.2 Existing Systems**

We have studied and worked with “**Survey on ID3 Algorithm in Data Mining** “but we also studied the algorithm of Edge Detection. Like;KDD.;SVM.;KNN. ;ANN. **Knowledge Discovery in Databases** (**KDD**) is an automatic, exploratory analysis andmodeling of large data repositories. **KDD** is the organized process of identifying valid, novel, useful, and understandable patterns from large and complex data sets.

A **Support Vector Machine** (**SVM**) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the **algorithm** outputs an optimal hyperplane which categorizes new examples. It can solve linear and non-linear problems and work well for many practical problems.

**K-nearest neighbors** (**KNN**) algorithm is a type of supervised ML algorithm whichcan be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well −

* **Lazy learning algorithm** − KNN is a lazy learning algorithm because it doesnot have a specialized training phase and uses all the data for training while classification.
* **Non-parametric learning algorithm** − KNN is also anon-parametriclearning algorithm because it doesn’t assume anything about the underlying data.

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.



*16*

**1.3 Motivation**

The major challenges in Classification is the selection of proper attributes and parameters based on the datasets that would be classified, and also after a huge number of calculations and selection of parameters the accuracy in classification is not achieved, hence we are proceeding with the supervised learning techniques like support vector machine, decision tree algorithm and clustering techniques, so that we could provide a novel system that can classify the data more properly and accurately.

**1.4 Aims and Objectives of the thesis**

The main objective of the thesis is to survey ID3 algorithm in data mining.

In [3], they extend their previous algorithm by changing the attribute selection to allow the neighboring second-place important attributes to be combined with the most important attributes. The proposed algorithm is tested on four standard benchmarks from the UCI repository. The experimental results indicate the significant reduction in the maximum depth and the classification depth of the decision tree. In addition, the testing time is also reduced while the classification accuracy is satisfying stable.

ID3 (Iterative Dichotomiser 3) algorithm which was presented by Quinlan is a famous decision tree algorithms, but ID3 has some shortcomings such as high complex computation in computing the information entropy expression, multivalue bios problem in the process of selecting an optimal attribute, large scales, etc. In order to solve the above problems, the improved ID3 algorithm is proposed, which combines the simplified information entropy with coordination degree in rough set theory. The experiment result has proved the feasibility of the optimized way [4].

We show that there are two ways to creating decision tree entropy and information gain.

The equation of information entropy is-

*Information Entropy:*

( ) = − ∑| | =1 log2

The equation of information gain is:



*17*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Information Gain:* |  |  |  |  |
| ( , ) = ( ) − | ∑ℎ=1 | | | | ( ) |  |
| | | |  |
|  |  |  |  |

In this paper we try to overview all the ID3 algorithm related papers. And also show which methods/rules are optimal for creating a decision tree.

**1.5 Contribution**

We tried to implement and experiment with existing ideas in our thesis work. In our system we proposed a way to gain more accuracy than previous works which can be said as the most important proposal of our work. For this we proposed a survey on ID3 algorithm.

It should be noted that it is not yet formally proven the correctness or falsehood of our proposed model but as we came to gain certain good outputs and by our calculation we can say that this proposal is good enough for the next level of accuracy. Instead, this thesis is limited to contributing, hopefully strong, evidence for or against its validity. Hence, this work is advisable as our consideration.

Here are our contributions of this thesis:

1. This thesis helps for generating new idea for higher accuracy of generating decision tree than the previous work and to make it more efficient with this work.
2. Our thesis will help to understand classification and recognition data in a easy way and we tried to implement the system in a simple manner so that anyone can use it for different purpose.



*18*

1. Supervised classification was used for getting different patterns on different datasets for better results and accuracy and for analyzing the system performance in different situations.
2. Generating new algorithm and also testing it for the best possible outcomes.
3. Working with Python and also Implementing C++ codes and analyze .

**1.7 Conclusions**

In this research paper, we represented the ID3 algorithm. ID3 algorithm is efficient

in classifying huge datasets and the simplest algorithm. This paper provides some basic fundamental ideas about the ID3 algorithm like pros and cons, procedures, applications. For mining a data set, It is the most efficient algorithm.

For future works, how to quickly and accurately adapt to more new samples in online classification systems should be researched, and choosing a more efficient assessment method that can reasonably assign the training set and the testing set is necessary. We have majorly focused on decision tree and support vector machine.ID3 algorithm is the easiest algorithm and capable of classifying huge datasets. A neural network is the competitor for support vector machine, but it fails in some cases with respect to nonlinear classification. We have seen that SVM is best suited for it. Selection of proper hyperplane and proper parameters for RBF kernel gives more accurate results as compared to neural networks.



*19*

**Chapter- 2**

**Machine Learning in a nutshell**

In this chapter we provide an overview to ML. First, we discuss how ML is different to the traditional form of every ML algorithm boils down into three components. We state the three major types of learning algorithms.

**2.1 Is Machine Learning magic?**

What makes computers so powerful is the ability to automate tasks that were done by people. This resulted to computers becoming drastically cheaper, and we are able to accomplish tasks that were once thought of as impossible. But while programming has given us these capabilities, we continue to face several challenges such as debugging, implementation, scaling, to name a few. Yet consider at a higher level, what if we were able to tell a computer what to do and it will learn and be able to program by itself by just looking at the data? Now, that seems quite a proposition. But, is this possible? Pedro Domingos [3], one of the leading researchers in ML, illustrates the fundamental differences between traditional programming and machine learning. Consider the figure below:

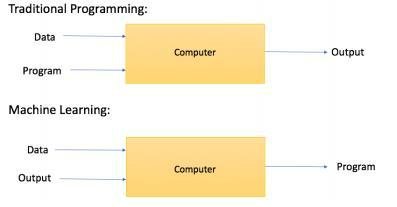


Figure 1.1: Traditional programming vs machine learning

Within the traditional context as we all know, as input to the computer we provide the data and the program. The computer provides the programmer with an output after



*20*

computation. But, within the ML paradigm, the data and the output are now inputs to the computer. And, the output of a ML algorithm is another algorithm. Think of ML as being the”inverse” of traditional programming. So to answer the question: no, ML is not magic. Domingos [3] provides a compelling analogy to farming! Consider the following:

* 1. Seeds = learning algorithms. A seed is simple in nature, yet a whole tree can grow out of it. In ML, learning algorithms are quite simple, but yields very powerful, complex results.

1. Fertilizer + water = Data. Just as fertilizers and water provide crops with nutrients and help them to grow. In ML, the data is what enables learning algorithms to grow.
   1. Farmer = Programmer

4. Crops = Programs (ML output)

**2.2 Every ML algorithm: three keys**

Given its success in practice and relevance in so many different applications, it should not be surprising that there are thousands of ML algorithms. However, the fundamental ingredients to every ML algorithm remains constant, it is the manner by which they are combined together based on the particular problem at hand that results in there being so many of them [2, 3]. The key components are: representation, evaluation and optimization.

**2.2.1 Representation**

If a learning algorithm outputs a program, first thing we must choose is what the language is that the program will be written in. For Human programmers, the choice could be: C, Python, and Java etc. For ML, the choice could be:



*21*

1. Decision Trees (nested if/else statements) - simple.
2. Instances – simplest ML program: “remember what you saw” (memory-based learning)
3. Graphical models (Bayes/Markov networks) – inside the brain of self-driving cars; Google ads uses Bayesian networks for user prediction (if advert will be clicked on or not).
4. Support Vector Machines – related to instances; kernel = measures the similarity between data points.
5. Model ensembles = take a bunch of the above-mentioned ML programs or variations of it and combine them. Eg: Netflix prize winner, Kaggle competitions.

These are only a few examples of the many different ML representations that are available. To go through them all would be beyond the scope of this report.

**2.2.2 Evaluation**

Say we choose Decision Trees as the ML program. Need to ask, “What is the best one that will model the phenomenon that I’m interested in?” “What is the best decision tree to decide if a person is a good credit risk or not?” To do so, we need to find a way to “score” our programs and in the case for Decision Trees, typically we would be after accuracy as a good measure.

**2.2.3 Optimization**

Once we have chosen a representation model and an evaluation measure, now there is the search process by which we optimize that measure: how do we find the most



*22*

optimal decision tree? Naively, we could try all possible decision trees – brute force.

But, there are more decision trees than atoms in the universe so that’s not going to

work. Thus, we do optimization (or in AI, “search”). There are three types:

combinatorial, convex and constrained. For the purposes of this report, we’ll be

focusing on combinatorial optimization. For example, doing a greedy search - try a

bunch of things, pick the best one. Keep going, until the single biggest gain is found,

irrespective of what might happen in the long run. This is the approach used in

discrete models like decision trees (which are essentially graphs with discrete

sections).

**2.3** **Learning: three types**

Next, we’ll briefly give an overview of the three major types of learning: supervised, unsupervised and reinforcement learning.

**2.3.1 Supervised learning**

Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

Y = f(X)

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

It is called supervised learning because the process of algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers; the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

The basic principle is the training data includes desired outputs. Think of this as a

”learning with a teacher” - somebody has already labeled what the “right” answer 9 is.

Eg: Somebody has already labeled which emails are spam and which aren’t;

Somebody labeled which x-rays show cancer and which don’t. Thus, we know what

to learn. Supervised or inductive learning is the most mature (widely studied and used

in practice) kind of learning, and forms the basis of decision trees learning.



*23*

Supervised learning problems can be further grouped into regression and classification problems.

* **Classification**: A classification problem is when the output variable is acategory,, such as “red” or “blue” or “disease” and “no disease”.
* **Regression**: A regression problem is when the output variable is a real value,such a as “dollars” or “weight”.

Some common types of problems built on top of classification and regression include recommendation and time series prediction respectively. Some popular examples of supervised machine learning algorithms are:

* Linear regression for regression problems.
* Random forest for classification and regression problems.
* Support vector machines for classification problems.

**2.3.2 Unsupervised Learning**

Unsupervised machine learning cannot be directly applied to a regression because it is unknown what the output values could be, therefore making it impossible to train the algorithm how you normally would.

Unsupervised learning is where you only have input data (X) and no corresponding output variables.

The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data.

In sharp contrast to inductive learning, the training data does not include desired

outputs within an unsupervised learning environment. This is a much harder but the

most important kind of learning, in the long run. As a useful analogy, think of how

babies learn how to walk on its own - this would be an instance of unsupervised



*24*

learning. If a parent tells the baby, “that’s a chair” or “that’s a table” – that’s supervised learning.

Unsupervised learning problems can be further grouped into clustering and association problems.

**Clustering**: A clustering problem is where you want to discover the inherentgroupings in the data, such as grouping customers by purchasing behavior.

**Association**: An association rule learning problem is where you want to discoverrules that describe large portions of your data, such as people that buy X also tend to buy Y.

Some popular examples of unsupervised learning algorithms are:

* ***[K-Means Clustering](https://algorithmia.com/algorithms/pappacena/kmeans) –*** clustering your data points into a number (K) ofmutually exclusive clusters. A lot of the complexity surrounds how to pick the right number for K.
* ***[Hierarchical Clustering](https://algorithmia.com/algorithms/weka/WekaHierarchicalClusterer)* –**clustering your data points into parent and childclusters. You might split your customers between younger and older ages, and then split each of those groups into their own individual clusters as well.
* ***[Probabilistic Clustering](https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/)*** –clustering your data points into clusters on aprobabilistic scale.

**2.3.3 Reinforcement learning**

Reinforcement learning is the training of machine learning models to [make a sequence](https://blog.openai.com/openai-gym-beta/)

[of decisions.](https://blog.openai.com/openai-gym-beta/) The agent learns to achieve a goal in an uncertain, potentially complex

environment. In reinforcement learning, an artificial intelligence faces a game-like

situation. The computer employs trial and error to come up with a solution

to the problem. To get the machine to do what the programmer wants, the artificial

intelligence gets either rewards or penalties for the actions it performs. Although

the designer sets the reward policy–that is, the rules of the game–he gives the model



*25*

no hints or suggestions for how to solve the game. It’s up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| and finishing | | with | | sophisticated | | tactics and superhuman | | | | | skills. By leveraging | | | |
| the power | of search | | | and many trials, | | | reinforcement learning | | | | is | currently | | the most |
| effective | way to hint machine’s | | | | | creativity. In contrast | | | | to human | | | beings, | artificial |
| intelligence | | can | gather | | experience | | from | | thousands | of parallel | | | game | plays if |
| a reinforcement | | | learning | | algorithm | | is | run | on a sufficiently | | | powerful | | computer |
| infrastructure. | | |  |  |  |  |  |  |  |  |  |  |  |  |
| Applications | | of reinforcement learning | | | | | | were | in the past | limited | | by weak | | computer |

infrastructure. However, as Gerard Tesauro’s backgamon AI superplayer developed in 1990’s shows, progress did happen. That early progress is now rapidly changing with powerful new computational technologies opening the way to completely new

inspiring applications. Training the models that control autonomous cars is an excellent example

of a potential application of reinforcement learning. In an ideal situation, the computer should get no instructions on driving the car. The programmer would avoid hard-wiring anything connected with the task and allow the machine to learn from its own errors. In a perfect situation, the only hard-wired element would be the reward function.

**For example**: in usual circumstances we would require an autonomous vehicle to putsafety first, minimize ride time, reduce pollution, offer passengers comfort and obey the rules of law. With an autonomous race car, on the other hand, we would emphasize speed much more than the driver’s comfort. The programmer cannot predict everything that could happen on the road. Instead of building lengthy “if-then” instructions, the programmer prepares the reinforcement learning agent to be capable of learning from the system of rewards and penalties. The agent (another name for reinforcement learning algorithms performing the task) gets rewards for reaching specific goals.

**Another example:** deepsense.ai took part in the[“Learning](https://deepsense.ai/learning-to-run-an-example-of-reinforcement-learning/)[to](https://deepsense.ai/learning-to-run-an-example-of-reinforcement-learning/)[run”](https://deepsense.ai/learning-to-run-an-example-of-reinforcement-learning/)[project,](https://deepsense.ai/learning-to-run-an-example-of-reinforcement-learning/) **which** aimed to train a virtual runner from scratch. The runner is an advanced and precise musculoskeletal model designed by the [Stanford Neuromuscular Biomechanics](https://www.crowdai.org/organizers/stanford-neuromuscular-biomechanics-laboratory) [Laboratory.](https://www.crowdai.org/organizers/stanford-neuromuscular-biomechanics-laboratory) Learning the agent how to run is a first step in building a new generation

****

*26*

of prosthetic legs, ones that automatically recognize people’s walking patterns

and tweak themselves to make moving easier and more effective. While it is possible and has been [done in Stanford’s labs,](https://www.ncbi.nlm.nih.gov/pubmed/26258930) hard-wiring all the commands and predicting all possible patterns of walking requires a lot of work from highly skilled programmers.

The main challenge in reinforcement learning lays in preparing the simulation

environment, which is highly dependant on the task to be performed. When the model has to go superhuman in Chess, Go or Atari games, preparing the simulation environment is relatively simple. When it comes to building a model capable

of driving an autonomous car, building a realistic simulator is crucial before letting the car ride on the street. The model has to figure out how to brake or avoid a collision in a safe environment, where sacrificing even a thousand cars comes at a minimal cost. Transferring the model out of the training environment and into to the real world. Scaling and tweaking the neural network controlling the agent is another challenge. There is no way to communicate with the network other than through the system of rewards and penalties. This in particular may lead to *catastrophic forgetting*, where acquiring new knowledge causes some of the old to be erased from the network Yet another challenge is reaching a local optimum – that is the agent performs the task as it is, but not in the optimal or required way. A “jumper” jumping like a kangaroo instead of doing the thing that was expected of it-walking-is a great example. Finally, there are agents that will optimize the prize without performing the task it was designed for. An interesting example can be found in the OpenAI video below, where the agent learned to gain rewards, but not to complete the race.

**2.4 The learning problem: an overview**

Let us revisit Inductive learning and describe it more formally. Given examples of a function (X, F(X)) pair where X is the input of vector values that are either continuous or discrete. Eg: Symptoms of a patient (temp, blood pressure, glucose levels etc. And, F(X) is the value of the function for that element. Eg: Diagnosis of the patient -”yes, the patient has diabetes” or”no, the patient does not have diabetes”. The crux is how can we predict F(X) or”generalize” for new examples - data that we have not seen before?



*27*

There three types of supervised learning that needs to be mentioned:

**1.** Discrete F(X): classification. Eg: Computer vision system that wants to labelthe object that is seen (chair or table, say). As we are predicting the class of the object, hence this is known as a classification problem.

**2.** Continuous F(X): regression. Eg: Predicting the gas mileage of a car given itscharacteristics?

**3.** F(X) = Probability(X): probability estimation (if that is what we arepredicting). Eg: Google might be interested in learning the probability of a user clicking on a particular ad? A harder problem would be predicting several things at the same time.

This brings us to the conclusion this chapter by stating the quintessence of the learning problem via the simple example below:

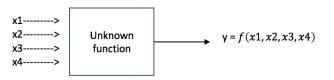


Figure 2.4: The Learning Problem

Suppose we are given a black box that represents a Boolean function. The input to this function are four Boolean inputs, x1, x2, x3, x4, each with a value of either 0 or 1. The output is y a y.

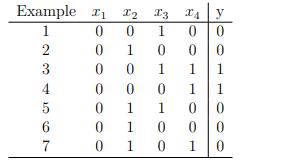


Table 2.5: An unknown Boolean function with 7 examples, each comprising of four Boolean inputs.

Each set of 4-values is known as an example. For these seven examples, we know what the outputs are - so what is y? If we are given an additional example set, what



*28*

would the corresponding value for y be? This, in essence, the inductive learning problem.

**2.5 Shannon's Entropy**

Entropy is the possibility measure of uncertainty. Entropy's responsibility is to control or how a decision tree breaks the data.When entropy is zero, then the dataset is perfectly classified. The equation of information entropy is- ,

( ) = − ∑| | =1 log2

Information Entropy of Has\_Diabetes

4 4 4 4

IE(Has\_Diabetes) = − 8 log2 8 − 8 log2 8

There are 3 occurrences of middle age. Has\_Diabetes of 3 items are no and 0 items are yes as shown below-

IE (Has\_Diabetes | Age=middle) = 0 − 33 log2 33 =0

There are 2 occurrences of old age. Has\_Diabetes of 2 items are yes as shown below-

IE (Has\_Diabetes | Age=old) = − 22 log2 22 − 0 = 0

There are 3 occurrences of young age. Has\_Diabetes of 1 item is no and 2 items are yes as shown below-

IE (Has\_Diabetes | Age=young) =− 23 log2 23 − 13 log2 13 = 0.92



*29*

**2.6Information Gain**

So far, we have computed the entropy for any one given subset. Note that for each node in a decision tree, there may be several children) multiple subsets. Thus, we need to take an average of the entropy for all subsets per node.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ( ) = − |  | log2 |  | | − |  | log2 |  |  |
| + |  | + | + | + |  |
|  |  |  |  |  |  |  |  |  |  |

,

( , ) = ( ) − ∑ + × ( )

=1 +

In conclusion, by taking every A that we have in our data, we compute the information gain for that A. We select the A that has the highest information gain. We implement this approach because that A will reduce our uncertainty the most and

it will lead to the purest possible split out of all the As. If we have mixed sets as Children, we then recursively compute the information gain.

**2.7Conclusions**

From this chapter we know the details about the related work of our thesis work. We have learnt many things by studying all these systems about Learning Methods and classification. Moreover, we gather knowledge and also came to learn about much functionality related to Machine Learning (ML) and also about the modules from this existing systems we discussed in this chapter. These ML systems were very useful for developing our system and to get the concept of developing better idea on this work. Now a day ID3 algorithm may not be very popular all over the world. But we hope that it will be in the upcoming future as this system and method is easy to understand than the existing system and also the performance is higher.



*30*

**Chapter-3**

**Decision Tree Learning**

**3.1** **Problem description and goals**

As stated earlier, a decision tree is a way to represent a ML program. It is easiest to get started with a concrete example.

We need to calculate information entropy and next measure information gain. To use the latest rule, we calculate these Information Entropy Calculate.

Here, give us some training sample and Has\_Diabetes consists of 8 columns. with two attributes: yes and or no.

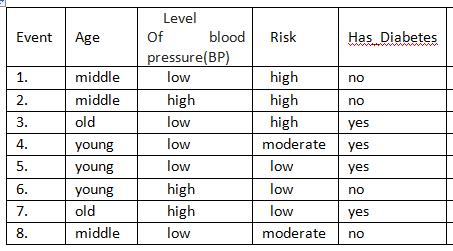


Table 3.1: Observations of having diabetes

**Goal**: Build a mechanism such that on Event 8 (unseen example), it would be able toclassify if a person has diabetes or not?



*31*

**3.2 How does it work?**

Decision trees works in a”divide and conquer” approach and while and while implementing recursion. The general idea is:

* 1. Split the data set (eg: 2.1) into disjoint subsets based on all values per attribute.
  2. Check if those subsets are pure (i.e. if the target value are all”yes” or”no”) or not?
     1. If yes, then STOP! We do not need to make any further

decisions.

* + - 1. If no, then we repeat the process, minus the attribute that we

just considered.

1. Continue the recursive call, until we are left with pure sets.

As a side note, it is acceptable use this approach considering that the toy dataset model in question only has 14-examples. In the real-world, the training datasets that practitioners interact with would be orders of magnitude in size. . In those situations, the standard practice is to partition the datasets into two subsets:

* A larger portion: training dataset (used for training the decision tree).
* A smaller portion: testing dataset (used for evaluating the accuracy of

decision tree).

**Overall goal:** Once we have a working decision tree (data structure), when we aregiven an unseen example/instance, we need to be able to predict the target. This is accomplished by traversing through the tree.

**3.2.1 Example - Age attribute**

Let us look at an example. From 3.1, consider the”Age” attribute. It has three distinct values: Middle, Young and Old.



*32*

1. First, we split the training dataset into three-disjoint subsets.
   * 1. Look at the target attribute to see if Roger plays consistently or not
        1. If yes, we can conclude that for that value of the attribute, it leads to John playing or not playing.
           1. If no, we repeat the process of splitting those disjoint subsets further, but removing the “Outlook” attribute value from those examples.
   1. Continue this recursively such that there is no uncertainty of whether John plays or not – left with pure sets in the end.

**3.2** **Algorithm**

The algorithm of the proposed ID3 technique is shown below and the corresponding flowchart is demonstrated in Fig. 2.

* Create a root node for the tree
  + If all examples are positive, Return the single-node tree Root, with label = +.
    - If all examples are negative, Return the single-node tree Root, with label = -

.

* If number of predicting attributes is empty, then Return the single node tree

Root, with label = most common value of the target attribute in the examples.

* + Else

– A = The Attribute that best classifies examples.

– Decision Tree attribute for Root = A.

– For each possible value, vi, of A,

* + - * Add a new tree branch below Root, corresponding to the test A = vi
    - Let Examples (vi), be the subset of examples that have the value vi for A
      * If Examples (vi) is empty

– Then below this new branch add a leaf node with label = most common target value in the examples

* + - * Else below this new branch add the subtree ID3 (Examples (vi), Target Attribute, Attributes – {A})

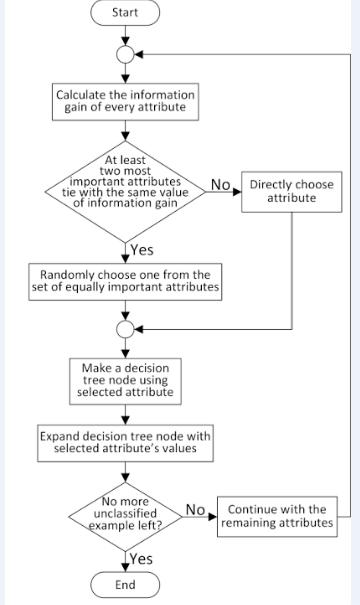


*33*

• End

* Return Root

**3.4 Flowchart**



**Figure 3.6: Flowchart of ID3 Algorithm**



*34*

**3.5 Advantages**

The main advantage of our system is that, we tried to develop a model that will result with better accuracy level. By this method we can gain a certain level of accuracy as we may need according to situation each time. For different situation it may be time efficient too while we may need less accuracy. It can also be faster in that case of operation. So, we can summarize it by following points:

1. Understandable prediction rules are created from the training data.
2. Builds the fastest tree.
3. Builds a short tree.
4. Only need to test enough attributes until all data is classified.
5. Whole dataset is searched to create tree.

**3.6 Conclusions**

In this chapter we have discussed about total procedure of our system. We provided all the necessary discussions. We added the necessary diagrams for each steps of the system for explaining. We also tried to discuss all the details of the system as easy as it can be. For better understanding we added a flow chart of the proposed model Moreover, the algorithm and a full example is discussed in this chapter. All the calculations were shown step by step with tables and values also. Finally we also tried to make understand why our system is better and the advantages of this proposed model.



*35*

**Chapter – 4**

**Experimental Results**

**4.1 Introduction**

The purpose of testing is to identify errors. Testing is the method of trying to discover every understandable error or defect in a work. It gives a way to check how much correct the result of any input is. It is the process of any system with the intent of ensuring that the system meets its obligations and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. By testing and analyzing the experimental result a system can be developed more for higher accuracy and better performances.

This chapter mainly describes the qualitative practice to be used to provide data to examine the issues acknowledged and extend the understanding of ID3 algorithm. We tried to by these tests.

**4.2 Result Analysis**

At the iteration, we need to know which is best attribute to be chosen as top root in our decision tree. To do that, ID3 will find the *best attribute* which is has maximum information gain. Given the information gain for each attribute. To get information gain, every column should be calculated in the same way.

IG(Has\_Diabetes, Risk) = 0.06

IG (Has\_Diabetes, Level of BP) = 0.05

Compare all the information gain and we see that information gain of age has the highest score which is the parent node of the decision tree.



*36*

The first decision tree is-

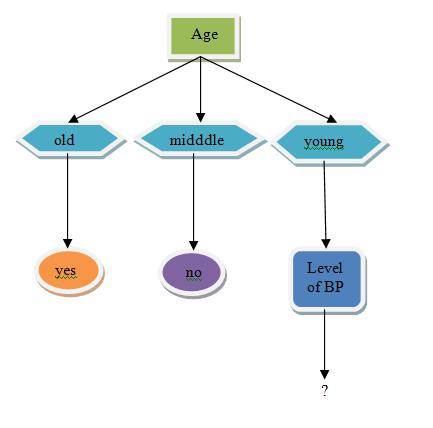


Figure 4.2: First Decision Tree

IG( young,Risk) = 0.25

IG( young, evel of BP) = 0.92

Compare all the Information Gain and the Information Gain of the level of BP is greater than the Information Gain of Risk. That's why we choose the Level of BP node.

The final decision tree is-



*37*

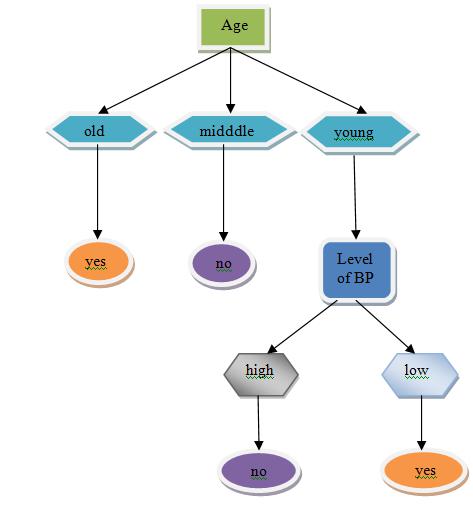


Figure 4.3: Final Decision Tree

**4.3 Applications**

**ID3 Application on Food Database**

Ashvini Kale, Nisha Auti[7] in their research explained about the use of ID3 algorithm in implementation of Automatic menu planning for children as recommended by dietary management system. This research was carried out using Indian food database as many of the Indian children are affecting to mal nutrition due to mothers ignorance on nutrition facts. Approximately 30% of the new born children are having problems of low weight and hence easily susceptible to diseases. Vitamins and mineral deficiencies also affect children’s survival and development. Anemia affects 74% of children below the age of three, more than 90% of adolescent girls and 50% of women. The proposed method of food suggestion for children is based on the



*38*

factors such as food preferences, availability of food, medical information, disease information, personal information, activity level of a child, for Indian food database. The important task in implementation is to recommend the particular food item from the food database based on certain attributes such as likeliness, availability, its nutritional contents and decision of child.

**Web Attack Detection**

Using ID3 In a research by Garcia, YH, Monroy, R., Quintana, M.,[8], explained how ID3 can be used in detection of web attacks. As today’s smart technologies enable every operation to perform online, transactions has increased a lot. At the same time, the attacks on these online sites have also increased. Hence many organizations prefer an Intrusion Detection System. The main problem of existing IDS is that they cannot detect mimicry attacks and new attacks as this problem prevails in well known IDSs, like snort. To solve this issue, IDS researchers have turned their attention to machine learning techniques, including classification rules and neural networks. In their experiments unlike other IDS, ID3 was able to classify even unseen Web application queries as an attack. To classify this, they used a training set of 400 web application attacks queries from three vulnerabilities, and also gathered 462 web application non attacked queries. They used various window sizes (5, 8, 10, 12, 15) among which size 10 gave best decision tree which precisely captured the examples. After building a decision tree, inputs were assigned to ID3 and they indicate both the false alarm rate and the missing alarm rate, considering two sets of attacks. The results obtained prove that IDS is a competitive alternative for detecting Web application attack queries by using ID3.

**Application of ID3 in Diabetes**

Diabetes is one among the challenging diseases that the human race is finding difficulties which is prevailing since years. Many of the analysts around the globe have been working on diabetes and making aware of the signs and effects of the disease on the various organs of the body.



*39*

**ID3 in Identifying Cancer**

Priyadharsini.C, Dr. Antony Selvadoss Thanamani et. al. [12] have analyzed an important process of identifying cancer in early stages using ID3 algorithm. As we know cancer has become a dreadful disease and affecting many people’s health, this analysis helped to identify the early stages of cancer. For this analysis, they used Multidimensional Array model with modified ID3 algorithm. The modified ID3 algorithm compares the current database with the previous dataset and identifies the results as positive or negative. In case a patient is affected with that disease, this algorithm shows the infection percentage [12]. Here ID3 is used to split training examples in to target classes, the one which gives highest classification is selected and Used.

**Application of ID3 in Computer Forensics**

Data analysis is the most crucial part in computer crime forensics system. The result of data analysis has a direct impact on the validity and credibility of the evidence. In the prototype system of computer crime forensics, the general practice is making use of ID3 algorithm directly, but in this way it does not effectively mine a reasonable model because of the versatility of ID3 algorithm and the uniqueness of forensic data. According to characteristics of computer crime forensics data, this paper [13] puts some improvements of ID3 algorithm in terms of information gain to make it more suitable for computer crime forensics field data, and experiments show that the improved algorithm is effective. As the diversity of attacks, in the extraction of features and attributes of behaviors, if we still choose the largest value of information gain as the property of division to construct division tree, it will generate very much redundant information, and even result in error message when march between the input event and the rule base.

**Application of ID3 in Knowledge Acquisition for Tolerances Design**

In a research on Knowledge Acquisition by Xinyu Shao, Guojun Zhang, Peigen Li, and Yubao Chen [14], ID3 algorithm has been improved using previous knowledge. Tolerance Design is the total amount by which a given dimension may vary, or the



*40*

difference between the limits. Tolerance engineering affects areas like Product design, Quality Control and Manufacturing. Knowledge processing can be used to aid engineering design. Knowledge processing technology is utilized in Intelligence and can be incorporated in existing CAD systems. Prior to implementation of ID3 some premises should be checked, they are:

1. Tolerance Description
2. Function satisfied
3. Parting line related
4. Mold design (Cool well designed and Gate well designed)
5. Machine capability.
6. Design

**Use of ID3 for Breast Tumor Diagnosis**

Decision tree classifiers are used extensively for diagnosis of breast tumor in ultrasonic images, ovarian cancer and heart sound diagnosis. D.Lavanya, Dr. K.Usha Rani [17] in their research on various decision tree algorithms showed that ID3 is used majorly. They compared the time complexities for ID3, CART, and C4.5 for different diseases and concluded that time complexity of ID3 algorithm is less to build a model among the three classifiers but Accuracy is very less compared to CART, which further needs to be improved. ID3 in multi array model algorithm is explained as follows: E = D1× D2× ... × Dn be finite-dimensional vector n, where Dj is a finite set of discrete symbols, E elements e = is the sample, vjDj, j = 1, 2, ..., n. PE is the positive sample set, NE is the anti-sample set, and the number of samples which are p and n depiction to the regulations of information theory. The proposed sample data used by ID3 has certain requirements.

**4.4 Conclusions**

This chapter focused on ID3 Decision Tree algorithm for classification. The present study reviewed Robust Decision tree algorithm ID3 and its applications in wide range spectrum of domains such as Health, medical, Education, Engineering etc. Across all the domains, the performance of ID3 has resulted in good performance. However, splitting criterion and pruning can be further improved to achieve higher accuracy and



*41*

generalization. A minute increase in performance and generalization will yield better results and analysis, particularly in Health care domain. Hence our future work focuses on developing a simplified decision tree algorithmic model by using a novel splitting criterion and a pruning technique, with the objective of increasing accuracy and generation.



*42*

**Chapter – 5**

**Conclusions**

**5.1 Conclusion**

In conclusion, in this report we first introduced the incredibly versatile and powerful area in Computer Science known as Machine Learning. We explored the key difference between the traditional programming and ML paradigms. Namely that while in the former, a program is one of the inputs to a computer and after computation, we get an output from the computer, in ML, the computer takes in both the data and the output as inputs and gives another program as the output. We explored that with an continual increasing number of applications where ML algorithms are becoming not just useful but the standard practice; every ML algorithm has three components: representation, evaluation and optimization. For the purposes of this report, decision tree learning is our chosen representation that is used to evaluate classification problems and its effectiveness is based upon how accurate it is able to do so. Decision trees implement a form of combinatorial optimization, namely doing a greedy search of which attribute in a dataset yields the maximum gain. We also explored the three primary types of learning: supervised, unsupervised and reinforcement learning and ended our introduction to ML by discussing the learning problem. We introduced decision tree learning using Mitchell’s [5] classic prediction problem of determining if a person plays tennis or not, depending on weather conditions. Despite being no where near as complicated as datasets used in practice today, the author found it particularly effective in motivating the ID3 algorithm that builds the decision tree data structure. We delineated how the ID3 algorithm is formulated and went into great detail regarding concepts such as Shannon’s Entropy and Information Gain. Finally, upon understanding how a decision tree is formulated, we then further investigate if the output of the ID3 algorithm always yields perfect solutions we learned that according to 4.1 when it came to training a decision tree,



*43*

due to the recursive nature of the algorithm the tree would continue splitting until singletons were obtained. While this does fit the bill when asking for a perfect solution, this may not be what we want as singletons do not help us when making predictions for massive datasets. Finally, we concluded this report by outlining two approaches that could help improve the ID3 algorithm. First, by understanding that while information gain helps 21 us identify the ”best” attribute to split our datasets, it may not give us desirable results based on its inherent greedy nature. We resolve this issue by introducing the notion of Grain ratio. Secondly, we outline the Random Forest Algorithm that extends the ID3 algorithm by creating K-trees, that are created/trained using subsets of examples/attributes from initial dataset, instead of just one. When making a prediction we input the unseen example into all K-trees and classify the example based on the majority output of all K-trees.

**5.2 Limitations**

We tried to develop a system so that this algorithm can be recognized in a easy, fast and more accurate way. We also tried it to be dynamic so that it can be used in different critical situations. For our work we used powerful tools and updated software and technologies. Still our proposed system may face some drawbacks and some are listed below:

1. Data may be over-fitted or over-classified, if a small sample is tested.
2. Only one attribute at a time is tested for making a decision.
3. Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.
4. To get the accuracy higher, the range of a net's weights and of the weight updates is very important for performance. We may need to normalize the input values in some cases when the features are different like in many of our given datasets and updates will all be on different systems.



*44*

**5.3 Future Works**

Our system can be used in various important different works and can be implemented easily. Although a lots of works are being done each and every day of this world This thesis presented a survey.. We thought about our work to be used in future works. Some are mentioned bellow.

1. We can achieve more accuracy can be achieved in many sectors based system through various problems using this system based on deep learning.
2. We can implement this system for automated traffic system from Traffic Signs Recognition and in this way we can develop an automated traffic system and can solve the traffic problems in many countries.
3. Space research field recognizes planetary star images and their movements and positions can be tracked by using this system as I will use unsupervised learning and deep learning so it will be easy apply in unknown situations
4. We can use this system to retrieve weather preview after experiencing weather image recognition and using the results we can develop systems for predicting weather conditions or can predict dangerous situations earlier to save lives and properties



*45*

**References**

1. Punlumjeak, Wattana, Nachirat Rachburee, and Jedsada Arunrerk. "Big data analytics: student performance prediction using feature selection and machine learning on microsoft azure platform." Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9.1-4 (2017): 113-117.
2. Sweeney, Mack, et al. "Next-term student performance prediction: A recommender systems approach." *arXiv preprint arXiv:1604.01840* (2016).
3. Mohammadi, Mehdi, et al. "Comparative study of supervised learning algorithms for student performance prediction." *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2019.
4. Kaur, Harleen, and Er Gourav Bathla. "Student performance prediction using educational data mining techniques." *International Journal on Future Revolution in Computer Science & Communication Engineering* 4.12 (2018): 93-7.
5. Parmar, Krina, Dineshkumar Vaghela, and Priyanka Sharma. "Performance prediction of students using distributed Data mining." *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE, 2015.
6. Xingjian, S. H. I., et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." *Advances in neural information processing systems*. 2015.
7. Ma, Yuling, et al. "Pre-course student performance prediction with multi-instance multi-label learning." *Science China Information Sciences* 62.2 (2019): 29101.
8. Al-Shehri, Huda, et al. "Student performance prediction using support vector machine and k-nearest neighbor." *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2017
9. Kabakchieva, Dorina. "Student performance prediction by using data mining classification algorithms." *International journal of computer science and management research* 1.4 (2012): 686-690.
10. Devasia, Tismy, T. P. Vinushree, and Vinayak Hegde. "Prediction of students performance using Educational Data Mining." *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*. IEEE, 2016.
11. Hasan, Raza, et al. "Student Academic Performance Prediction by using Decision Tree Algorithm." *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*. IEEE, 2018.
12. Hasan, Raza, et al. "Student Academic Performance Prediction by using Decision Tree Algorithm." *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*. IEEE, 2018.
13. Luo, Ling, Irena Koprinska, and Wei Liu. "Discrimination-Aware Classifiers for Student Performance Prediction." *International Educational Data Mining Society* (2015).
14. Xu, Jie, et al. "Progressive prediction of student performance in college programs." *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.
15. Saa, Amjad Abu. "Educational data mining & students’ performance prediction." *International Journal of Advanced Computer Science and Applications* 7.5 (2016): 212-220.
16. Gadhavi, Mahesh, and Chirag Patel. "Student final grade prediction based on linear regression." *Indian J. Comput. Sci. Eng.* 8.3 (2017): 274-279.
17. Jamil A, Ahsan M, Farooq T, Hussain A, Ashraf R. "Student Performance Prediction Using Algorithms of Data Mining". In2018 International Conference on Computing, Engineering, and Design (ICCED) 2018 Sep 6 (pp. 244-249). IEEE.
18. Kumar, M., Singh, A.J. and Handa, D., 2017. "Literature survey on student’s performance prediction in education using data mining techniques." *International Journal of Education and Management Engineering*, *7*(6), pp.42-49.
19. Sahebi, Shaghayegh, and Peter Brusilovsky. "Student Performance Prediction by Discovering Inter-Activity Relations." *International Educational Data Mining Society* (2018).
20. Roy, S., Nag, S., Maitra, I.K. and Bandyopadhyay, S.K., 2013. "International Journal of Advanced Research in Computer Science and Software Engineering." *International Journal*, *3*(6).
21. Zhao, Z., Zhang, L. and Hu, D. 2016 August, "Student Performance Prediction Method Based on Fuzzy Cognitive Model." In *2016 International Conference on Education, E-learning and Management Technology*. Atlantis Press.
22. Bydžovská, H., 2015, June. " Student performance prediction using collaborative filtering methods." In *International Conference on Artificial Intelligence in Education* (pp. 550-553). Springer, Cham.
23. Omisore, O. M., and N. A. Azeez. 2015 "Predicting Academic Performance of Students in Higher Institutions with k-NN Classifier." International Conference on Computer Science Research and Innovations. .



*50*