Learning of Contextual Bandits using Adaptive Neuro Fuzzy Inference System

THESIS SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE OF

Master of Technology
in
Software Engineering

Under the guidance of
Mr. Nipun Bansal
(Assistant Professor, Department of Computer Science and Engineering)
Delhi Technological University

Submitted By -
Paramveer Singh
(Roll No. 2K17/SWE/13)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road, Delhi-110042

June 2019

DECLARATION

I hereby declare that the thesis work entitled "***Learning of Contextual Bandits using Adaptive Neuro Fuzzy Inference System***" which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of degree of Master of Technology (Software Engineering) is a bonafide report of Major Project-II carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Place: Delhi   Paramveer Singh

Date: Roll No. 2K17/SWE/13

CERTIFICATE

This is to certify that Project Report entitled “***Learning of Contextual Bandits using Adaptive Neuro Fuzzy Inference System***” submitted by Paramveer Singh (roll no. 2K17/SWE/13) in partial fulfilment of the requirement for the award of degree Master of Technology (Software Engineering) is a record of the original work carried out by him under my supervision.

Place: Delhi

Date:

SUPERVISOR

Mr. Nipun Bansal

Assistant Professor

Department of Computer Science and Engineering

Delhi Technological University

Bawana Road, Delhi -110042

ACKNOWLEDGEMENT

I am very thankful to **Mr. Nipun Bansal** (Assistant Professor, Computer Science and Engineering Department) and all the faculty members of the Computer Science and Engineering Department of Delhi Technological University. They all provided us with immense support and guidance for the project.

I would also like to express my gratitude to the university for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Paramveer Singh

Roll No. 2K17/SWE/13

M. Tech. (Software Engineering)

Delhi Technological University

ABSTRACT

In the Traditional computer science, we program the computer to achieve some task. In this paradigm we give instructions to a computer to do its task successfully, but now a days we are moving towards a new paradigm which is called machine learning. In this paradigm we provide some labelled examples to a machine, with the help of which it will automatically derive the rules and patterns and save the extracted information to predict the testing data i.e. machine learns to accomplish the task based on the examples provided by us. For example, in Natural Language Processing we give the two large corpus of documents to the machine as an example and the machine will learn to discover patterns in order to match the right words and right expression to go from one language to another. If we try to write the rules for the same then we have to write immunes amount of code and we may not be able to write all the translations form one language to another.

In this report, we will illustrate the reinforcement learning using ANFIS (Adaptive Neuro Fuzzy Inference System). The different number of datasets are used as an experiment in the purposed model which is Contextual Multi-Armed Bandit Algorithm named as Adaptive Neuro Fuzzy Inference System (ANFIS). UCI which is a machine learning repository provided the datasets related to these kinds of problems i.e. these datasets comes under the category of CMAB problem.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1.	MAB:	Multi-armed Bandits
2.	Diff:	Difference
3.	BTW:	Between
4.	S_i:	States
5.	A_i:	Actions
6.	R_i:	Rewards
7.	ANFIS:	Adaptive Neuro Fuzzy Inference System

CHAPTER 1

INTRODUCTION

1.1 MACHINE LEARNING TECHNIQUES

In the Traditional computer science, we program the computer to achieve some task. In this paradigm we give instructions to a computer to do its task successfully, but now a days we are moving towards a new paradigm which is called machine learning. In this paradigm we provide some labelled examples to a machine, with the help of which it will automatically derive the rules and patterns and save the extracted information to predict the testing data i.e. machine learns to accomplish the task based on the examples provided by us. For example, in Natural Language Processing we give the two large corpus of documents to the machine as an example and the machine will learn to discover patterns in order to match the right words and right expression to go from one language to another. If we try to write the rules for the same then we have to write immunes amount of code and we may not be able to write all the translations form one language to another.

There are different types of Machine Learning.

1. **Supervised Learning:** In this we provide the lots of labelled data to the machine. So that it will extract some information form it and used it in the future. Bottleneck for this type of learning is that we need lots of labelled data which may not be possible for many applications.
2. **Unsupervised Learning:** By this learning we train our machine by using the dataset which is neither labelled nor classified. It allows the algorithm to do some task on the given information without any guidance. Unlike supervised learning, no teacher is provided. We mostly do cluster in this type of learning.
3. **Reinforcement Learning:** It is inspired from psychology, behavioural psychology. Similar to trained the animals to behave in a certain way without talk to them directly. Similarly, we train the machine by giving some positive or negative rewards according to the action taken by the machine.

In Reinforcement learning, there are some agents which will take some actions in an environment and the goal is to get the maximum cumulative reward. As we know that in this type of learning we do not use the labelled data instead of this there are multiple positive or negative rewards which will be used by the algorithm. We use some mathematically numerical functions and the machine will simply optimize or maximize the given function.

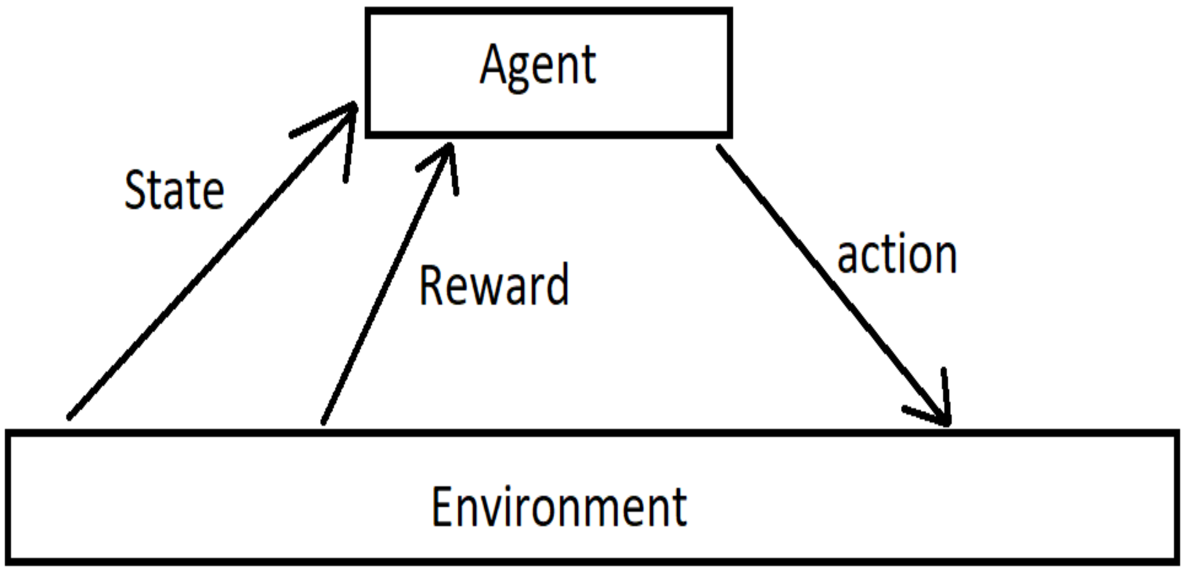


Fig. 1.1 Reinforcement Learning

The above figure 1.1 shows the framework used is the reinforcement learning. The agent executes some actions. These actions will be performed on the environment and the environment give some rewards to the agent and agent will move to some state according to the reward given by the environment. Here we characterize our environment by some notion of states that means it is the description of the status of the environment and based on this agent take some actions and get some rewards. By this machine goal is to learn to choose actions that will give maximum reward. This reward is going to be some numerical value.

Some Examples of Reinforcement Learning

- Game Playing (Go, Atari, backgammon)
- Operations research (pricing, vehicle routing)
- Robotics
- Helicopter Control
- Self-managing network systems

- Data centre energy optimization
- Computational Finance

Operations Research: Vehicle routing

- **Agent:** vehicle routing software
- **Environment:** Stochastic demand
- **State:** vehicle location, capacity and depot request
- **Action:** vehicle route
- **Reward:** travel cost (negative)

Game Playing: Go (one of the oldest and hardest board game)

- **Agent:** player
- **Environment:** opponent
- **State:** configuration
- **Action:** Next stone location
- **Reward:** +1 if win / -1 if loose

Conversational Agent

- **Agent:** Virtual assistant
- **Environment:** User
- **State:** conversational history
- **Action:** next utterance
- **Reward:** points based on task completion, user satisfaction, etc

1.2 MULTI-ARMED BANDITS

This is a special class of sequential optimization problems This is the challenge here; we want to maximize the rewards using fixed number of choices. To explain this let's take an example, in a casino there are k-bandit machines. Single machine can be chosen by a gambler in a single round, but the gambler does not know about the winning probability of any machine so, he has to play on ever machine in the beginning to know about the machines. He has to switch between the maximum reward giving machine according to his current knowledge which is called **exploiting** or test the new machine to increase its knowledge which is called **exploring**. So, we use the multi-armed bandit problem in the areas where we are not sure to choose between exploiting and exploring.

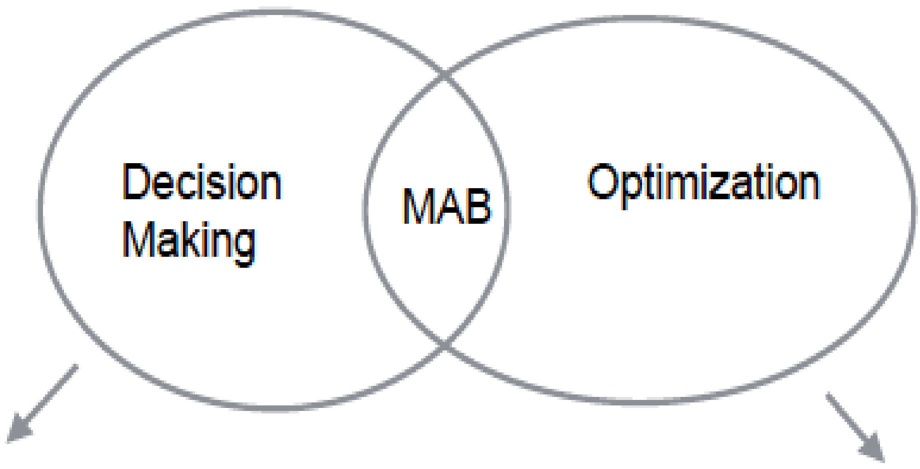
Below figure 1.2 shows the application area for reinforcement learning (Contextual Bandits)

Figure 1.2 Application areas

Let’s take an example to understand it better.

Robotic Control: Helicopter control

- **Agent:** Controllers. These are used to control the helicopter.
- **Environment:** helicopter
- **State:** position, orientation, velocity and angular velocity
- **Action:** collective pitch, cyclic pitch, tail rotor control
- **Reward:** deviation form desired trajectory (negative)
- **Exploration:** Learning.
- **Exploiting:** Getting maximum reward on basis of current knowledge.



Choose the Best way to fly

Minimize total regret by avoiding the crash

Figure 1.3 MAB Big Picture

We can classify the multi-armed bandit strategies mainly into two categories.

- Non-Contextual multi-armed bandit
- Contextual multi-armed bandit

To understand these, consider a restaurant recommendation system. System will find the best restaurant according to the user that means it has to minimize the cumulative rewards. On the other hand, it has to try to reduce the regrets by sending users to the wrong restaurant which are not suitable for the users. Below figure shows the difference between the multi-armed bandit categories. Non-contextual multi-armed bandit simply choose the restaurant which is good for everyone it does not choose according to the user feature. While contextual multi-armed bandit choose restaurant according to the user and his/her features.

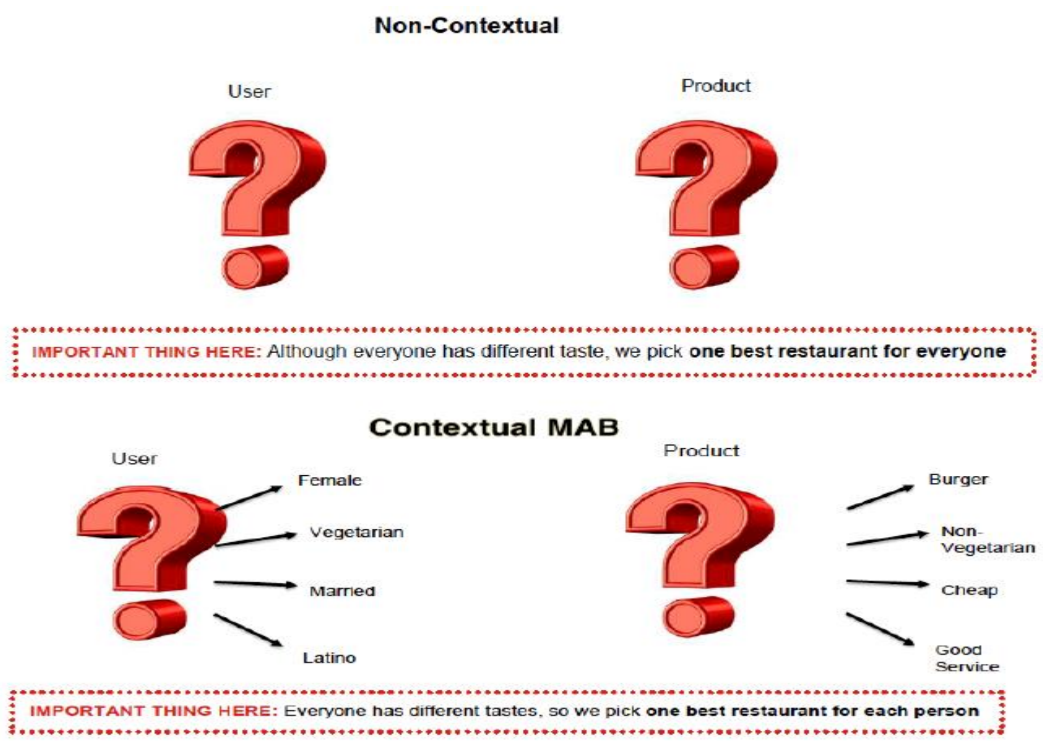

Non-Contextual
User
Product
IMPORTANT THING HERE: Although everyone has different taste, we pick one best restaurant for everyone
Contextual MAB
User
Female
Vegetarian
Married
Latino
Product
Burger
Non-Vegetarian
Cheap
Good Service
IMPORTANT THING HERE: Everyone has different tastes, so we pick one best restaurant for each person

Figure 1.4 Diff. btw contextual and non-contextual MAB

Markov Processes is mathematical framework that underlies reinforcement learning.

It is used to model Environment Dynamics which is evolving.to do this it uses stochastic processes and these processes make two important assumptions. A) Markovian Assumption B) Stationary assumption

If we unroll the control loop of reinforcement learning as explained above leads to a sequence of state, action, rewards. Figure shows the sequence of control loop of reinforcement learning.

$$S_0A_0R_0\ S_1A_1R_1\ S_2A_2R_2\ S_3A_3R_3$$

Figure 1.5 Sequence of state, action and reward

This is just like finite automata; we are on state 0 we take a action and we get a reword and move to the next state. This process continuously going on but the problem here is we do not know anything about he rewards. So this above sequence forms a stochastic process sue to some uncertainty in the dynamics of the process.

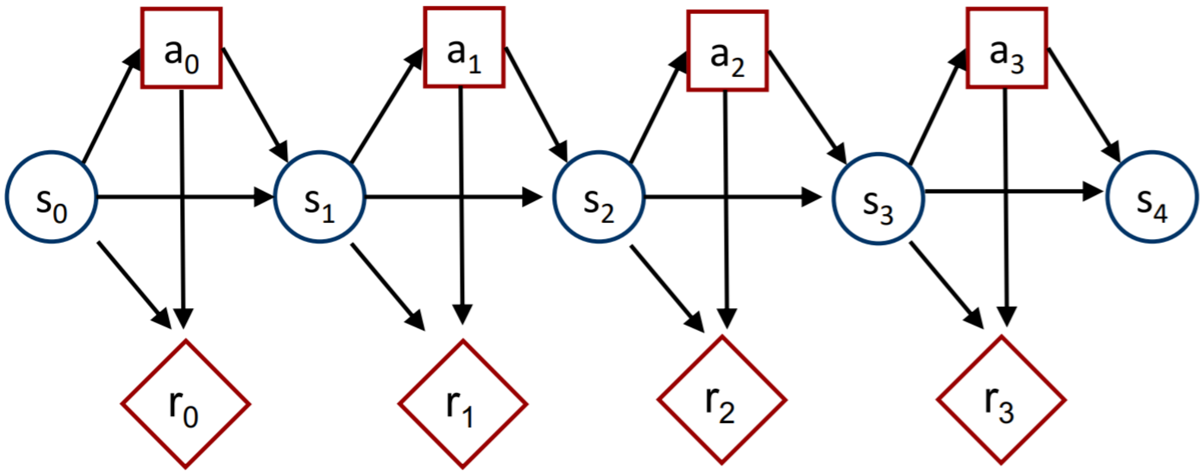


Figure 1.6. Markov Decision Processes

Markov decision processes form the foundation for reinforcement learning. In the above figure circles represent the random variables. Squares corresponds to actions and diamonds corresponds to the rewards or utilities. Arrow shows the decencies. As shown in figure states depends on previous states as well as on actions that is selected at each time stamp. The choice of action depends on the current state because they will describe the environment. So that we can make the best decision.

We make some assumptions to implement our proposed model. The process is not deterministic in nature. The process is stochastic and sequential.

1.3 CONTEXTUAL MULTI-ARMED BANDITS

On every single emphasis, specialist can pick a solitary arm structure the k various arms. The learning rate of the machine does not change. Operator sees a d-dimensional vector before picking between the arms. This vector is otherwise called setting related with cycles for settling on its decision. Operator attempt to discover the connection between the specific circumstance and the reward, over various cycles. We should consider ad rendering framework, the d-measurement vector could be the client's highlights vector in commercial rendering framework. With the assistance of client's specific situation, the framework can customize the includes for every single client as per client inclinations or we can say that the framework will pick those notice which will be bound to be clicked by the specific client.

There are many problems or challenges tightly bound with this problem.

1. **Tenable Parameters:** There are many tuneable parameters in multi-arm bandit problem which are associated with the past solutions. Setting the right parameters is very much difficult because we have very little and no prior information about the user similarly, setting up the wrong parameter may produce negative result or we can say that negative reward.
2. **Counterfactual Learning Problem:** The nature of learning is another type of learning in multi-arm bandits. The system only sees the reward given by the particular arm which he/she chooses.
3. **Stochastic User Behaviour:** Let's take an example to understand it. Suppose a user is searching to buy something online. Using the history of user searches our system will show the relevant advertisement to the user, but when any user already bought that thing then probability of click on that advertisement by the particular user is very vey less. To adapt the behaviour of user the system will take some time. This is known as stochastic user behaviour.
4. **Cold Start:** At first system does not have enough information to draw any inference for the user which uses the system. This is called cold start problem.

In our work, we are trying to solve the above problem by modelling the online multi-armed bandit problem. By using the fuzzy logic, our model would not need a huge amount of data to feed in. Model will produce meaningful results in few amounts of data. This will also solve the cold start problem as mentioned above. The purposed model has better adaptability, that means it will change its behaviour very fast according to user. We utilize a pe-arm model that will assist us with modelling each decision for the client as a different model. This will assist us with providing positive or negative input for the specific are which is pick by us without changing different arms. In future, on the off chance that we need to include or evacuate a few arms structure the framework, this will support us. As clarified above there are number of tuneable parameters, those parameters are difficulties for us as purposed in numerous past strategies by various specialists.

CHAPTER 2
LITERATURE SURVEY

The scientist named Robbins gave the presentation about MAB [2] in his original paper as Successive Choice issue [5]. At first, this issue was tried different things with measurable suspicion. The prizes were conveyed over each arm to settle the lower headed for these highwayman tests [6][8]. Yet, continuously, prizes may not be measurable in nature and the suppositions we make in our model may not genuine. Scientist named Auer [5] given a changed type of introductory outlaw issue. In this variation there was no supposition in regards to age of remunerations. In ill-disposed scoundrels, as foe assume responsibility for the prizes not at all like polite stochastic procedure. There are three sorts of developments which give us the ideal answer for MAB [2]. The definitions are stochastic, ill-disposed [4] and Bayesian [11]. Loads are given to each arm in an EXP-3 try. These loads changed their qualities exponentially for every which gives us ideal outcome. To pick the best arm EXP-4 [6] which is gotten from EXP-3 uses number of various specialists which will assist the calculation with choosing the best arm.

2.1 CONTEXTUAL BANDITS

In the above arrangements we pursue the procedure which has one decision and we attempt to fit the equivalent to all. In any case, it can't be use for the personalization on a for every client premise. During every cycle a setting vector is utilized in every emphasis which will give the data stick that cycle in CMAB issue [1][8][9]. A scientist named Dudik et al utilizes the arrangement dismissal procedure to expel the undesirable or awful strategies in the working set, for example just positive approaches are kept in the working set. Be that as it may, it is exceptionally agonizing to execute for example it's extremely tricky to monitor the good strategies. On the off chance that the ideal approach is evacuated unintentionally the calculation will never recuperate. Banditron named new model presented. It protects weight vectors for each arm and the yield will be an expectation to the arm with the most noteworthy score. To think about our model, we take the banditron model as a base model. To display the normal reward, banditron utilizes a perceptron per arm model. The calculation utilizes the direct model. Confidit [4] is one when we coupled the banditron with upper certainty bound strategies. They give the preferred exhibition over banditron which is its base calculation. To streamline the prizes, closeness data is utilized in the arm space by certain calculations. Non-stationary marauders handled the circumstances where the client conduct changes after some time. This is uncommon.

2.2 EXPLORATION STRATEGIES

As clarified above fortification learning is about investigation or misuse, such a significant number of various sorts to investigation methods are utilized to adjust between the exchange off of investigation and abuse. There are various sorts of investigation techniques which are utilized. Epsilon-insatiable is one of the investigation procedures. It utilizes the gamma which is the investigation parameter to deal with the exchange off between the investigation and abuse. Age Ravenous calculation [6] changes its states between complete investigation and complete misuse steps and these systems give irregular investigation. There is a system named Thompson Testing [2][3][4] which uses heuristic to deal with the exchange off between the investigation and abuse. This technique keeps up the likelihood dissemination for every single arm and with the assistance of testing we pick the one which will foresee best outcome for each preliminary. After the consequence of the specific cycle out then the dispersion of the picked arm is refreshed. LinUCB assume that there is a liner connection between the normal reward and the unique circumstance. When we play an arm LinUCB keeps up the upper certainty bound. It gets more tightly and more tightly every time when we play the arm. The upper certainty bound reveals to us how much the specific arm investigated for example on the off chance that the worth is high than specific arm is investigated less. In this way, the normal reward may differ from real reward. Chapelle and Li are the specialists those demonstrate that really Thompson Examining beats UCB. UCB picks the arm based on the idealistic worth separated from vulnerability esteem. It picks the unexplored arm on the way that the change of the normal reward is greater than the investigated arm for example it investigated the arm that have not been investigated at this point. On various hub Thompson Examining does not pursue the methodology which is utilized by UCB. It utilizes the completely Bayesian Methodology. With the assistance of completely Bayesian methodology it creates its own crook arrangement structure a back circulation, which will be refreshed with the assistance of past remunerations.

2.3 DEEP BAYESIAN MODELS

Deep models [13][14] can beat the need illustrative intensity of the Straight calculations. To express the mind-boggling portrayal in support adapting profound models are generally utilized. For effectively refreshing the arms for example including or expelling the arms structure the model, Neural Highwaymen are utilized which keeps up the neural model for every single arm. Contribution for each neural model is setting vector at each progression and a score is acquired. The arm picked by the model is the one which has the most astounding score. For this situation for investigation system, Epsilon-Eager is utilized. There are various sorts of profound Bayesian strategies, which are investigated in Deep Bayesian Standoff [41]. Bayesian straight relapse is utilized on the last layer of the Neural system in the Neural Direct strategies. This last layer might be covered up to find out about non-linearity.

Dropout [48] is a system which is proposed as of late other than investigation procedures which are referenced previously. It haphazardly zero out the yield of the neuron with a likelihood during its forward pass. It encourages by ceasing us to overfit the neural system. We can consider the to be as the recreation of Thompson Testing for investigating lesser investigated arms. To surmised the testing dissemination, Bootstrapping [11][12] is utilized. It prepares the distinctive K models on various N datasets however these N datasets ought to be the subset of the first dataset. By utilizing this we can recreate Thompson Examining. Its choses any model which has likelihood 1/q. After this it will choose, he best arm which is anticipate by this. Group [6] is made by this. Bootstrapping strategy is utilized by Choice Trees [1][18] in a system, which uses the Choice trees as the base student to reproduce the Thompson Inspecting. Choice Trees don't require hyper-parameter tuning. Nonetheless, in the past methods various kinds of parameters and constants are utilized and by and by and large those parameters impractical to find in the ongoing.

Isotropic Gaussian clamor is utilized by Direct Commotion Infusion [39] procedure, which is as of late purposed. it is an investigation strategy. This procedure utilizes the isotropic gaussian commotion to annoy the model loads while choosing activities. The loads of the system ought to be standardized. The Greatness of Gaussian Commotion is kept up so the bothers created by the Gaussian Clamor are on a similar scale as the epsilon-covetous with rotting epsilon. On the model parameters, bothers are performed. Along these lines, we expected that it will give the preferred outcome over epsilon-greedy.

CHAPTER 3
ALGORITHMS

3.1 EXPLORATION STRATEGIES

3.1.1 Epoch Greedy Algorithm

Epoch Greedy is an algorithm which solves our problem of exploration by just exploring in epochs. It also solves the dilemma between the exploration and the exploitation by simply changing between the different phases of the complete exploration and complete exploitation. Now we need to know that we have to divide the time horizon into the phases of the exploration and exploitation. To solve this problem, researcher purposed that epoch greedy algorithm do pure exploration or pure exploitation at any instance of time or we can say that in a particular phase. So, if we know beforehand that there are total number of X steps then it should explore the first X' steps. After that it will start exploitation for remaining X-X'. This step is performed by the model because the model has no prior knowledge from exploration. When there is no knowledge no need of exploiting. However, given that T is generally unknown, so Epoch greedy algorithm is run in mini-batches, in a number of epochs: The below shows the epoch Greedy algorithm.

```
**Require:** s ($W_t$): exploitation steps given samples $W_l$
Init exploration samples $W_0 = \{\,\}, t_1 = 1$
For $l$ = 1, 2, … do
    t = $t_l$  >>> One step of exploration
    Draw an arm $a_t \in \{1, \ldots\ldots, K\}$ $uniformly\ at\ random$
    Receive reward $r_{at} \in [0,1]$
    $W_t = W_{l-1} \cup (x_t, a_{t,} r_{at})$
    Solve $\hat{h}_l = max_{h \in H} \sum_{(x,a,r_a) \in w_l} \frac{r_a \prod(h(x)=\alpha)}{1/K}$
    $t_{l+1} = t_l + s(W_l) + 1$
    For t = $t_l + 1, \ldots\ldots, t_l - 1$ do  >>> s$w_l$ steps of exploration
```

```
            Select arm $a_t = \widehat{h_l}(x_t)$
            Receive reward $r_{at} \in [0,1]$
        end for
    end for
```

3.1.2 EPSILON GREEDY ALGORITHM

As we realize that eager calculations are those calculations that choses the best from the at present accessible decisions, without considering the long haul impact of that choice which is picked around then which may happen to be an imperfect choice. The calculation dependably does not pick the best accessible alternative, rather than this it will haphazardly investigate different choices with likelihood = ϵ or pick the best choice with the likelihood = 1-ϵ. In this way, we can undoubtedly change the arbitrariness of the calculation by simply changing the estimation of ϵ. This procedure will assist us with exploring different choices all the more regularly. The estimation of ϵ is should be tuned by the test for example we can say that there is no single worth that will take a shot at all sort of investigation. Underneath figure demonstrates the working of the epsilon greedy algorithm.

Assume we have two options: A and B.

- Assume the probability of coming head = ϵ So, the probability of coming tails = 1- ϵ. Therefore.
  i) If head comes, explore available options randomly (exploration)
  ii) Probability = ½
  iii) If tails come, exploiting the best available option (exploiting)

By this if we have N options and the probability of selecting option randomly is ϵ 1/N; therefore, probability of other one is 1- ϵ i.e. selection the best option.

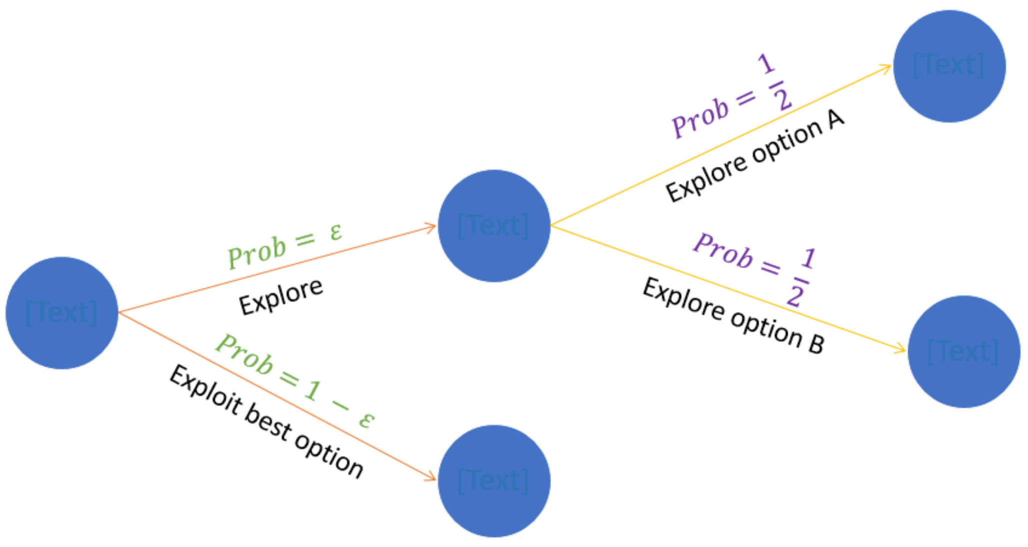


Figure 3.1 Epsilon Greedy Strategy

3.1.3 THOMPSON SAMPLING

Thompson Sampling is a Bayesian method. It will help us to handle the dilemma between the exploration and exploitation. While using Thomson sampling in an experiment, it will give us better results than upper confidence bounds [8]. To explore the different problem spaces, we used the Thompson sampling technique.

Thompson Sampling uses the estimated probability distribution for each arm, which is totally different from epsilon greedy technique which explore the nodes randomly.

At first Thompson Sampling exploring the arms at least once, which will increase the knowledge of the model and the probability distribution for each arm is constructed by this. By this it has its own configuration for bandits. To estimate the probability distribution, it uses the prior distribution P.

Consider this:

- Set of actions = a
- Rewards = r

In each example of setting, the calculation tests one case from each earlier dispersion and afterward settles on a decision among them, picking the best one. Thompson examining arbitrarily chooses an activity as per:

$$\int [\, E(r|a,\theta) = \max E(e|a',\theta)] P(\theta|D) d\theta \qquad (3.1)$$

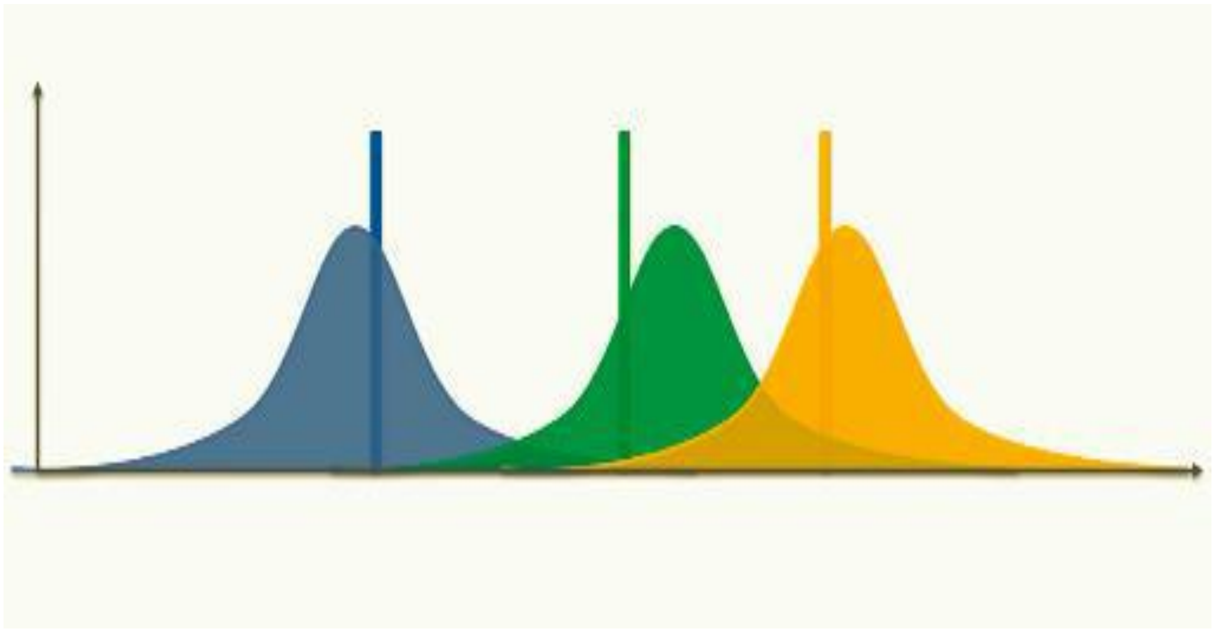


Figure 3.2 Thompson Sampling

Algorithm of Thomson Sampling

```
Define D = {}
For t= 1,………,T do
    Receive context $x_t$
    Draw $\theta t$ form posterior distribution P ($\theta|D$)
    Select arm $a_t$ = arg $\max_a$ E (r|$x_t$,a, $\theta t$)
    Receive reward $r_t$
    D= D U {$x_t$, $a_t$, $r_t$}
end for
```

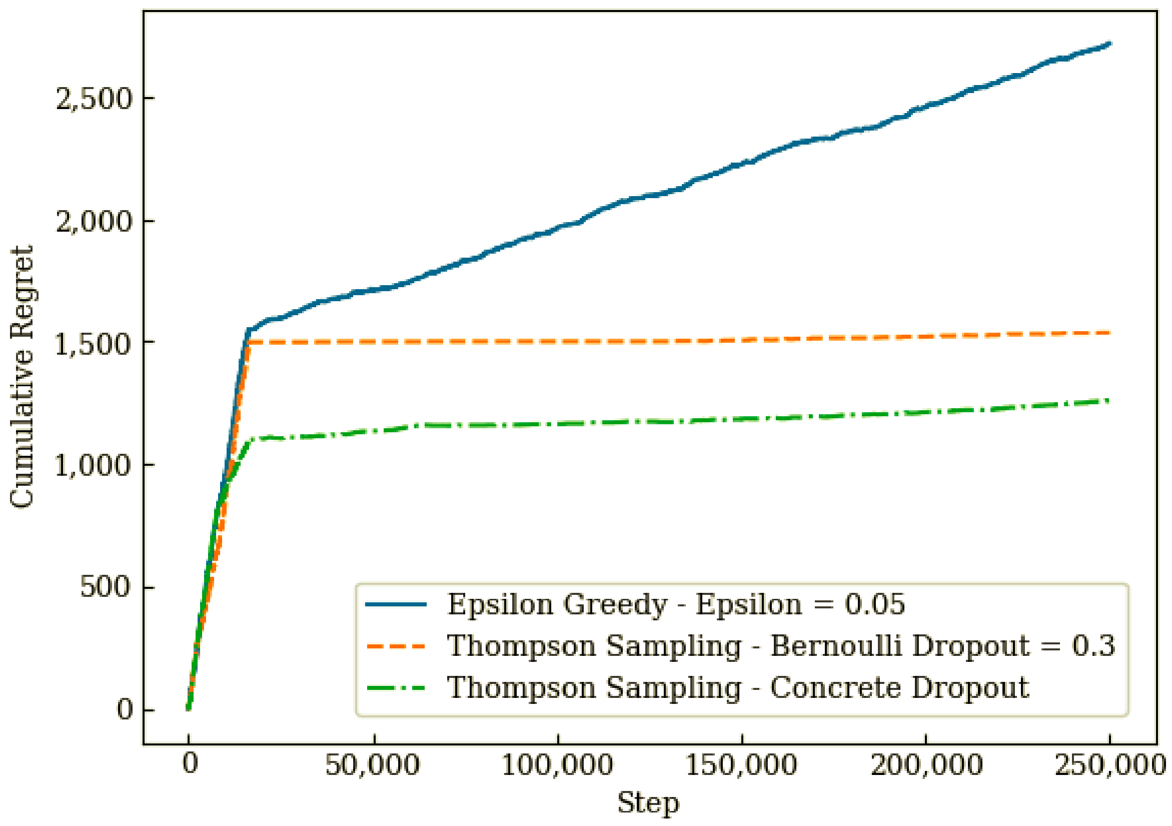


Figure 3.3 Graph Between Thompson Sampling and Epsilon Greedy

CHAPTER 4
WORK DONE

4.1 PROPOSED MODEL/ STRATEGY

The space of ANIFS [10][11][12] (Versatile Neural Fluffy Induction Framework) is fluffy framework. As we definitely know in CMAB (Multi-arm marauder) the framework predicts the best arm. Thus, in our purposed model we utilize the ANIFS model which is Versatile Fluffy model. We speak to each arm as an independent model in our purposed model. Thus, or we can say that as a reward the model will give us a score when the setting of the client is feed to the framework as info. To diminish the distinctive number of tuning parameters, we formulated a calculation which will assist us with training the model and these tuning parameters are difficulties in the past arrangements given by analysts. To acquire a fresh yield, the enrolment capacity was joined comparing to each arm. The yield will give us the general participation of setting highlights. The point by point model of the ANFIS model are examined underneath.

ANFIS has a place with class of versatile systems which are practically proportional to fluffy induction framework. It speaks to Takagi-Sugeno-Kang fluffy models (TSK) [10][12], likewise called as Sugeno Model clarified underneath.

As we combine the neural network and fuzzy logic just to develop the neuro-fuzzy system. There are two approaches which will be the base for these neuro-fuzzy systems.

- Neuro-Fuzzy Systems (Mamdani Approach)
- Neuro-Fuzzy Systems (Takagi & Sugeno’s Approach)

Neural networks can be combined with fuzzy logic in two different ways

1) **Neuro-Fuzzy systems (NFS):** A fuzzy Logic controller is represented using the structure of a neural network and trained it using either a back-propagation algorithm or a genetic algorithm or any other nature inspired optimization tool. The main purpose of developing the neuro-fuzzy system is to design and develop the fuzzy listening tool.
2) **Fuzzy Neural Network (FNN):** The neurons of the neural network have been designed using the concept of fuzzy set theory. There are three different ways to develop FNN.
   a) Real inputs but fuzzy weights

b) Fuzzy inputs but real weights
c) Fuzzy inputs and fuzzy weights

4.2 SUGENO MODEL

Accept a fuzzy derivation framework with x and y as info and z as yield. At that point a first-request Sugeno fuzzy model [49][50] has manages as the accompanying:

Rule 1:

If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_2$

Rule 2:

If x is A_2 and y is B_2 then $f_2 = p_2x + q_2y + r_2$

Where A1, A2, B1, B2 speaks to enrollment capacities. The parameters related with Ak, Bk enrollment capacities are called as predecessor parameters and parameter (pk, qk, rk) are called as subsequent parameters.

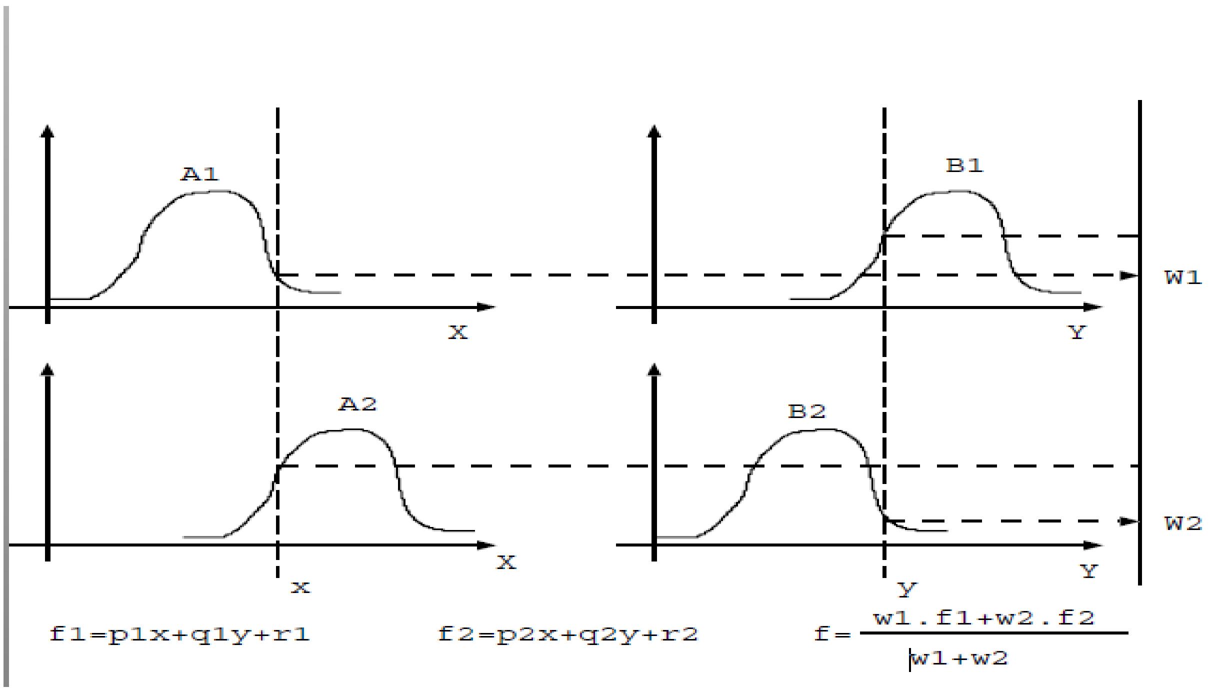


Figure 4.1: Sugeno Model

4.3 ANFIS ARCHITECTURE

As clarified above ANFIS [10] is a versatile system, which has five distinct layers and every single layer manage work which is extraordinary in nature. These capacities contribute in ascertaining the last score or the reward for the relating model. In neural system as we realize that there are loads, which are related with every hub except in ANFIS engineering, no loads are related with hubs. In this way, every hub takes the information and the parameters which are modifiable and related with specific hub as a capacity and ascertains its yield. Like fake neural system the yield of the past layer hub will be the contribution to the following layer hubs. As appeared in figure 10 underneath layer one and four are versatile layers which has parameters. No different layers with the exception of one and four hubs have parameters. The layers which have parameter masters during preparing process. Modifiable parameter

Since layer one and four have modifiable parameters, they are additionally called versatile layers and henceforth hubs having a place with these layers are called as versatile hubs. The parameters related with layer one is known as reason parameters while layer four parameters are called as subsequent parameters. The forward progression of sign in a system can be envisioned from the figure demonstrated as follows.

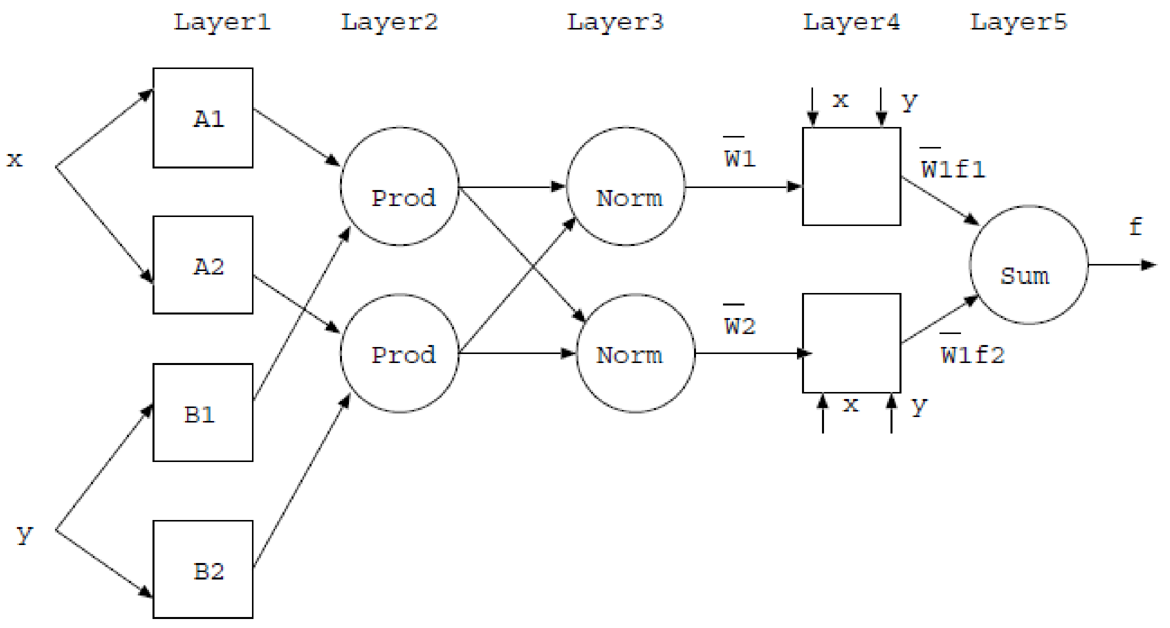


Figure 4.2: ANFIS Architecture

The model uses n*m:m:m:m:1 architecture, where n is the number of features of contextual vector and m represents the number of membership function associated with each feature in layer one. The node function of each layer and their significance are described above.

Input

Input (x,y) to the model is the feature vector representing the contextual features of the entity concerned.

Node Output

$O_{l,i}$ represents the output of i^{th} node of l^{th} layer.

Layer 1

In this layer crisp input is converted to value belonging to a fuzzy set. Hence, this layer is also known as the fuzzification layer. Each node is represented as a square representing an adaptive node. Each feature of contextual vector input is associated with Gaussian membership function in our proposed model, so there are $n * m$ nodes in this layer. Node function for each node is a Gaussian function as following:

$$O_{1,i,j} = \mu_{i,j} = e^{-\frac{(x_i - a_{i,j})^2}{2b_{i,j}^2}} \qquad (4.1)$$

where $1 <= i <= n$, $1 <= j <= m$ and $(a_{i,j}), (b_{i,j})$, and represents set of parameters associated with each node and are modified as per the error and thus accordingly linguistic value of each membership function for each feature are generated. Parameters belonging to this layer are called as premise parameters.

Layer 2

It multiplies the incoming signals and forwards the same to the next layer. Hence, also known as prod layer. Node is represented as a circle, implying there are no modifiable parameters associated with them. There are nodes in this layer as we have chosen membership function associated with each feature calculating the linguistic value of each feature in corresponding membership function representing the factor of m rules. Each node function can be described by following the node equation:

$$O_{2,j} = w_j = \prod_{i=1}^{n} O_{1,i,j} = \prod_{i=1}^{n} \mu_{i,j} \qquad (4.2)$$

Where $1 <= j <= m$. Each node output represents the firing strength of each rule.

Layer 3

It computes the normalized strength of each rule, hence called as normalization layer. Similar to layer 2, nodes are also represented as circle. There are nodes in this layer which are normalized firing strength of each rule. Each node can be described by following equation:

$$O_{3,j} = w'_j = \frac{O_{2,j}}{\sum_{k=1}^{3} O_{2,k}} = \frac{w_j}{w_1 + w_2 + w_3} \qquad (4.3)$$

Where $1 <= j <= m$.

Layer 4

Also known as defuzzification layer as it produces crisp output value y, resulting from inference of rules. Each node calculates inference of normalized firing strength of each rule and first order polynomial of input contextual vector. The nodes in this layer are represented by square, implying the presence of modifiable parameters which are the coefficients of first order polynomial. There are 3 nodes in this layer. Node equation can be given as following:

$$O_{4,j} = y_i = O_{3,j} * f_j = w'_j * (\textstyle\sum_{k=0}^{n} q_{j,k} * x_k) \tag{4.4}$$

Where $1 <= j <= m$ and $x_0 = 1$. $(x_1, x_2, \cdots, x_n)$ and n. represents features of contextual

input vector. The set of $(q_{j,0}, q_{j,1}, \cdots, q_{j,n})$ represents node parameters and are known as consequent parameters.

Layer 5

It is also known as output layer consist of single node. The node computes summation of output of nodes belonging to fourth layer. The single node generates the Score of corresponding models. The node equation can be represented as:

$$O_{5,1} = z = \textstyle\sum_{j=1}^{m} O_{4,j} = y_1 + y_2 \ldots\ldots + y_m \tag{4.5}$$

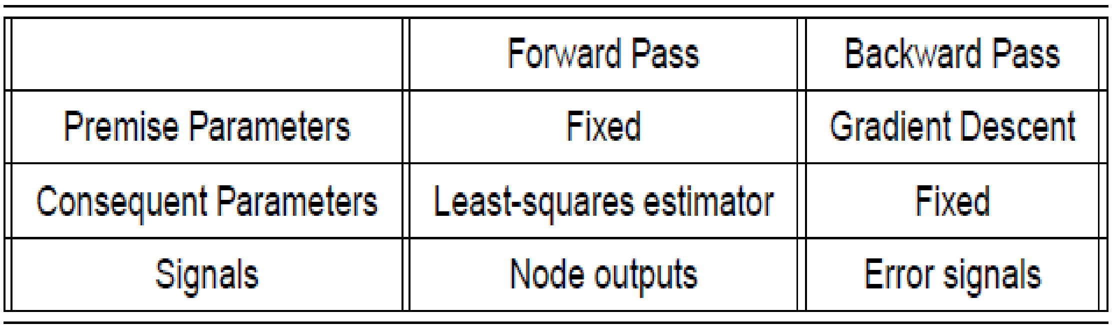
Output Significance

Output signifies membership score of the input context for corresponding arm implemented as an ANFIS model.”

4.4 HYBRID LEARNING ALGORITHM

"The ANFIS [40][31][38][47] can be trained by a hybrid learning algorithm, consisting of two passes. The forward pass uses least-squares (LSE) method to identify the consequent parameters belonging to layer 4 keeping premise parameter fixed.

The backward pass [57] propagated the errors backward and update the premise parameters by gradient descent method keeping consequent parameters fixed."



	Forward Pass	Backward Pass
Premise Parameters	Fixed	Gradient Descent
Consequent Parameters	Least-squares estimator	Fixed
Signals	Node outputs	Error signals

Table 4.1: Hybrid Learning Algorithm for ANFIS

"4.5 LEARNING RULE DEFINITION

Assume an adaptive network with L layers with k^{th} layer having $\#(k)$ nodes.

Let us denote the i^{th} node of k^{th} layer by (k, i) and node function as O_i^k.

Since incoming signals at node and parameters determine node output. Thus, we have

$$O_i^k = O_i^k(O_i^{k-1}, \ldots\ldots, O_{\#(k-1)}^{k-1}, a, b, c) \tag{4.6}$$

Here O_i^k represents both node output and node function."

"4.6 ERROR MEASURE

Assume we have a data set for training with P records. Then error for p^{th} record can be computed as sum of squared error as:

$$E_p = \sum_{m=1}^{\#(L)} (T_{m,p} - O_{m,p}^L)^2 \tag{4.7}$$

$T_{m,p}$ is the m^{th} component of p^{th} record.

$O_{m,p}^L$ is the m^{th} component of actual vector. Then overall error is calculated as

$$E = \sum_{p=1}^{P} E_p \tag{4.8}$$

4.6.1 Error Rate for each Output

We calculate the error rate $\frac{\partial E}{\partial O}$ for the p^{th} training data for each node output. The error rate for the output note at

$$\frac{\partial E_p}{\partial O_{i,p}^L} = -2(T_{i,p} - O_{i,p}^L) \tag{4.9}$$

For the internal node at, we can use chain rule to derive the error rate as:

$$\frac{\partial E_p}{\partial O_{i,p}^k} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{m,p}^{k+1}} \frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k}, \tag{4.10}$$

Where $1 \leq k \leq L - 1$.

“4.6.2 Error Rate for each Premise Parameter

Let represent premise parameter. Then

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \epsilon\, S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \alpha} \qquad (4.11)$$

where S is the set of nodes whose output depends on α.

Then *derivative of the overall error* with respect to α is,

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \alpha} \qquad (4.12)$$

Thus, we can update α using following formula: $\Delta\alpha$

$$\Delta\alpha = n\, \frac{\partial E}{\partial \alpha} \qquad (4.13)$$

4.6.3 Learning Paradigms

For updating premise parameters after each record (online training), we have following formula for update:

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \alpha} \qquad (4.14)$$

For batch learning (off-line learning) the update formula reduces to the derivative of the overall error with respect to α:

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \alpha} \qquad (4.15)$$

4.7 ALGORITHM FOR ANFIS

For each characteristic vector x, consisting of n features do:

Forward pass

Calculate layer 1:

$$O^1_{m,i} = \mu_{m,i}(x_i)$$

$1 <= i <= n$, $1 <= m <=$ number of rules

Calculate layer 2:

$$O^2_m = \Pi_i \, \mu_{m,i}$$

Calculate layer 3:

$$O^3_m = O^2_m \big/ \textstyle\sum_m O^2_m$$

Least Square Estimate, calculate consequent parameters keeping premise parameters fixed.

Calculate layer 4:

$$O^4\, m \; = \; O^3\, m\, (Fm\, x)$$

Calculate layer 5:

$$O^5{}_1 = \prod_m O^4{}_m$$

Backward pass

Using backpropagation algorithm update premise parameters, keeping the consequent parameters fixed.

4.8 ALGORITHM FOR PER-ARM ANFIS

```
1. Construct anfis model for each arm.
2. Randomly initialize the membership function for each model.
3. For each data do:
    3.1. Predict output from each model.
    3.2. Choose the arm with maximum output value.
    3.3. If calculated chosen arm is the actually chosen arm, assign reward = 1
    else reward = 0.
    3.4. Update the chosen arm in step 3.2 with the reward obtained in step 3.3.
```

4.8 EXPERIMENTAL RESULTS FOR PER ARM ANFIS

The experiment performed on the Per Arm Anfis algorithm after successful implementation was conducted on different datasets, viz., Covertype, Mushroom, Adult Income and Statlog.

The experiments were conducted on two different exploration values of 10% and 5% for the Epsilon-Greedy exploration technique setup with the Per Arm Anfis model. The results of the experiment are as follows:”

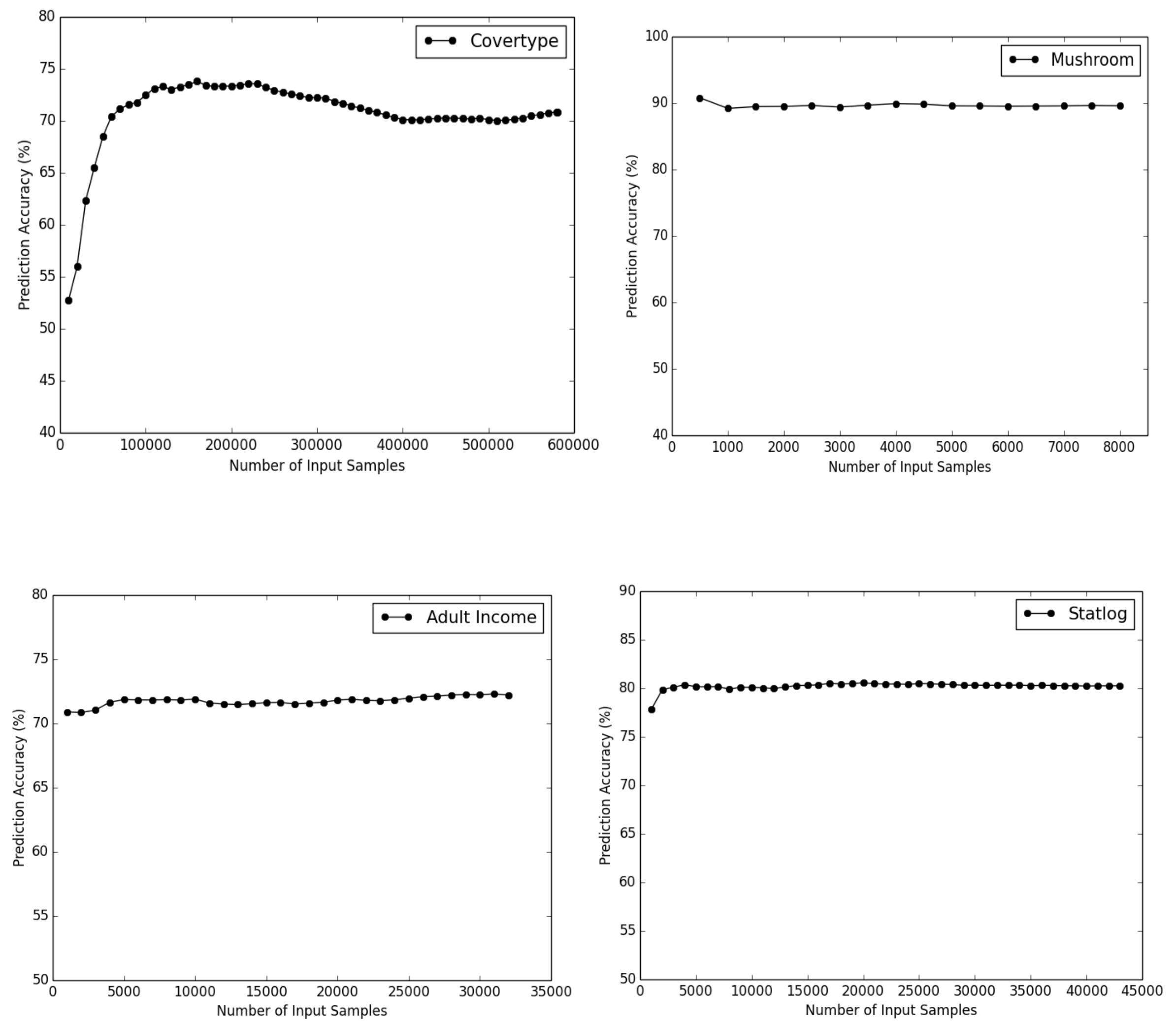
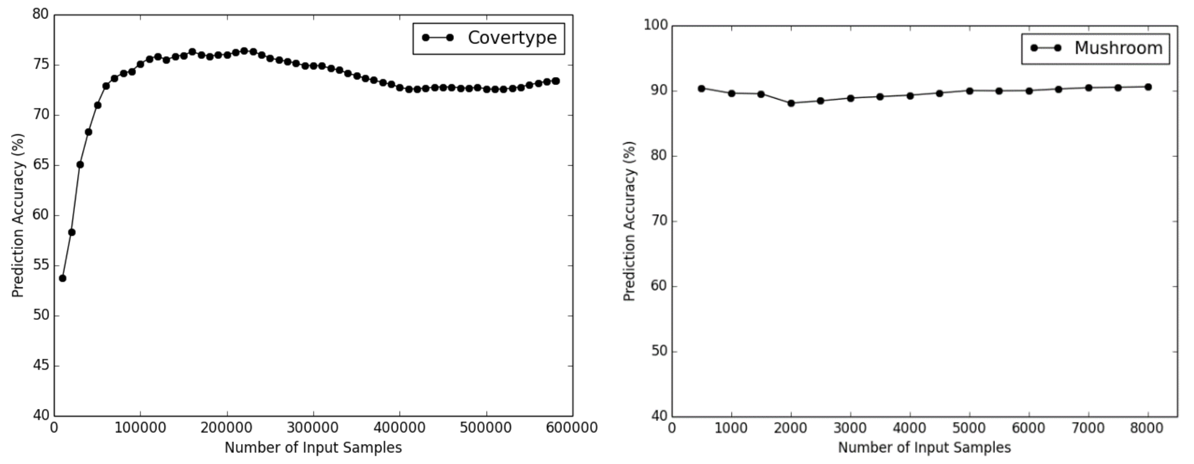


Figure 4.3: 10% Exploration of ANFIS model

“The above figure shows the number of input samples vs. prediction accuracy graphs for the proposed Anfis model for 10% exploration value.



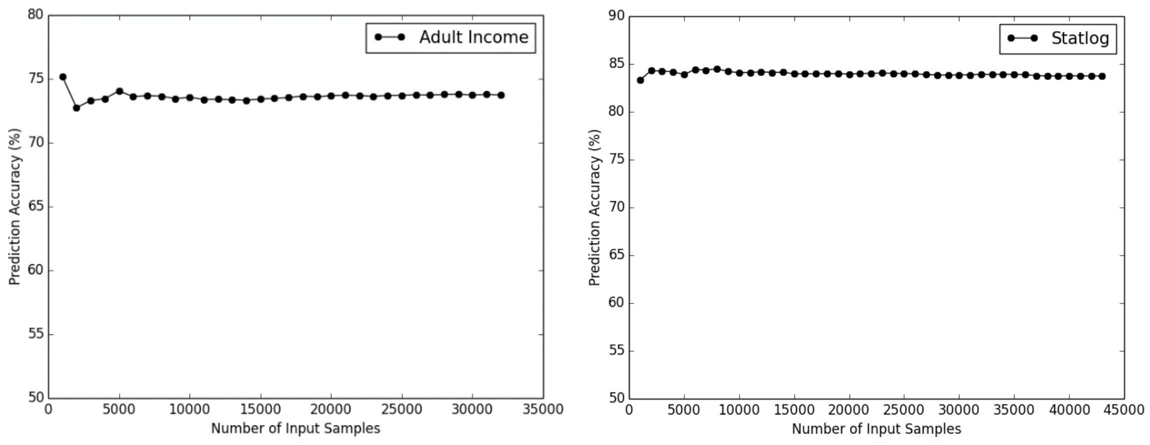


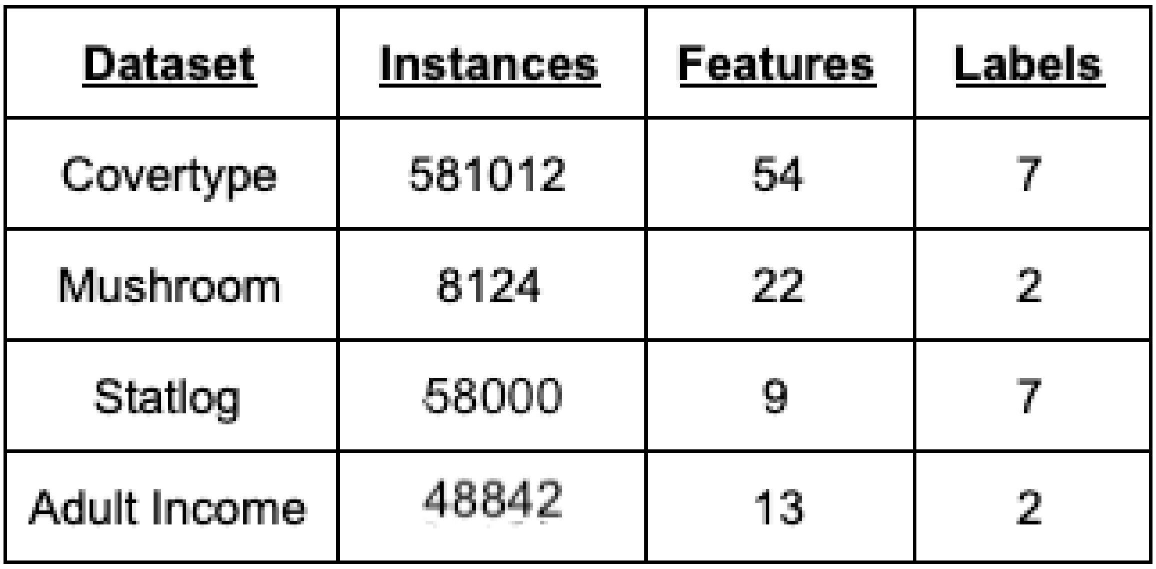
Figure 4.4: 5% Exploration of ANFIS model

The figure represents the Number of input samples vs. prediction accuracy graphs for the per arm anfis model for 5% exploration value of epsilon greedy technique.”

CHAPTER 5
EXPERIMENTS

5.1 EXPERIMENTAL DESCRIPTION

The different number of datasets are used as an experiment in the purposed model which is Contextual Multi-Armed Bandit Algorithm named as Adaptive Neuro Fuzzy Inference System (ANFIS). UCI which is a machine learning repository provided the datasets related to these kinds of problems i.e. these datasets comes under the category of CMAB problem. The elaborated form of dataset shown in the **table** below on which the experiment is conducted. The contextual feature and a labelled class are associated with each dataset, which used in the multi-arm bandit as an arm. Our algorithm should be capable to predict the class in the testing phase.



Dataset	Instances	Features	Labels
Covertype	581012	54	7
Mushroom	8124	22	2
Statlog	58000	9	7
Adult Income	48842	13	2

Table 5.1: Dataset table

Following explains the different datasets which are used the experiment.

- **FOREST COVERTYPE DATASET:**
  There are around 600000 rows with 54 different attributes. These attributes/ features columns are used to predict the label / class. The last column of the Covertype dataset is the class of that particular instance. It has 7 different type of classes numbered from 1 to 7 i.e. These 7 different values show that it has 7 different type of classes. During testing multi-arm bandit algorithm should be able to classify the correct label of the instance of the data using the remaining features of the particular instance. The nature of characteristics of the dataset is multivariant. As explained above there are 54 attributes / features are present in the forest covertype dataset which are characterized in the 12 different type of measures. Out of those 54 attributes 10 are quantitative variables, 4 are wilderness areas which are binary in nature and the remaining 40 are soil type variables which are also binary in nature. There is no need to pre-process the data because the dataset does not have any missing value. So, our algorithm should be robust to handle the missing data if there is any missing value in the dataset. The values in the dataset is quantitative in nature as an experimental purpose. so, there is no need to pre-process the data.
  This dataset is mostly used in the Contextual Multi-Armed Bandit Problem.

- **MUSHROOM DATASET:**
  In this dataset 22 attributes / features are represented as columns and around 9000 instances of those attributes are recorded
- **STATLOG DATASET**
- **ADULT INCOME DATASET**

There are different algorithms named Random Policy, Neural Bandits with and without dropout Confidit etc which are used to compare the ANFIS model. The dataset which is pre-processed feed to the models and compare the results of the different models with our purposed model.

The basics of these algorithms given below.

- ➢ **RANDOM POLICY:**
  In this process one instance of the input is processed, after that others will be processed i.e. algorithm does not know anything about the whole data form the beginning. Due to this it is known as online algorithm. It chooses

the arms randomly in each iteration. It totally ignores the context vector of the input attributes.

- **NEURAL BANDIT MODEL:**
  Each arm of in this model is models as a neural model. To deal with the exploration it uses the epsilon greedy technique. "Each arm of the Multi-Armed Bandit Problem is made as a neural model. The
  context vector is fed to each arm model which then generates a score for that particular arm. The one arm that gives the best score is predicted and is the output of the overall Neural Bandit Model. Also, it uses the epsilon-greedy approach for exploration which gives an epsilon time, the chance to explore the unexplored arms. If the predicted arm is correct than a reward of 1 else reward 0 is fed to the particular selected arm model to learn."

- **BANDITRON:**
  It uses the perceptron model, which takes the input vector and give reward as labelled arm for that vector (input instance). Epsilon Greedy algorithm is used in this algorithm for exploration. The model learns form the given rewards and the rewards are given only when algorithm predicts the correct label of the class. To explore the unexplored arm, it uses the epsilon greedy technique. This is also an online algorithm.
- **CONFIDIT:**
  This model is better than above discussed model named Banditron. There are many improvements in this model. As explained above for exploration in each iteration it uses the upper confidence bound. The arm which played a greater number of times, its confidence bound is very small and vice-versa. For any particular arm this model computes the upper confidence bound in each iteration. From the smaller value of confidence bound we can conclude that the prediction of this model is correct.

"All these existing models are compared with our proposed model namely, ANFIS (Adaptive Neuro-Fuzzy Inference System) to get the experimental performance measure for each model. The ANFIS model is implemented as a Per Arm Anfis model which has n Anfis model for each arm of the multi-armed bandit setting. It is also an Online Model which does not have any knowledge about the complete data and just processes each instance piece by piece. Each instance of the dataset is fed to each arm of the Per Arm Anfis model which finds which arm gives the highest score for that instance and thus predicts the highest score arm. If the predicted arm is correct, a reward 1 is given else a reward 0 is given to the predicted arm (Anfis model) which then modifies its parameters (i.e. the membership values) according to the achieved reward. The Anfis model uses the Epsilon-Greedy exploration approach to predict the arm to be chosen from the set of all arms, giving each arm an equal opportunity an epsilon number of times.

The cumulative reward is calculated after each instance is evaluated. A cumulative reward is the total reward achieved so far by the model is calculated by adding the reward for each instance of the dataset. The Accuracy at each step is calculated by dividing the cumulative reward achieved until that step by the total number of instances evaluated.

For better comparison, the experiments each model is conducted on different exploration rate viz., 10% and 5% for all the datasets."

5.2 TUNABLE PARAMETER EXPERIMENT: K-MEANS ALGORITHM

The Anfis Model uses a tunable parameter which portrays the quantity of enrollment rules for each element, where more than one participation capacity suits the likelihood of more than one semantic variable related with an element. This prompts a tuneable parameter to be chosen according to the dataset.

Subsequently, an analysis was led to discover the advanced number of principles, i.e., the estimation of m, for each dataset. We tested this issue by utilizing the K-Means elbow technique to locate the best number of enrollment rules for each dataset.

The investigation was directed by running the Element K-Means Calculation for each dataset.

ALGORITHM

```
Begin
sum_of_square_dist = [] //Empty list
for each k in [2,10]
    dist = 0.0
    for each feature in dataset
        Use k-Means to find k optimal centers
        dist += Calculate distance of each $x_f$ with k optimal centers
    sum_of_square_dist.append(dist)
Plot (k,sum_of_square_dist).
Elbow in the plot relates to the optimal value for $m$.
```

EXPERIMENT CONCLUSION

In spite of the fact that Investigation directed gives an understanding for finding an ideal incentive for however because of the idea of the test, there is equivocalness in choosing the elbow. Subsequently, the analysis can't effectively foresee the definite estimation of yet it gives some affectation focuses to be considered as potential qualities for.

The consequences of the elbow test and the comparing exploratory outcomes for various estimations of directed are as per the following:

COVERTYPE DATASET

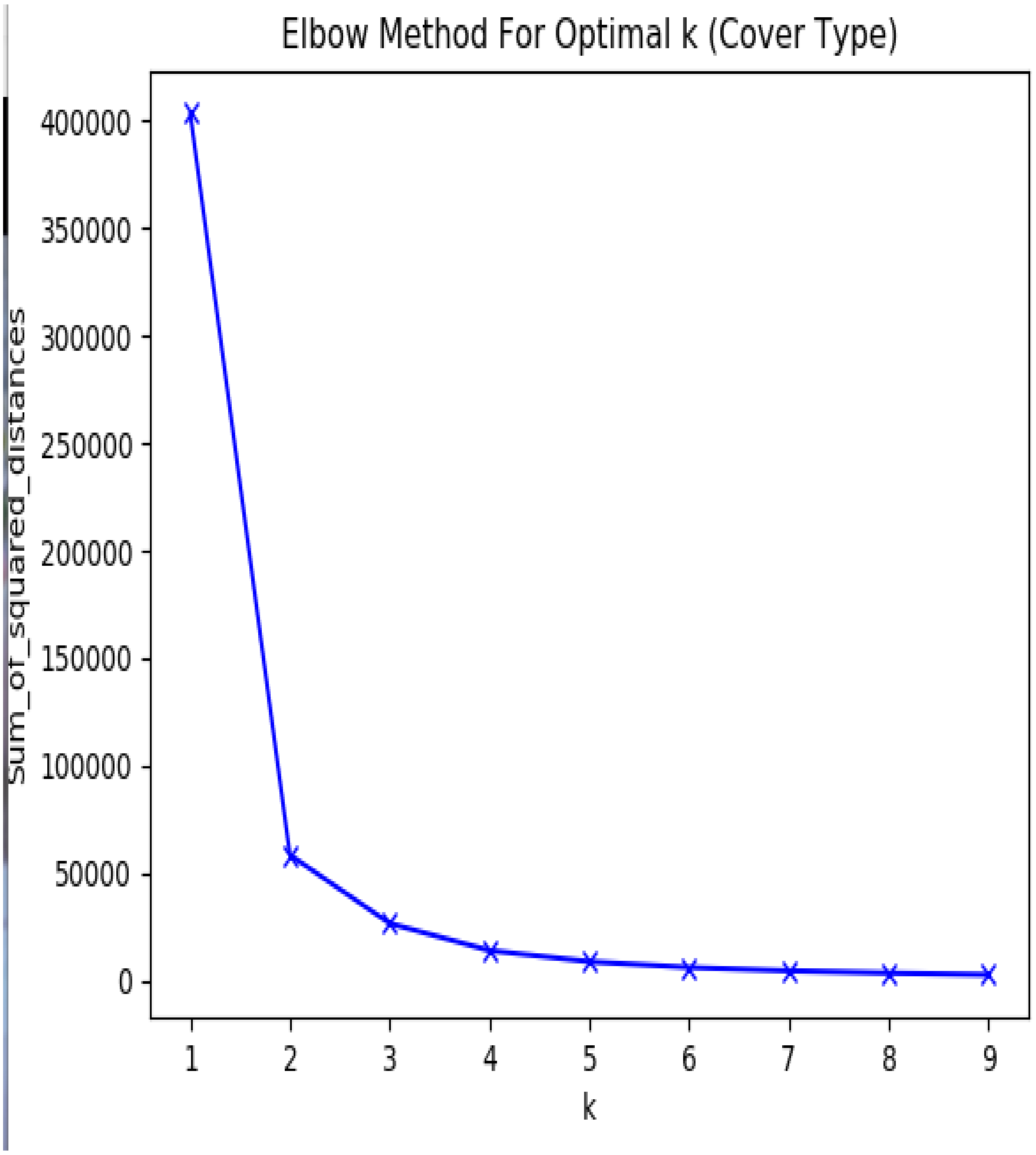


Figure 5.1: Elbow Method Result (Covertype)

The result of the above-mentioned algorithm via the elbow test for the Covertype dataset suggest that the most suitable value of m is 3.

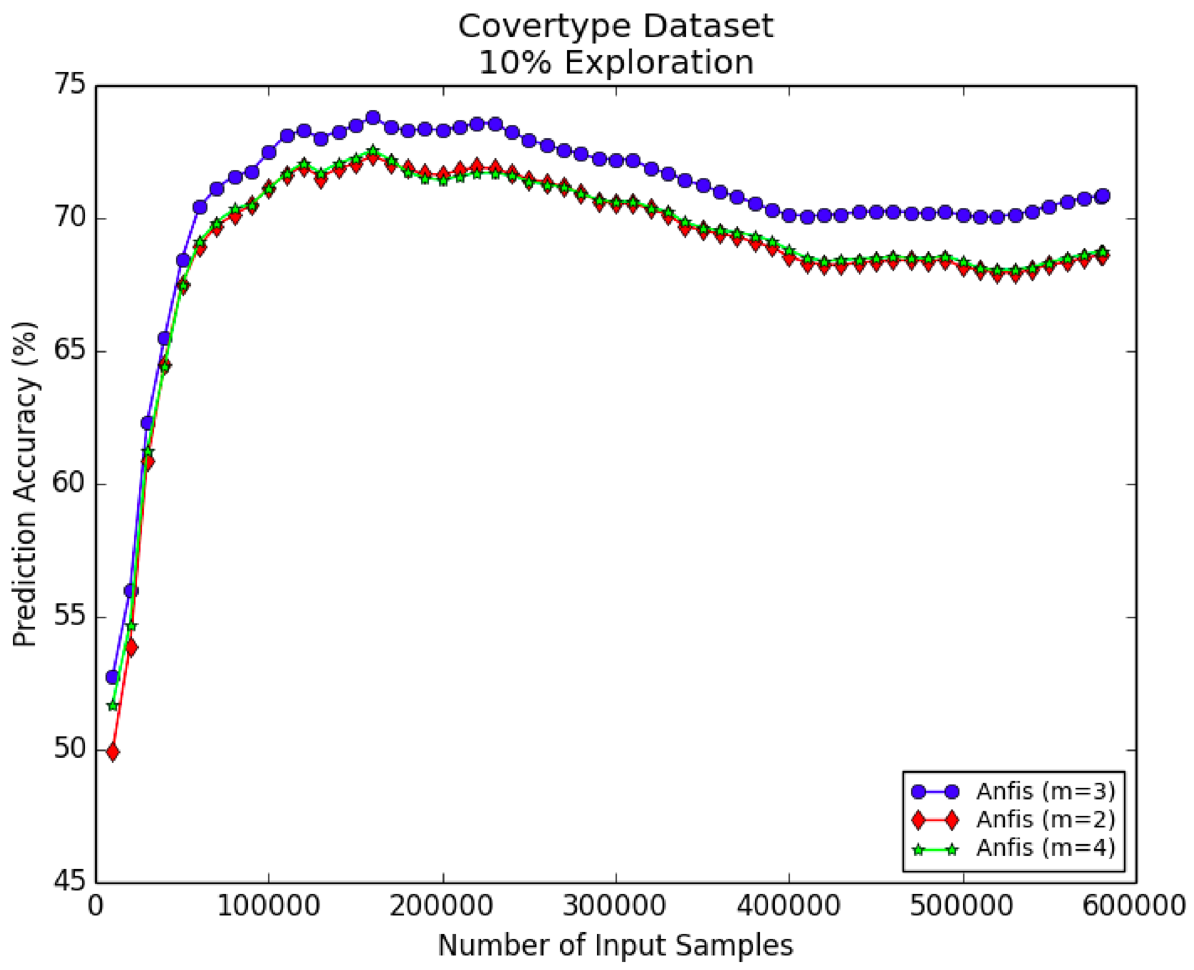


Figure 5.2: Experimental Results (Covertype)

The results of the Anfis algorithm on the different values of after performing the experiment as shown in the above graph specifies the most suitable value of m as 3.

MUSHROOM DATASET

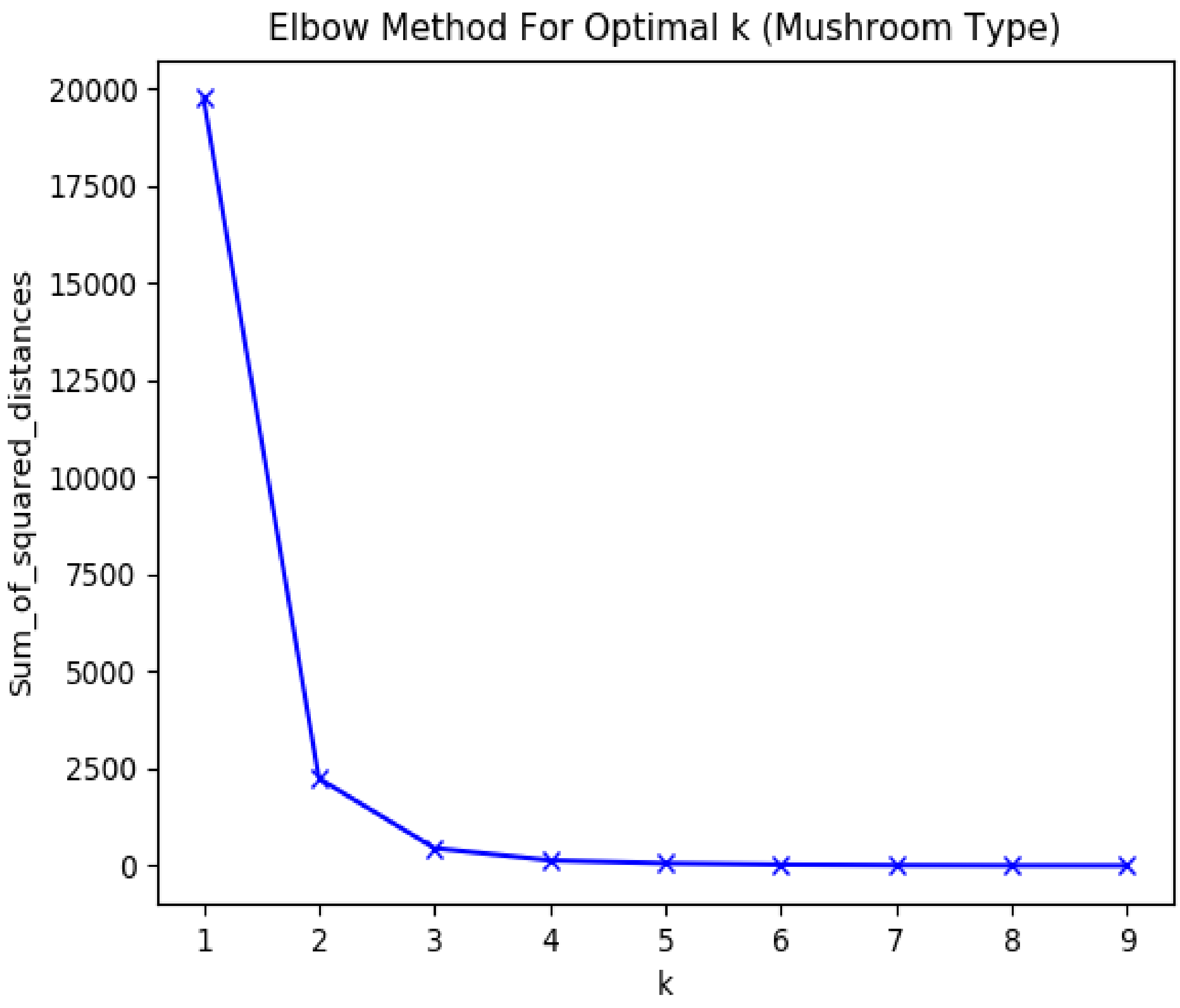


Figure 5.3: Elbow Method Result (Mushroom)

The result of the Tunable Parameter Experiment via the elbow test for the Mushroom dataset suggest that the most suitable value of m is 2.

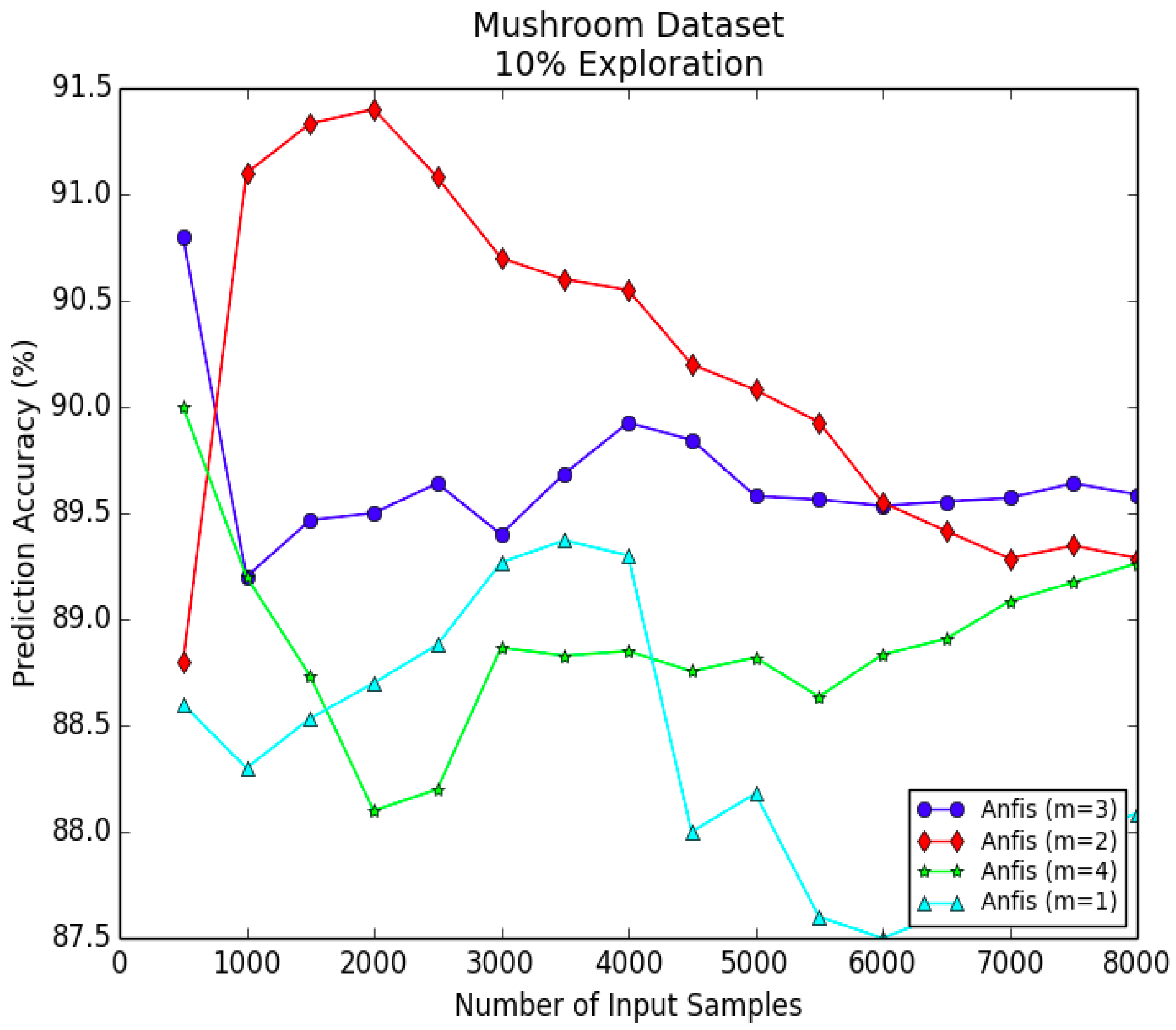


Figure 5.4 Experimental Result (Mashroom)

“The result of the Anfis algorithm for the different values of m after performing the experiment as shown in the above graph state that the most suitable value of m for the Mushroom dataset is 3.”

CHAPTER 6
RESULTS

“We presented another logical marauder calculation known as Versatile Fluffy Model (Anfis). Here we report the test results acquired on the distinctive datasets utilized. The investigations are completed in a web-based setting where each case of the information is bolstered one by one and the model has no clue about the all-out dataset.

Every one of the figures demonstrate the pattern of total reward accomplished as for the quantity of assessed tests. The combined reward is the aggregate of remunerations accomplished for various assessed tests. The figures demonstrate the combined prizes accomplished as Exactness as determined by isolating the aggregate reward by the all-out number of assessed tests.

The figures demonstrate the Exactness versus Number of Info Tests diagrams to get the similar thought regarding the working of the current calculations and the proposed Anfis algorithmic model in the zone of CMAB Issue.

Numerous papers which manage the CMAB Issue utilize Combined lament as an assessment metric. It is the misfortune caused when we pick the off-base arm. In any case, we pick aggregate rewards as our assessment metric to think about different models.

We utilize a pattern Arbitrary Arrangement that does not think about the unique circumstance and picks an irregular arm for each information test. Our Model accomplishes a more prominent combined reward than Confidit, Neural Criminals. Analyses were led with 10% investigation and 5% investigation in the Epsilon-Ravenous Investigation procedure.

Results are as per the following:"

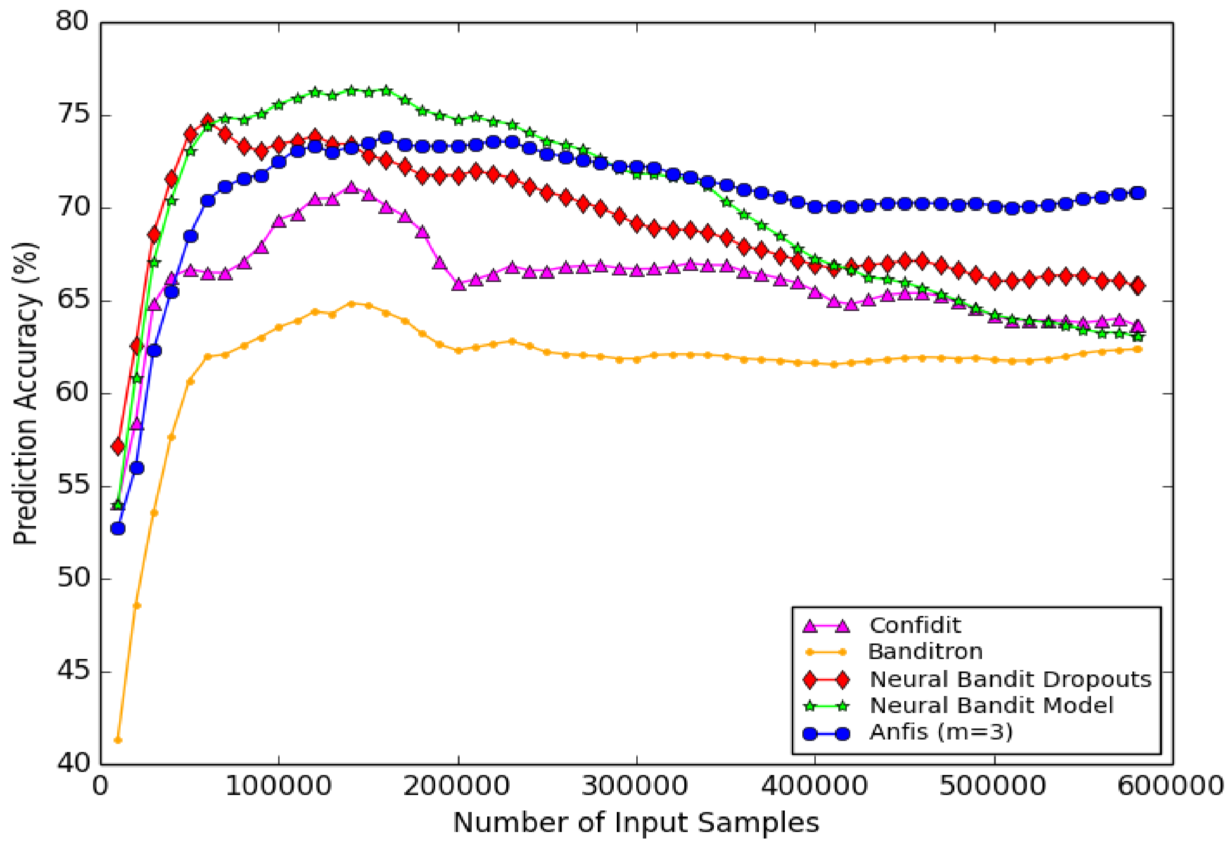


Figure 6.1 Comparison Result of Covertype – 10% Exploration

The outcome demonstrates the Quantity of Information Tests versus the Forecast Exactness chart. This chart speaks to the similar exactness forecast for the various models. The outcome is determined for the Covertype Dataset on 10% investigation esteem. Each bend achieves a pinnacle and afterward start to fall as the investigation advances. The Versatile Fluffy Model (ANFIS) adjusts better to the contribution when contrasted with others and smoothes out as the examination advances. Our model accomplishes a superior outcome when contrasted with the Banditron and Confidit model by a more noteworthy degree.

This outcome above demonstrates that the ANFIS model predicts the best in examination with every single other model. The general precision accomplished by the Banditron model is 62-63% and for the confidit model is in the scope of 63-64%. The Neural model accomplishes a general exactness of 63% and the Neural Model utilizing Dropouts accomplishes around 65-66% precision. The ANFIS model beats every one of the models and gives a general precision of 70-71%.

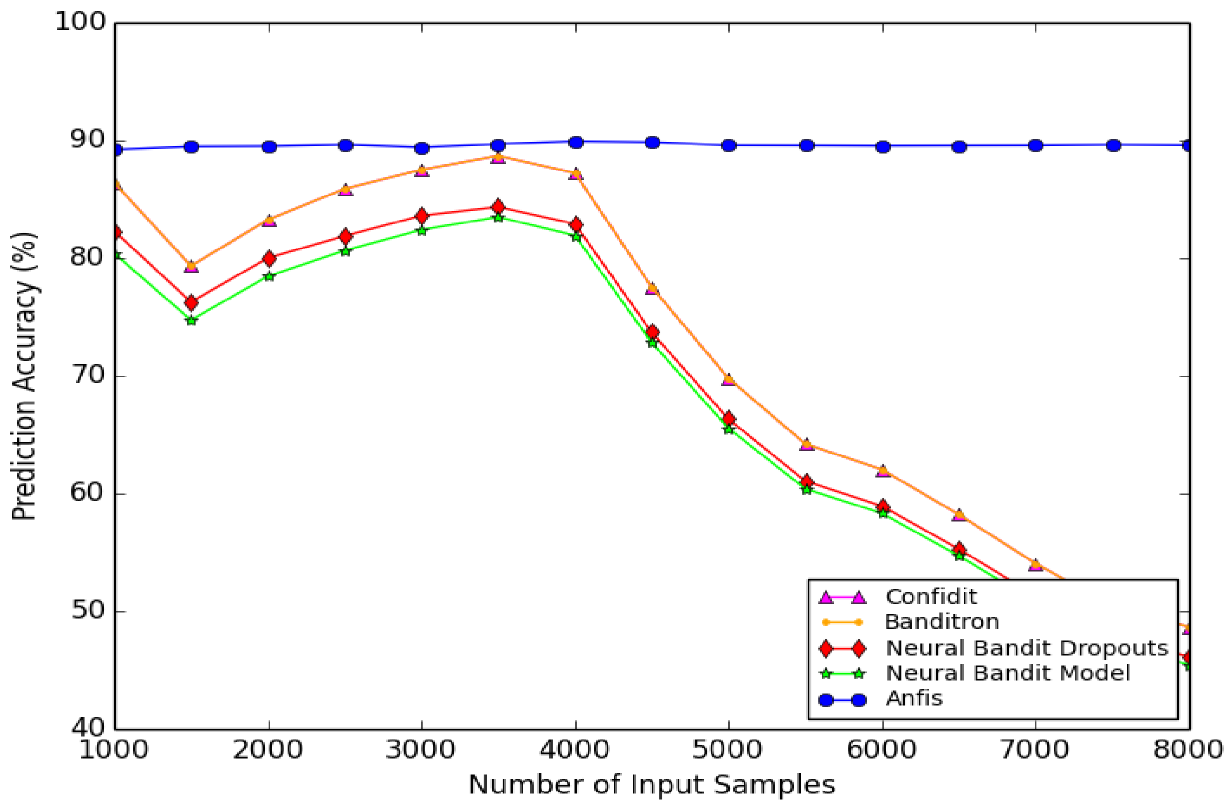


Figure 6.2: Comparison Result of CoverType- 5% Exploration

The figure demonstrates the Quantity of information tests versus forecast exactness diagram. The chart speaks to the examination between the distinctive existing calculations and the proposed model based on forecast of precision. The analysis is led on Mushroom Dataset with an investigation estimation of 10%. It tends to be effectively construed from the assume that the bends for every one of the calculations including Neural Highwayman and Confidit achieve a pinnacle an incentive in the beginning of the test and after that drop definitely in the rest of the piece of the examination. Notwithstanding, the Versatile Fluffy model (ANFIS) adjusts better to the information and demonstrates no lofty drop in precision as the analysis advances. The bend for Anfis smoothes out when contrasted with the bends of different calculations.

The outcome above demonstrates that the ANFIS plays out the best in contrast with different calculations. The Banditron model gives a general precision of 48-49%. The Confidit model predicts with around a similar precision. The Neural

Scoundrel model performs more awful than the confidit model by giving an exactness of 45-46%. The Neural Desperado utilizing dropout plays out somewhat better by giving a precision of 50-51%. The ANFIS model beats everybody by giving 88-89% forecast exactness.

CHAPTER 7
REFERENCES

[1] K. J. Astrom and B. Wittenmark, Computer Controller Systems: Theory and Design. Prentice-Hall, 1984.

[2] S. M. Botros and C. G. Atkeson, "Generalization properties of radial basis functions," in Advances in Neural Information Processing Systems III, D. S. Touretzky, Ed. San Mateo, CA Morgan Kaufmann, 1991

[3] S. Cben, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," IEEE Trans. NeuralNetworks, vol. 2, no. 2, pp. 302-309, Mar. 1991.

[4] D. Dubois and H. Prade, Fuzzy Sets and Systems: Theory and Applications. New York: Academic, 1980.

[5] S. E. Fahlman, "Faster-learning variations on back-propagation: an empirical study," in Proc. 1988 Connectionist Models Summer School, D. Touretzky, G. Hinton, and T. Sejnowski, Eds., Camegie Mellon Univ., 1988, pp. 38-51.

[6] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in Advances in Neural Information Processing Systems II, D. S. Touretzky, G. Hinton, and T. Sejnowski, Eds. San Mateo, CA Morgan Kaufmann, 1990.

[7] G. C. Goodwin and K. S. Sin. Adaptive Filtering Prediction and Control. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[8] S. S. Haykin, Adaptive Filter Theory. Englewood Cliffs, NJ: Prentice Hall, second ed., 1991.

[9] W. T. Miller 111, R. S. Sutton, and P. J. Werbos, Eds., Neural Networks for Control. Cambridge, MA: MIT Press, 1990.

[10] J.-S. Roger Jang, "Fuzzy modeling using generalized neural networks and Kalman filter algorithm," in Proc. Ninth Nat. Conj Artificial Intell.

[11] "Rule extraction using generalized neural networks," in Proc. 4th IFSA World Congress, July 1991.

[12] "Self-learning fuzzy controller based on temporal backpropagation," IEEE Trans. Neural Networks, Sept. 1992.

[13] J.-S. Roger Jang and C.-T. Sun, “Functional equivalence between radial basis function networks and fuzzy inference systems,” IEEE Trans. Neural Networks, vol. 4, pp. 156-159, Jan. 1993.

[14] R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, K. Lee, and P. S. Lewis, “Function approximation and time series prediction with neural networks,” in Proc. IEEE In?. Joint Con$ Neural Networks, 1990, pp. 1-649-665.

[15] V. Kadirkamanathan, M. Niranjan, and F. Fallside, “Sequential adaptation of radial basis function neural networks,” in Advances in Neural Information Processing Systems Ill, D. S. Touretzky, Ed. San Mateo, CA Morgan Kaufmann, 1991, pp. 721-727..

[16] A. Kandel. Fuzzy Expert Systems. Reading, MA Addison-Wesley, 1988.

[17] A. Kandel, Fuuy Expert Systems. Boca Raton, FL CRC Press, 1992.

[18] L. V. Kantorovich and G. P. Akilov, FunctionalAnalysis, second edition. Oxford, UK Pergamon, 1982.

[19] M. S. Klassen and Y.-H. Pao. “Characteristics of the functional-link net: A higher order delta rule net,” In ZEEE Proc. Int. Conj Neural Networks, San Diego, June 1988.

[20] T. Kondo, “Revised GMDH algorithm estimating degree of the complete polynomial,” Trans. SOC. Instrument and Contr. Engineers, vol. 22, no. 9, pp. 928-934, 1986 (in Japanese).

[21] B. Kosko, Neural networks for signal processing. Englewood Ciffs, NJ: Prentice Hall, 1991.