DECENTRALIZED INTRUSION PREVENTION AGAINST CO-ORDINATED CYBERATTACKS ON DISTRIBUTION AUTOMATION SYSTEMS

A Project report submitted in

Partial fulfillment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED

By

B.PRAVALLIKA	**17671A0558**
B.BHARGAVI	**17671A0563**
N.ESHWITHA	**17671A0568**
RITIKA KALYANI	**17671A0596**

Under the esteemed guidance of

DR. R. VIJAYANAND

ASSISTANT PROFESSOR

Department of Computer Science and Engineering

J.B. Institute of Engineering and Technology

(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated to JNTUH)

Yenkapally, Moinabad mandal, R.R. Dist-75 (TG)

2017-2021

J.B.INSTITUTE OF ENGINEERING AND TECHNOLOGY

(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated to JNTUH)

Yenkapally, Moinabad Mandal, R.R. Dist. -500 075

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "DECENTRALISED INTRUSION PREVENTION AGAINST CO-ORDINATED CYBERATTACKS ON DISTRIBUTION AUTOMATION SYSTEMS" submitted to the Department of Computer Science and Engineering, J.B Institute of Engineering and Technology, in accordance with Jawaharlal Nehru Technological University regulations as partial fulfillment required for successful completion of Bachelor of Technology is a record of bonafide work carried out during the academic year 2020-21 by,

B.PRAVALLIKA	**17671A0558**
B.BHARGAVI	**17671A0563**
N.ESHWITHA	**17671A0568**
RITIKA KALYANI	**17671A0596**

Internal Guide

DR. R. VIJAYANAND

ASSISTANT PROFESSOR

Head of the Department

DR. P. SRINIVASA RAO

PROFESSOR

External Examiner

J.B.INSTITUTE OF ENGINEERING AND TECHNOLOGY

(UGC Autonomous)

(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated to JNTUH)

Yenkapally, Moinabad Mandal, R.R. Dist. -500 075

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby certify that the Main Project report entitled **"DECENTRALISED INTRUSION PREVENTION SYSTEM AGAINST CO-ORDINATED CYBERATTACKS ON DISTRIBUTION AUTOMATION SYSTEMS"** carried out under the guidance of**, DR. R. VIJAYANAND , Assistant Professor in computer science and engineering** is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **computer science and engineering.** This is a record of bonafide work carried out by us and the results embodied in this project report have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Date:

B.PRAVALLIKA	**17671A0558**
B.BHARGAVI	**17671A0563**
N.ESHWITHA	**17671A0568**
RITIKA KALYANI	**17671A0596**

ACKNOWLEDGEMENT

At outset we express our gratitude to almighty lord for showering his grace and blessings upon us to complete this Main Project. Although our name appears on the cover of this book, many people had contributed in some form or the other to this project Development. We could not have done this Project without the assistance or support of each of the following.

First of all we are highly indebted to **Dr. P. C. KRISHNAMACHARY**, Principal for giving us the permission to carry out this Main Project.

We would like to thank **Dr. P. SRINIVASA RAO,** Professor & Head of the Department of COMPUTER SCIENCE AND ENGINEERING**,** for being moral support throughout the period of the study in the Department.

We are grateful to **DR. R. VIJAYANAND ,** Assistant Professor COMPUTER SCIENCE AND ENGINEERING, for his valuable suggestions and guidance given by him during the execution of this Project work.

We would like to thank Teaching and Non-Teaching Staff of Department of Computer Science & Engineering for sharing their knowledge with us.

B.PRAVALLIKA	**17671A0558**
B.BHARGAVI	**17671A0563**
N.ESHWITHA	**17671A0568**
RITIKA KALYANI	**17671A0596**

ABSTRACT

Recently, the huge amounts of data and its incremental increase have changed the importance of information security and data analysis systems. Integration of Information and Communications Technology into the distribution system makes today's power grid more remotely monitored and controlled than it has been. The fast increasing connectivity, however, also implies that the distribution grid today, or smart grid, is more vulnerable. Thus, research into intrusion/anomaly detection systems at the distribution level is in critical need. Current research on Intrusion Detection Systems for the power grid has been focused primarily on cyber security at the Supervisory Control And Data Acquisition, and single node levels with little attention on coordinated cyberattacks at multiple nodes. A holistic approach toward system-wide cyber security for distribution systems is yet to be developed. This presents an approach toward intrusion prevention, using a multi-agent system, at the distribution system level. Simulations of the method have been performed on the dataset, and the results compared to those obtained from existing methods. In this project, we are using Whale Optimization algorithm with K-NN (K-Nearest Neighbor) classifier on CICIDS2017 dataset. The results have validated the performance of the proposed method for protection against cyber intrusions at the distribution system level with 98.42% accuracy.

KEYWORDS: Intrusion detection, Cyberattacks, Intrusion prevention, Whale optimization, KNN Classifier.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

SNO.	WORD	ABBREVIATION
1	IDS	Intrusion detection system
2	IPS	Intrusion prevention system
3	DOS	Denial of service
4	DDOS	Distributed Denial of Service
5	DAS	Distribution automation system
6	PSO	Particle swarm optimization
7	GA	Genetic algorithm
8	GWO	Grey wolves optimization
9	FFA	Firefly algorithm
10	WOA	Whale optimisation algorithm
11	SVM	Support vector machine
12	K-NN	K-nearest neighbors
13	NN	neural network
14	ABC	Ant bee colony
15	DFD	Data flow diagram

1. INTRODUCTION

With the development of technology and the further popularization of computer, the use of network has become more extensive, which not only changes the way people study and work, but also creates great values for economic development. However, the network security problem is becoming more and more prominent, means of intrusion attack is becoming more complex and diverse, which means greater and stronger harms, and the difficulty of intrusion prevention is becoming higher. In order for ensuring the security of network infrastructure and communications through the Internet, several approaches and techniques have been developed. Intrusion detection and prevention systems, anti-virus software packages, and firewalls are examples of that method, and techniques have been wildly used to achieve security requirement. However, firewalls alone cannot defend against all types of intrusions and attacks, where intrusions try to break network security by taking advantage of vulnerabilities in the network. The detection of abnormal behaviors in the networks such as penetrations, break-ins, or any other form of suspicious activity is called intrusion detection. An intrusion detection system (IDS) is responsible to monitor all of the activities in the network and user behaviors to check if there are any suspicious activities or any violations in the specified policy. In addition, IDS can provide a report to the management station. Moreover, IDS is considered as an added wall that provides extra security to the network.

1.1 INTRUSION DETECTION SYSTEM(IDS):

The IDS is a method that determines if there are any threats caused by intrusions on the system throughout the observations of the network traffic. It is available around the clock to generate information regarding the state of the system, monitor the activities of the users, and provide reports to a management station. The classifications of IDS are network-based, host-based, and hybrid-based. The classification depends on the source and type of information for identifying security breaches. There is no standard definition for IDS which we consider as any breach to the system; however, this also does not report the issues properly. Governments sectors, private sectors, companies, small business establishments, health sectors, and even

individual users need to implement the IDS for identifying attacks and prevent in both host-based systems and network-based systems.

The operation contains set of rules and policies to identify any type of threats, attacks, or intrusions to gain unauthorized access to any source of data or intercept a package on its way to the destination. IoT devices that connect to the Internet directly can be subjected to several threats and can be attacked easily. Although there are several techniques that have been applied to protect such environment, for instance, safe configuration, up-to-date patching, and firewalls, all of them are not easy to maintain and cannot ensure that the system can be secure form different types of attacks. IDS provides protection in which it monitors network or systems for policy violations or malicious activity. An IDS works like a "guard" which monitors the network and provides better security than other measures.

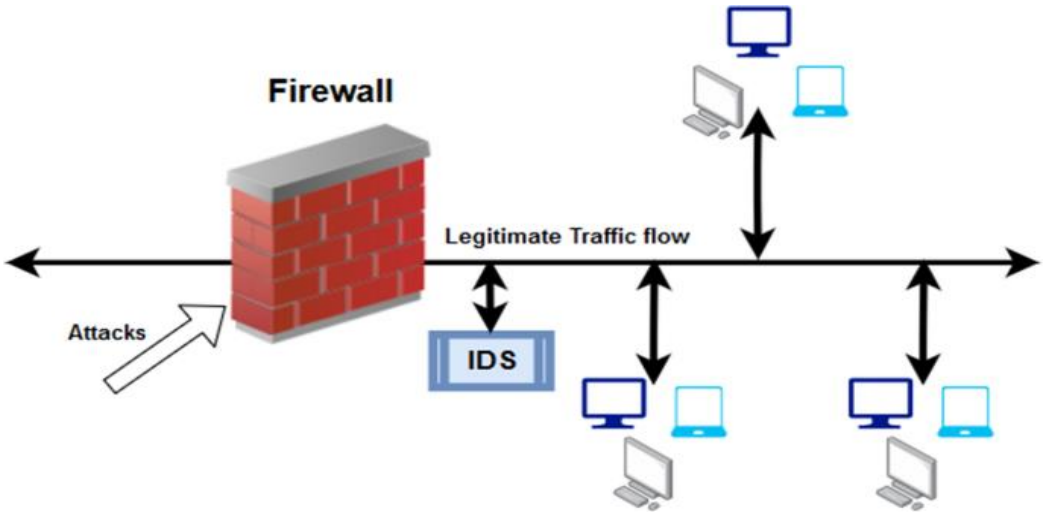

Fig:1.1.1 Intrusion detection system

Intrusion detection methods and techniques:

IDS can be classified into two main categories host-based and network-based:

i. **Host-based IDS** monitors a single system. In most cases, the IDS software runs on the host. It looks for logs and activities occurring on the system and tries to find anomalies.

ii. **Network-based IDS** system monitors a network segment in which IDS is sampling all the packets that pass through a specific point on the network. The network interface card listens to all the packets.

1.2 INTRUSION PREVENTION SYSTEM(IPS):

An intrusion prevention system (IPS) is a form of network security that works to detect and prevent identified threats. Intrusion prevention systems continuously monitor your network, looking for possible malicious incidents and capturing information about them. The IPS reports these events to system administrators and takes preventative action, such as closing access points and configuring firewalls to prevent future attacks. With so many access points present on a typical business network, it is essential that you have a way to monitor for signs of potential violations, incidents and imminent threats. Today's network threats are becoming more and more sophisticated and able to infiltrate even the most robust security solutions.

Intrusion prevention systems work by scanning all network traffic. There are a number of different threats that an IPS is designed to prevent, including:

- Denial of Service (DoS) attack
- Distributed Denial of Service (DDoS) attack
- Various types of exploits
- Worms
- Viruses

The IPS performs real-time packet inspection, deeply inspecting every packet that travels across the network. If any malicious or suspicious packets are detected, the IPS will carry out one of the following actions:

- Terminate the TCP session that has been exploited and block the offending source IP address or user account from accessing any application, target hosts or other network resources unethically.
- Reprogram or reconfigure the firewall to prevent a similar attack occurring in the future.

- Remove or replace any malicious content that remains on the network following an attack. This is done by repackaging payloads, removing header information and removing any infected attachments from file or email servers.

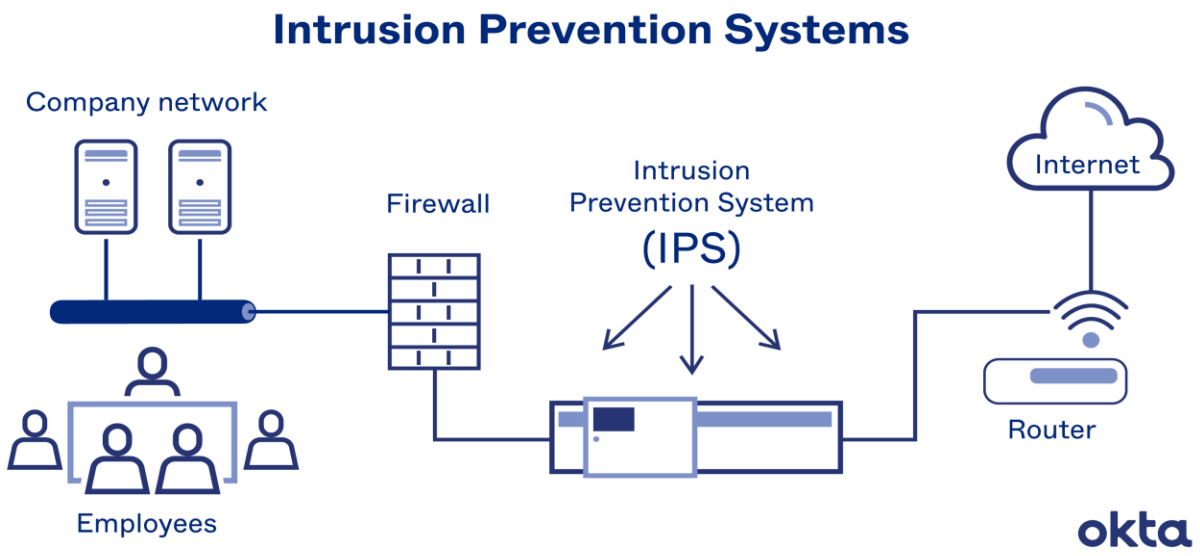

Fig:1.2.1 Intrusion prevention system

1.3 DISTRIBUTION AUTOMATION SYSTEMS(DAS):

DAS integrates communication with digital controls, switching devices, etc. to provide automated functionalities. Key Distribution Automation (DA) applications are outage management, feeder restoration, Volt-Var management, DER management, and condition monitoring. Some devices, e.g., smart reclosers, used in DA may be autonomous and not controlled from a remote location. The vulnerabilities of DA are mainly due to the use of unmonitored field devices and communication protocols with known vulnerabilities. DA systems may be subject to attacks such as Denial of Service (DoS), replay, packet modification, false packet injection, and physical tampering. However, due to the simplicity of field devices in a distribution system, sophisticated security algorithms may be impractical.

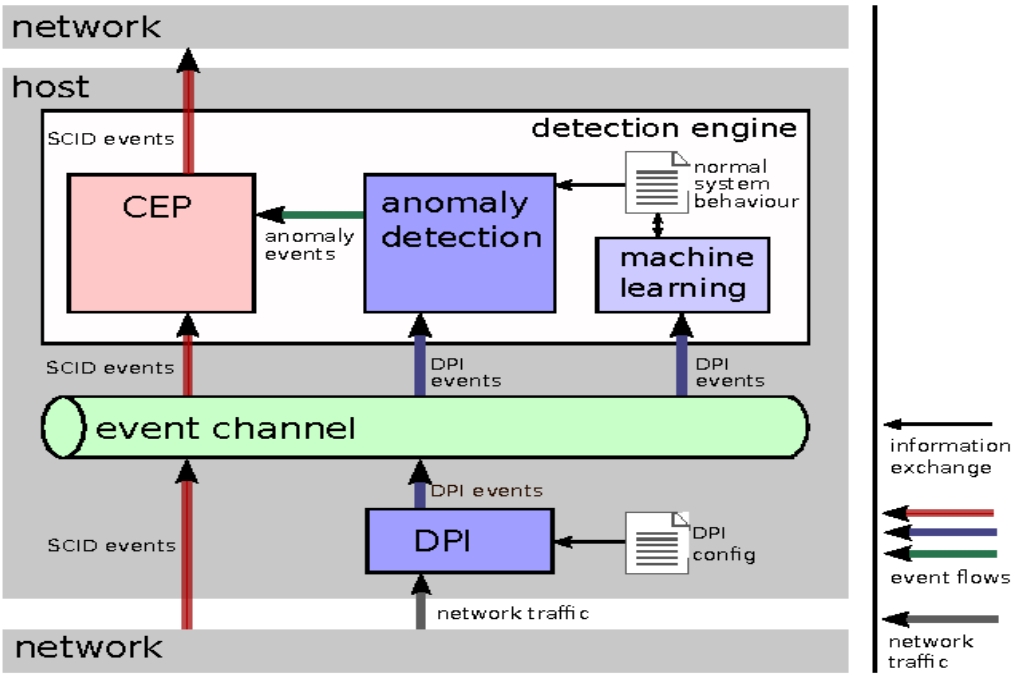

Fig:1.3.1 Distribution automation system

1.4 COORDINATED CYBER ATTACKS:

In a coordinated cyberattack, an attacker may use several attack strategies to attack one target, or may attack several parts of one system, or both. The literature on studies pertaining to coordinated cyberattacks is diverse. Cyber attacks often happen in stages, starting with hackers surveying or scanning for vulnerabilities or access points, initiating the initial compromise and then executing the full attack whether it's stealing valuable data, disabling the computer systems or both.

Reducing the risk of a cyber attack relies on using a combination of skilled security professionals, processes and technology.

Reducing risk also involves three broad categories of defensive action:

1. Preventing attempted attacks from actually entering the organization's IT systems
2. Detecting intrusions and

3. Disrupting attacks already in motion ideally, at the earliest possible time.

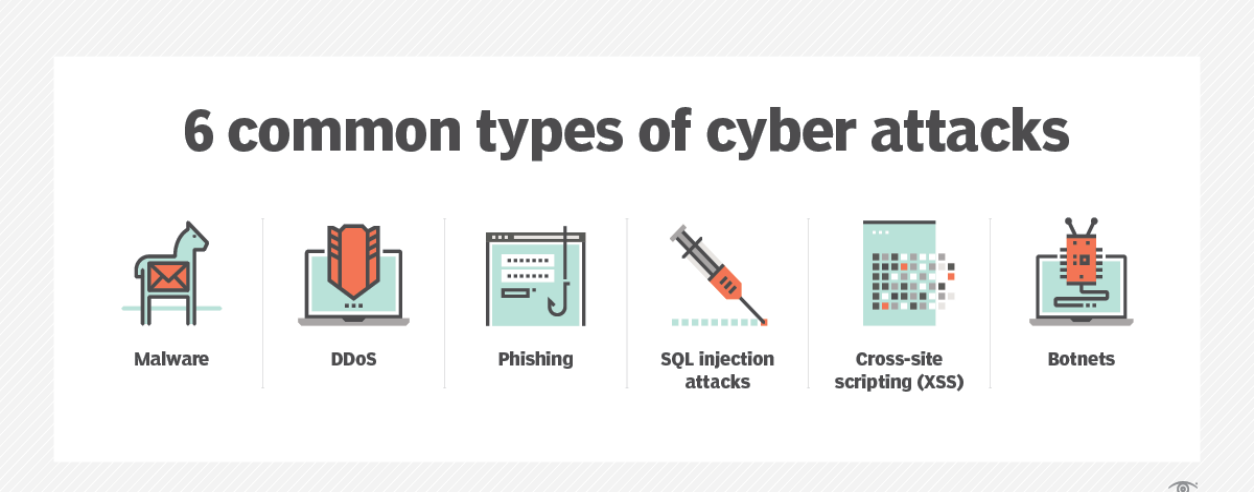

Fig:1.4.1 Types of cyberattacks

1.5 LIMITATIONS:

In spite of the advances in DAS technologies, research on intrusion detection systems, specifically for DAS, is in an early stage. There is also a lack of a response framework for DA cyberattacks. In addition, the current literature is primarily concentrated on DA applications with communication to a central office system. Furthermore, the proposed methods do not defend against coordinated cyberattacks, which may target multiple dispersed nodes. This underscores the critical need for a collaborative distributed approach towards intrusion prevention.

The literature on coordinated cyberattacks is heavily focused on the transmission system. Thus, the solutions proposed may be inapplicable at the distribution system level, due to hardware constraints, and the use of methods specific to the transmission system. In addition, the complete delegation of attack response strategies to the human operator is undesirable: in certain scenarios, the human operator may be unable to implement specific defense strategies. An example is when an attacker implements DoS so that the network is unreachable to the operator. Thus, there is a critical need to investigate coordinated attacks at the distribution system level and to formulate solutions in the context of its limitations.

1.6 EXISTING SYSTEM:

1.6.1 FEATURE SELECTORS:

Feature selection selects representative set of attributes from the set of original attributes. This representative set keeps only the relevant and important attributes, learning algorithm takes less time to learn and produces a more general classifier as it removes unnecessary and irrelevant attributes for the original set. Feature selection also facilitates data visualization and data understanding. Some of the popular feature selection techniques used in the past are briefly presented below.

PSO FEATURES SELECTION:

Particle swarm optimization (PSO) was developed by Russell Eberhart and James Kennedy. It was developed based on a simple concept derived from the movement of bird flocks and fish schools. It was developed after making several interpretations through using computer simulations. PSO employs a variety of agents (particles) that make up a swarm. This swarm travels around in the search space in order to find the solution deemed the best. Regarding each particle in the search space, it alters its “flying” to match its flying experience and other particles’ flying experience.

PSO is launched by randomly generated particles and their velocity, which indicate the search speed. Then, similar to the GA algorithm, the particles are evaluated in terms of fitness. Such evaluation is followed by two main tests. The first test compares the experience of a particle with itself, which is called personal best (pbest). The second test compares the fitness of a particle with the whole swarm experience. It is called global best (gbest). Performing these two tests leads to saving the best particle. After that, the termination criterion is met.

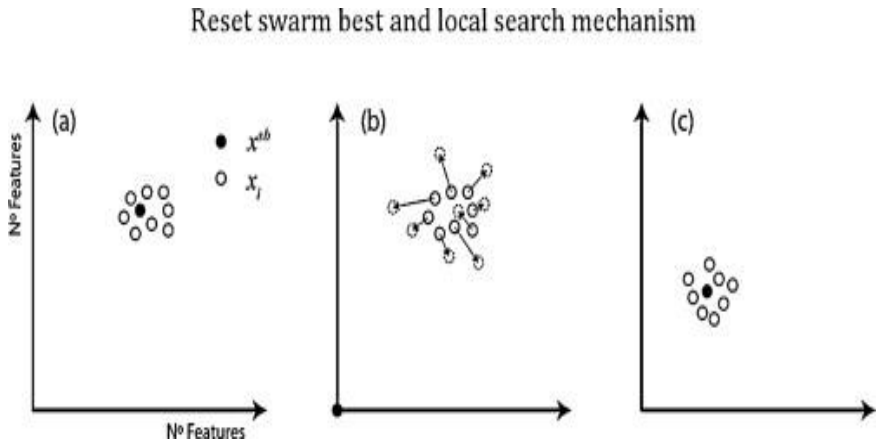

Fig:1.6.1.1 Particle swarm optimization

GA FEATURES SELECTION:

GA is an evolutionary search method that is employed for addressing the optimization problems based on a natural selection method. GA encodes a set of solutions for addressing the optimization problem. Those solutions are randomly generated to form a population. Then, GA evaluates this population in terms of a fitness function. The best solution is selected based on the problem being solved. It is assessed in terms of accuracy, root mean squared error (RMSE), F-measure or the area under curve (AUC). The fitter individuals were chosen for a set of reproduction operations, which are crossover and mutation. This operation gets repeated until it meets the termination criterion. This shall lead to forming a set of generations.

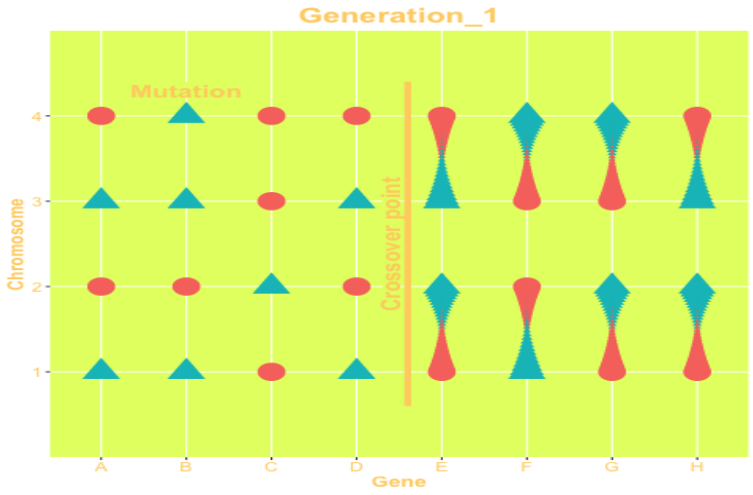

Fig:1.6.1.2 Genetic algorithm

GWO FEATURES SELECTION:

GWO was proposed by Mirjalili. It was developed through performing hunting procedures. It was developed based on the leadership skills of grey wolves. The social hierarchy of wolves is shown in Figure. It describes four kinds of wolves: beta, alpha, omega and delta.

Fig:1.6.1.3 Hierarchy of wolves

Alpha wolves are decision makers. They may not be the strongest in the pack, but they are certainly the best to manage the pack. Beta is a lower level wolf in the pack. It operates as an advisor to the alpha. It should be capable of taking the alpha's place in case of death or any other circumstances. Moreover, it reinforces the alpha's decisions among the members of the pack. It provides the alpha with the feedback of the members of the pack about the decision made by the alpha. Omega is deemed as the lowest level wolf among the pack. It acts as a scapegoat as the members of the pack submit to dominants. The existence of omega is very important. That is because omega preserves the dominant structure and satisfies the whole pack. Delta represents the rest of the pack which submits to beta and alpha. It includes: sentinels, scouts, elders, caretakers and hunters belonging to this level.

Based on this hierarchy, the group hunting process is performed through following three main steps. These steps are identified below:

1) Tracking the prey and chasing and approaching it;

2) Pursuing the prey and encircling and harassing it to stop its movement;

3) Launching an attack against the prey being attacked.

The algorithm mimics the whole described hierarchy and group hunting procedures. It mimics those procedures to solve complex engineering problems.

FFA FEATURES SELECTION:

The firefly optimization algorithm (FFA) for feature selection is a metaheuristic algorithm. It was proposed by Xin-She Yang. It is based on tropical fireflies' communication behavior. It is also based on the idealized flashing pattern behavior. FFA employs the following idealized rules to construct the mathematical model of the algorithm.

a) Regarding all the fireflies, they are unisex;

b) The brightness of the fireflies is proportional to their attractiveness;

c) The firefly's brightness is determined and influenced by the environment of the objective functions.

In terms of the maximization problem, the brightness may be proportional to the value of the objective function. The regular firefly algorithm includes two significant points. The first point is the formulation of the light intensity. The second point is the shift in attractiveness. One can always presume that the encoded objective feature landscape shall determine the brightness of the firefly. One should describe the light intensity difference and formulate the attractiveness adjustment.

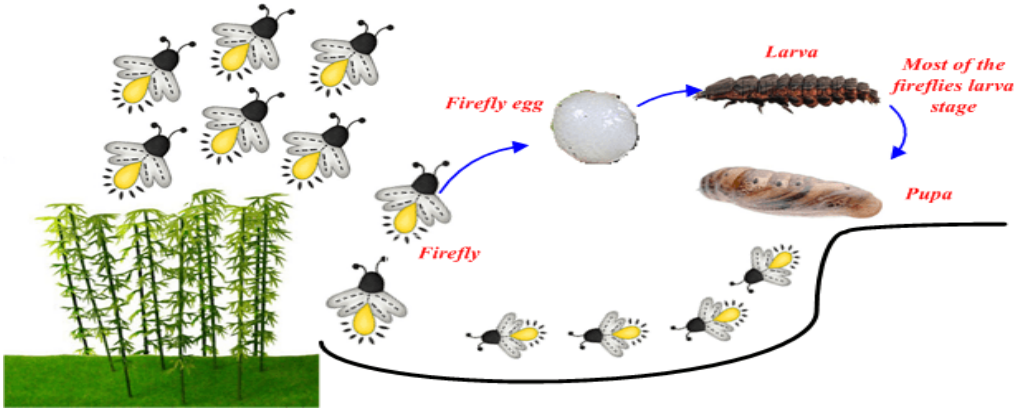

Fig:1.6.1.4 Firefly algorithm

1.6.2 CLASSIFIERS:

A classifier is a tool which categorizes unseen patterns in suitable classes. The classifier is first given training data which it uses to construct a decision model. Then the model is given some unseen examples to classify. Some of the popular classification techniques used are briefly presented below:

NAIVE BAYES:

Naive Bayes is a form of Bayes networks which are used for inference tasks . It is based on Bayes probability theory. It is a classification algorithm based on Bayes's theorem which gives an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Even if the features depend on each other, all of these properties contribute to the probability independently. Naive Bayes model is easy to make and is particularly useful for comparatively large data sets. Even with a simplistic approach, Naive Bayes is known to outperform most of the classification methods in machine learning.

SUPPORT VECTOR MACHINE:

It is based on the idea of structural risk minimization which gains advantage in speed and scalability . The basic idea of Support Vector Machine (SVM) is to find a decision boundary in multidimensional space which separates unseen patterns in different classes. SVM is a binary classifier. It is a common approach for making classifications between two classes. In SVM, a hyper plan is created to distinguish the positive sample class from the negative sample class based on the structural risk minimization principle. Alternatively, by choosing from different kernel functions, SVM can solve the problems of linear classification. SVM can get extended to nonlinear classification cases. It is a significant classification ML method because it employs the statistical learning theory. Furthermore, due o its use for the structure risk minimization method, SVM has a strong generalization capability.

DECISION TREE:

Decision Tree algorithms are well-known tool for classification and prediction tasks. It builds a model that predicts the output of a pattern based on different input attribute values of the

pattern. The construction of DT does not require any domain knowledge or parameter setting, just the given data set is learnt and modelled. It consists of three basic elements: Decision node, edges or branch and leaf node. The tree is constructed in a top-down recursive divide and conquer approach. A decision node will have two or more branches and a leaf represents a classification or decision. The topmost node in the decision tree that corresponds to the best predictor is called the root node, and the best thing about a decision tree is that it can handle both categorical and numerical data.

RANDOM FOREST:

Random decision trees or random forest are an ensemble learning method for classification, regression, etc. It operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes or classification or mean prediction(regression) of the individual trees. A random forest is a meta-estimator that fits a number of trees on various subsamples of data sets and then uses an average to improve the accuracy in the model's predictive nature. The sub-sample size is always the same as that of the original input size but the samples are often drawn with replacements.

NEURAL NETWORK:

Neural Network (NN) is a connectionist approach which processes information through a connection of large number of artificial neurons. It consists of hidden layers which further processes the data before passing the output to the output layer. To train the NN a pattern from training data set is given as input to the NN. The output obtained is observed, and if it is correct the next pattern is given as input. In case of error, the error is propagated till the input layer using back-propagation algorithm and weights are adjusted to obtain correct output for all the training patterns. Once trained the, newer unseen data is fed to the trained NN and the output obtained classifies to which class it belongs.

1.7 PROPOSED SYSTEM:

1.7.1 WOA FEATURE SELECTION:

Whale Optimization Algorithm (WOA) is a recently proposed (2016) optimization algorithm mimicking the hunting mechanism of humpback whales in nature. According to Hof and Van Der Gucht , whales have common cells in certain areas of their brains similar to those of human called spindle cells. These cells are responsible for judgment, emotions, and social behaviors in humans. In other words the spindle cells make us distinct from other creatures. Whales have twice number of these cells than an adult human which is the main cause of their smartness. It has been proven that whale can think, learn, judge, communicate, and become even emotional as a human does, but obviously with a much lower level of smartness. It has been observed that whales (mostly killer whales) are able to develop their own dialect as well. The most interesting thing about the humpback whales is their special hunting method. This foraging behavior is called bubble-net feeding method . Humpback whales prefer to hunt school of krill or small fishes close to the surface. It has been observed that this foraging is done by creating distinctive bubbles along a circle or '9'-shaped path. Before 2011, this behavior was only investigated based on the observation from surface. However, Goldbogen et al. investigated this behavior utilizing tag sensors. They captured 300 tag-derived bubble-net feeding events of 9 individual humpback whales. They found two maneuvers associated with bubble and named them 'upward-spirals' and 'double-loops'. In the former maneuver, humpback whales dive around 12 meters down and then start to create bubble in a spiral shape around the prey and swim up toward the surface. The later maneuver includes three different stages: coral loop, lobtail, and capture loop.

This method contains the following three stages:

- **Circling hunting**: Humpback whales can recognize the location of prey and encircle them. Since the position of the optimal design in the search space is not known a priori, the WOA algorithm assumes that the current best candidate solution is the target prey or is close to the optimum. After the best search agent is defined, the other search agents will hence try to update their positions towards the best search agent.

- **Bubble-net attacking:** Two approaches are designed in order to understand the bubble net behavior of humpback whales that is called the exploitation phase.

(1) Encircling Prey: After the humpback whales discover the position of the prey, they encircle around them. Firstly, the location of the optimal design in the search space is unidentified; thus, the WOA algorithm assumes that the present leading candidate solution is the target prey or near to the optimum. Then the other search agents will attempt to change their locations to the best search agents.

(2) Spiral Updating Position: After calculating the distance between the whale located at (X, Y) and prey located at (X*,Y*). At that point, a spiral equation is generated between the location of the whale and prey to imitate the helix-shaped movement of humpback whales a

- **Prey hunting:** In the search phase for the prey, which is called the exploration phase, the whales actually use random search to discover their prey depending on the position of each other. Throughout the exploration phase, the location of a search agent is reorganized according to randomly selected search agent rather than the best search agent (exploitation phase).

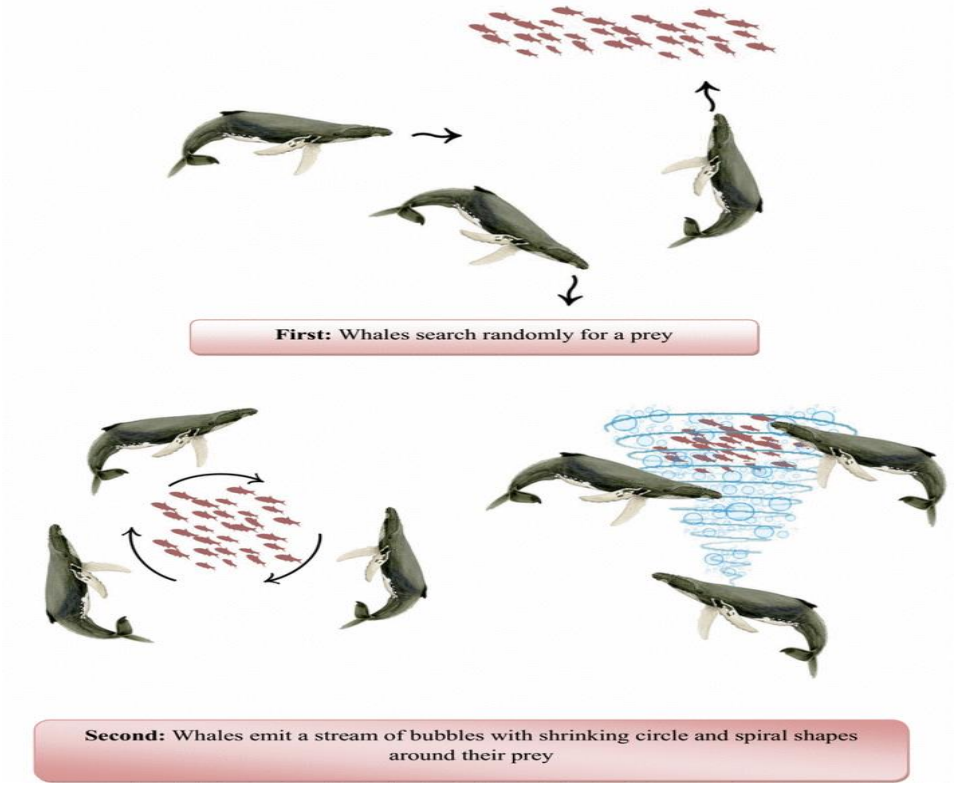

Fig:1.7.1.1 Whale optimization algorithm

ADVANTAGES:

- Appropriate for solving different optimization problems.
- Helps to get global optimization solutions.

1.7.2 K- NEAREST NEIGHBORS(K-NN) CLASSIFIER:

It is a lazy learning algorithm that stores all instances corresponding to training data in n-dimensional space. It is a lazy learning algorithm as it does not focus on constructing a general internal model, instead, it works on storing instances of training data.

Classification is computed from a simple majority vote of the k nearest neighbors of each point. It is supervised and takes a bunch of labeled points and uses them to label other points. To

label a new point, it looks at the labeled points closest to that new point also known as its nearest neighbors. It has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point. The “k” is the number of neighbors it checks.

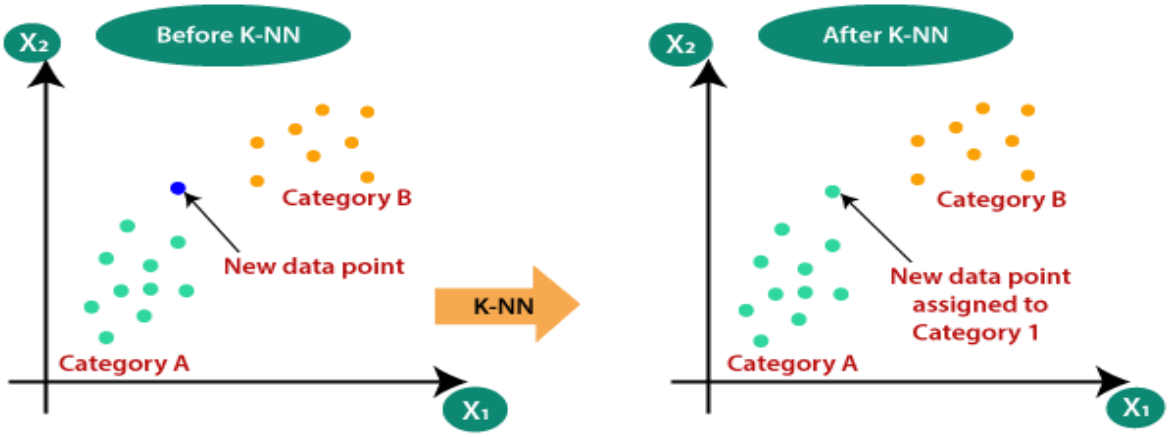

Fig:1.7.2.1 K-NN classification

ADVANTAGES:

- Quick calculation time.
- Simple classification– to interpret.
- Versatile – useful for regression and classification.
- High accuracy – you do not need to compare with better-supervised learning models.

1.8 SYSTEM SCOPE:

An **intrusion prevention system** (IPS) is a network security tool (which can be a hardware device or software) that continuously monitors a network for malicious activity and takes action to prevent it, including reporting, blocking, or dropping it, when it does occur.

It is more advanced than an intrusion detection system (IDS), which simply detects malicious activity but cannot take action against it beyond alerting an administrator. Intrusion prevention systems are sometimes included as part of a next-generation firewall or unified threat management solution. Like many network security technologies, they must be powerful enough to scan a high volume of traffic without slowing down network performance.

There are several reasons why an IPS is a key part of any enterprise security system. A modern network has many access points and deals with a high volume of traffic, making manual monitoring and response an unrealistic option. (This is particularly true when it comes to cloud security, where a highly connected environment can mean an expanded attack surface and thus greater vulnerability to threats.) In addition, the threats that enterprise security systems face are growing ever more numerous and sophisticated. The automated capabilities of an IPS are vital in this situation, allowing an enterprise to respond to threats quickly without placing a strain on IT teams. As part of an enterprise's security infrastructure, an IPS is a crucial way to help prevent some of the most serious and sophisticated attacks.

2. LITERATURE SURVEY

The following are the experiments conducted by various several authors in past on feature selection algorithms for intrusion detection:

In [1], Ahmad et al. developed a feature selection model that is based on multilayer perception (MLP) for IDSs. This model is based on a combination of principal component analysis (PCA) and GA. The latter researchers conducted PCA to plan the features space to a principal feature space. They selected the features corresponding to the highest eigenvalues. The features that were selected by PCA may lack the adequate detection for the classifier. So, the researchers adopted GA to explore the principal feature space in order to find a subset with optimal sensitivity.

In [2], Ghanem and Jantan developed an artificial bee colony (ABC) method for the feature selection of IDSs. The latter method consists of two stages: Through stage 1, the subsets of features were generated through using the Pareto front non-dominated solutions; - Through stage 2, a feed forward neural network (FFNN) and ABC (and PSO) were employed for assessing the feature subsets that were derived from the first stage. Thus, the proposed method employs a new feature selection model. It is called (the multi-objective ABC method). It aims at reducing the number of network traffic features. The latter method adopts a new classification approach.

In [3], Researchers proposed a model to select features for IDSs. Those features were selected through using evolutionary algorithms: GA, PSO and differential evolution. They conducted a comparison between these algorithms in terms of efficiency.

In [4], Researchers in proposed a new IDS model. The latter model is based on intelligent dynamic swarm through the use of a rough set. It is abbreviated as (IDS-RS) and simplified swarm optimization (SSO). It is considered as a new version of PSO that employs a

new weighted local search strategy. IDS-RS is conducted with a weighted sum fitness function to choose the most important features for having the features of the dataset reduced.

In [5], the researcher in aimed to explore the performance level of the feature selection model that is in the NIDS. They aimed to explore that through using GA and PSO as algorithms for feature selection. GA and PSO played an effective role in having the number of the selected features reduced. The latter researchers found that GA can successfully reduce the number of the selected features from 41 features to 15 features. They found that PSO can have the number of the selected features successfully reduced from 41 features to 9 features.

In [6], Researchers employed the grey wolf optimization (GWO) method to search the feature space to find the optimal feature subset that improves the classification accuracy. The latter method used mutual information and filter-based principles. Second, the wrapper approach was adopted to raise the accuracy of the classifiers.

In [7], Researchers used the firefly algorithm based on the filter and wrapper methods to select the features. They also proposed a procedure for raising the dimensionality. They used the wrapper ensemble method with the Bayesian network, C4.5 and mutual information (MI).

In [8], Researchers introduced the Whale Optimization Algorithm inspired by humpback whales. The WOA algorithm is benchmarked on 29 well-known test functions. The results on the unimodal functions show the superior exploitation of WOA. The exploration ability of WOA is confirmed by the results on multimodal functions. The results on structural design problems confirm the performance of WOA in practice.

3. SOFTWARE REQUIREMENTS ANALYSIS

3.1 SOFTWARE REQUIREMENTS

- Operating System : Windows10 (64 bit)
- Front End : Python
- Tool Used : Python IDLE 3.8
- Python Modules : Numpy , Pandas, Sklearn , Matplotlib

3.2 HARDWARE REQUIREMENTS

- Processor : Dual core
- Ram : 16 GB (minimum)
- Hard Disk : 20GB

3.3 SYSTEM MODULES:

- Pre-processing
- Feature extraction
- Classification
- Validation of model
- Attack prevention

4. SYSTEM DESIGN

4.1 SYSTEM AECHITECTURE:

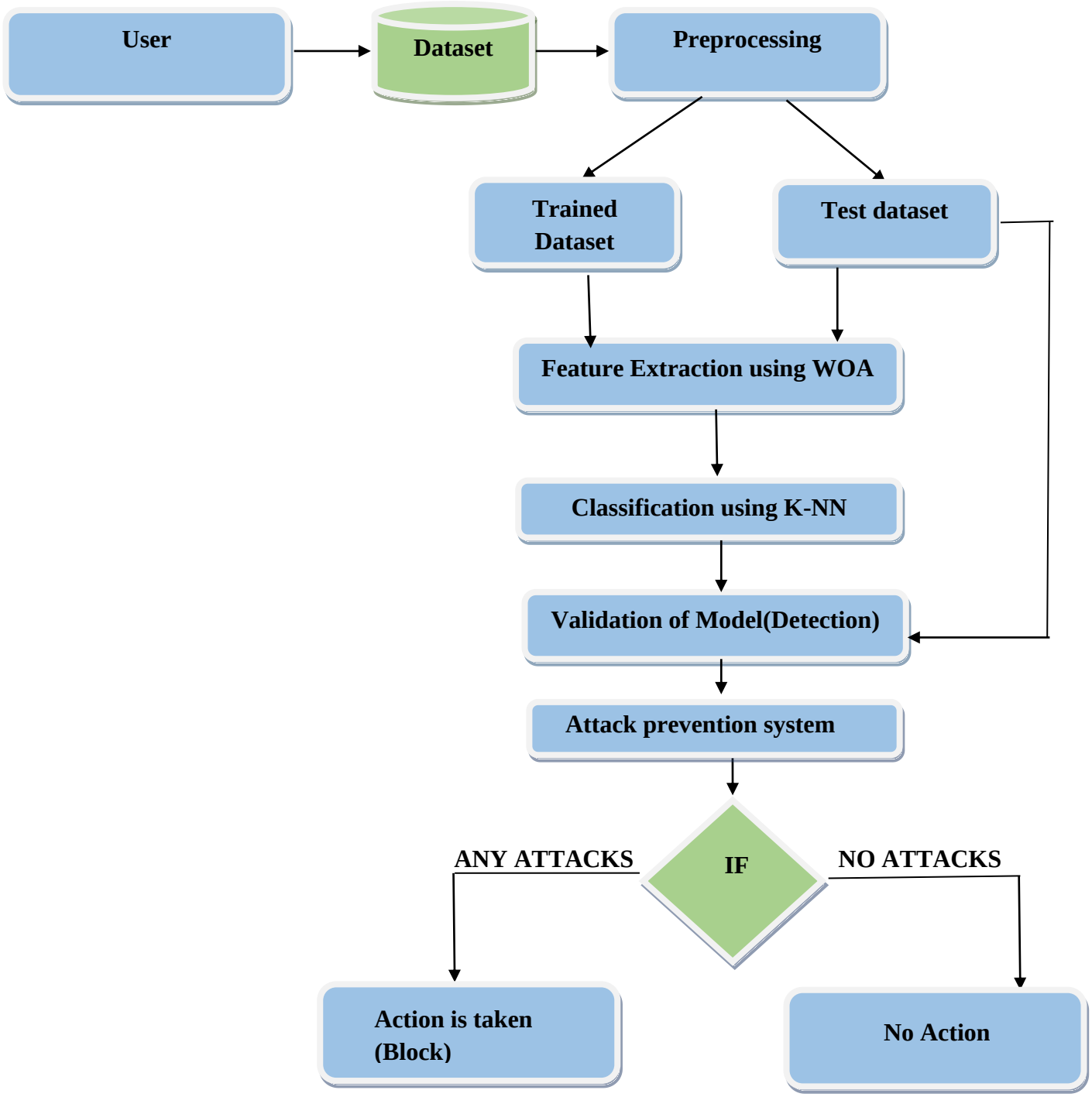

Fig:4.1 System Architecture

ARCHITECTURE DESCRIPTION:

DATSET:

Given a labeled data set in which each data point is assigned to the class normal or attack, the number of detected attacks or the number of false alarms may be used as evaluation criteria. Basically it consists of attributes which are commonly used in the identification of intrusion in an individual system. The main “goal” is to find the presence of the intrusion . The overall data consists of all the coordinated cyberattacks results from various platforms that are augmented to form a dataset.

CICIDS2017 dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files).

CLEANING DATA:

Data cleansing is the process of altering data in a given storage resource to make sure that it is accurate and correct. There are many ways to pursue data cleansing in various software and data storage architectures; most of them center on the careful review of data sets and the protocols associated with any particular data storage technology. Data cleansing is also known as data cleaning or data scrubbing.

PREPROCESSING:

Data preprocessor is responsible for collecting and providing the audit data (in a specified form) that will be used by the next module to make a decision. Data preprocessor is, thus, concerned with collecting the data from the dataset provided and converting it into a format that is understandable by the intrusion detector. The dataset is passed through several preprocessing steps

- **The removal of labels:** Each feature in the original given dataset has a label. Removing those labels is important in order to adapt the dataset with the framework environment
- **Removing features:** The original given dataset has many features. Some features of those features are class labels which cannot be considered as a feature. Thus, it is important to delete them. Deleting them is important because the main objective sought from this work is represented in reducing the features.
- **Label encoding:** Some labels in the dataset are given string values. Therefore, it is very significant to have those values encoded into numerical values.
- **Data binarization:** The numerical data in the dataset are in various ranges. During the training process, these data provide the classifier with a variety of challenges in order to compensate for such variations. Therefore, the values in each feature must be standardized. Thus, the least value in each one of the features should be 0. However, the maximum value should be 1. It makes the classifier more homogeneous. It preserves the difference between the values of each feature.

TRAINING AND TESTING DATASET:

Simply put, training data is used to train an algorithm. Generally, training data is a certain percentage of an overall dataset along with testing set. As a rule, the better the training data, the better the algorithm or classifier performs. Assuming that your test set meets the preceding two conditions, your goal is to create a model that generalizes well to new data. Our test set serves as a proxy for new data.

Some test data is used to confirm the expected result, i.e. When test data is entered the expected result should come and some test data is used to verify the software behavior to invalid input data. Test data is generated by testers or by automation tools which support testing. Most of the times in regression testing the test data is re-used, it is always a good practice to verify the test data before re-using it in any kind of test.

The observations in the training set form the experience that the algorithm uses to learn. The test set is a set of observations used to evaluate the performance of the model using some performance metric. It is important that no observations from the training set are included in the

test set. Training and testing on the same data is not an optimal approach, so we do split the data into two pieces, training set and testing set. We use 'train_test_split' function to split the data. Optional parameter 'test-size' determines the split percentage.

FEATURE SELECTION:

Feature selection selects representative set of attributes from the set of original attributes. This representative set keeps only the relevant and important attributes, learning algorithm takes less time to learn and produces a more general classifier as it removes unnecessary and irrelevant attributes for the original set. Feature selection also facilitates data visualization and data understanding.

Methods for feature selection are generally classified into filter and wrapper methods. Filter algorithms utilize an independent measure (such as, information measures, distance measures, or consistency measures) as a criterion for estimating the relation of a set of features, while wrapper algorithms make use of particular learning algorithms to evaluate the value of features. In comparison with filter methods, wrapper methods are often much more computationally expensive when dealing with high-dimensional data or large-scale data. In this study hence, we focus on filter methods for IDS. Due to the continuous growth of data dimensionality, feature selection as a pre-processing step is becoming an essential part in building intrusion detection systems.

In this project, a feature selection system is introduced applies the whale optimization algorithm (WOA). WOA is a recently introduced meta-heuristic optimization algorithm that mimics the natural behavior of the humpback whales. The proposed model applies the wrapper-based method to reach the optimal subset of features. This technique was applied to find the best feature subset that maximizes the accuracy of the classification while preserving the minimum number of features.

MACHINE LEARNING MODEL:

Machine learning (ML) is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic

premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. Beyond personalized marketing, other common machine learning use cases include fraud detection, spam filtering, network security threat detection, predictive maintenance and building news feeds.

CLASSIFICATION:

Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories. The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into.

A classifier is a tool which categorizes unseen patterns in suitable classes. The classifier is first given training data which it uses to construct a decision model. Then the model is given some unseen examples to classify. Once the optimal subset of features is selected, this subset is then taken into the classifier training phase where K-NN is employed. If the similarity score of one training normal process is equal to 1, which means the system call frequencies of the new process and the training process match perfectly, then the new process would be classified as a normal process immediately. Otherwise, the similarity scores are sorted and the k nearest neighbors are chosen to determine whether the new program execution is normal or not. We calculate the average similarity value of the k nearest neighbors (with highest similarity scores) and set a threshold. Only when the average similarity value is above the threshold, is the new process considered normal. The performance of the K-NN classifier algorithm also depends on the value of k, the number of nearest neighbors of the test process. Usually the optimal value of k is empirically determined. We applied the k value as ‘5’.

VALIDATION OF MODEL(DETECTION):

Model Validation is the task of confirming that the outputs of a statistical model have enough fidelity to the outputs of the data-generating process that the objectives of the

investigation are achieved. After completing all the aforementioned steps and the classifier is trained using the optimal subset of features which includes the most correlated and important features, the normal and intrusion traffics can be identified by using the saved trained classifier. The test data is then directed to the saved trained model to detect intrusions. Records matching to the normal class are considered as normal data, and the other records are reported as attacks. If the classifier model confirms that the record is abnormal, the subclass of the abnormal record (type of attacks) can be used to determine the record's type.

ATTACK PREVENTION:

Major functions of intrusion prevention systems are to identify malicious activity, collect information about this activity, report it and attempt to block or stop it. IPS security solutions can stop any attack based on malicious traffic sent over a network, provided it has a known attack signature, or can be detected as anomalous compared to normal traffic. If there is any malicious activity encountered then the main system connected in network blocks the network or traffic from the system which is sending malicious traffic/signals. In this way the anomaly/intrusion can be prevented. And if there are no malicious traffic the system performs no action and allows good traffic to pass through.

4.2 DATA FLOW DIAGRAM:

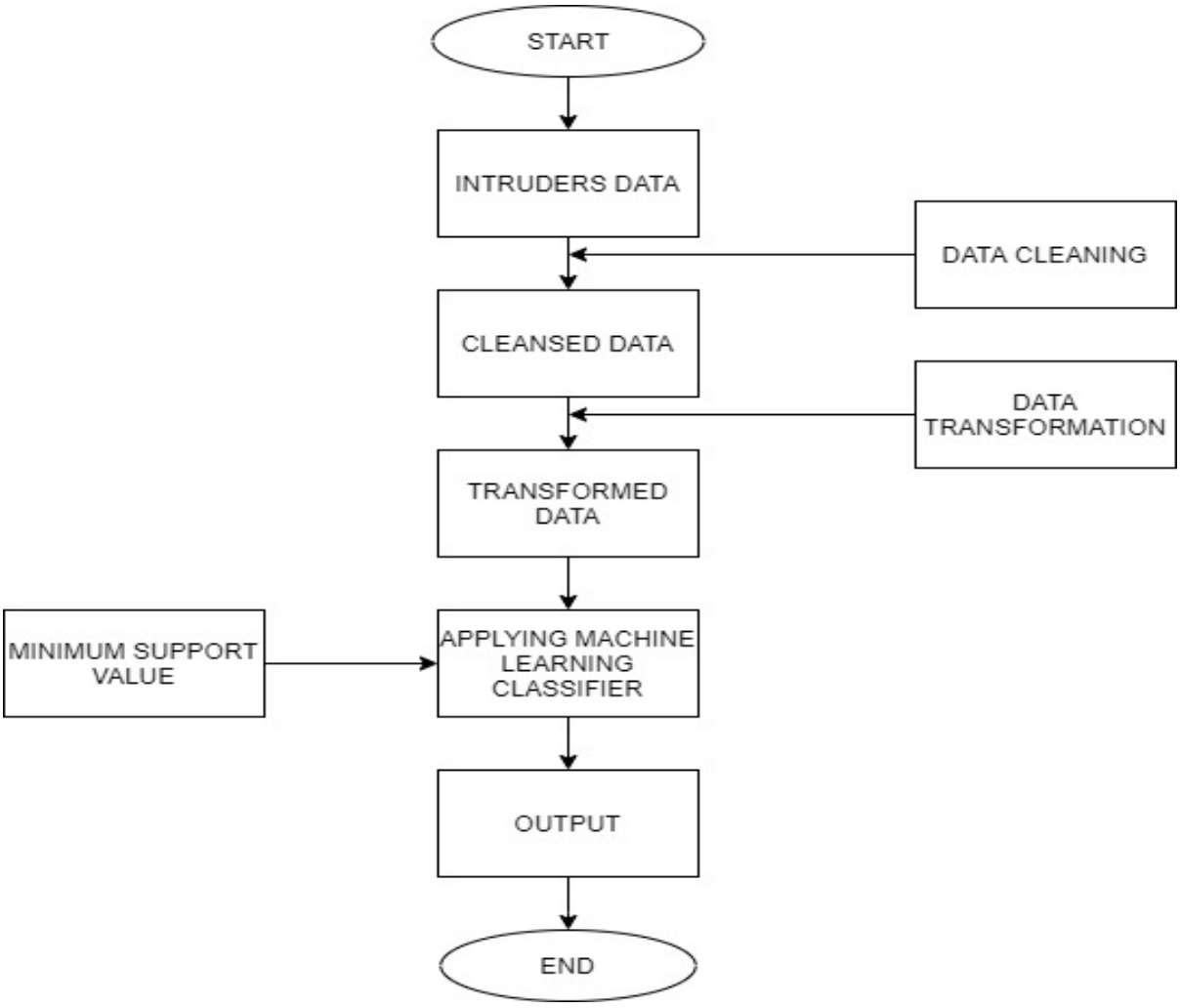

Fig 4.2 Data flow diagram

A picture is worth a thousand words. A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

4.3 UML DIAGRAMS:

4.3.1 USE CASE DIAGRAM:

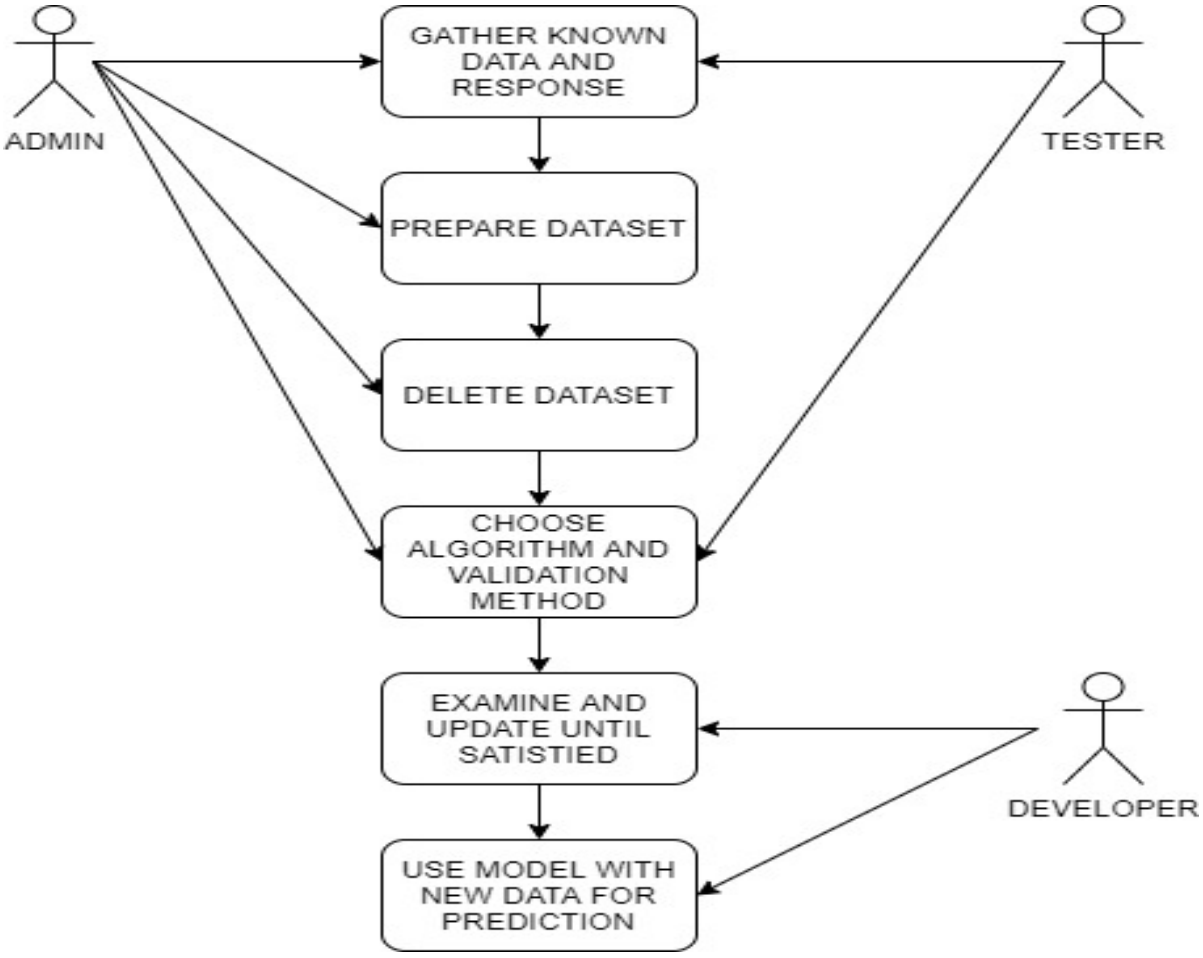

Fig:4.3.1 Use case diagram

Use case diagrams represent the functionality of the system from a user point of view. A Use case describes a function provided by the system that yields a visible result for an actor. An actor describe any entity that interacts with the system.

4.3.2 CLASS DIAGRAM:

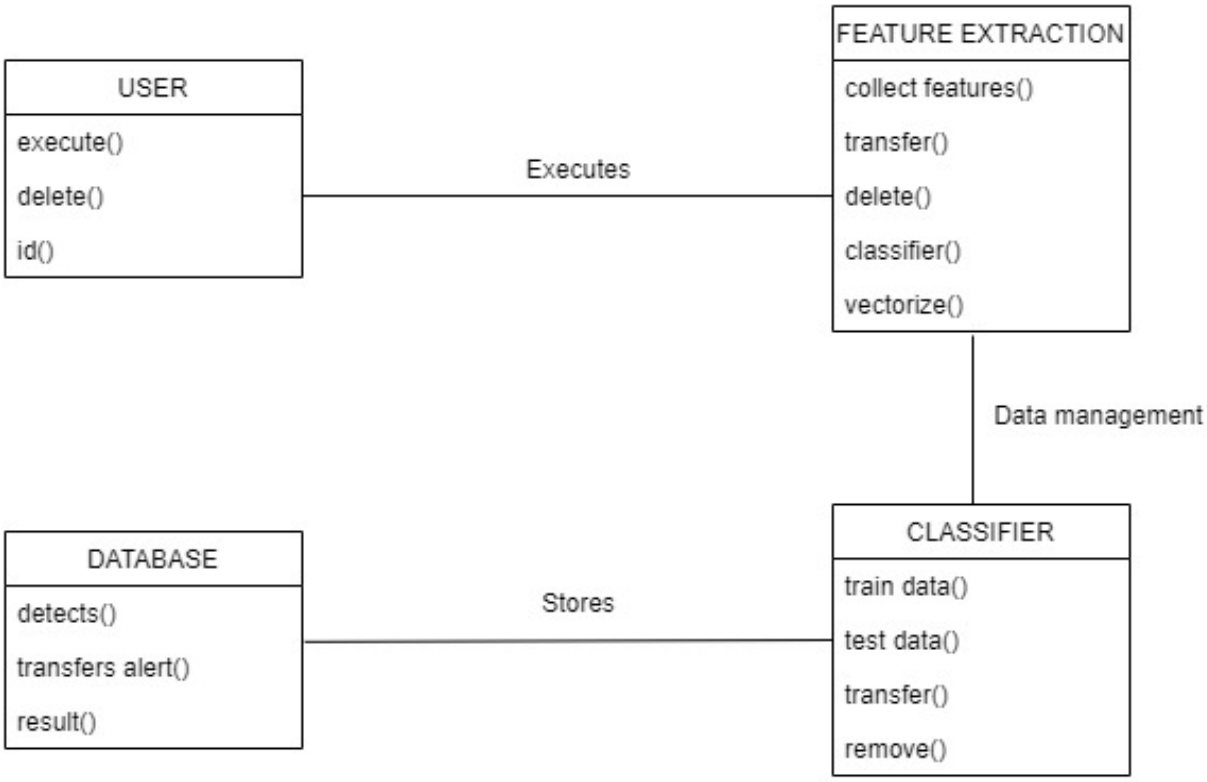

Fig:4.3.2 Class diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints.

Classes:

- Data
- Input Results

Interfaces:

- Preprocessing
- Training

4.3.3 SEQUENCE DIAGRAM:

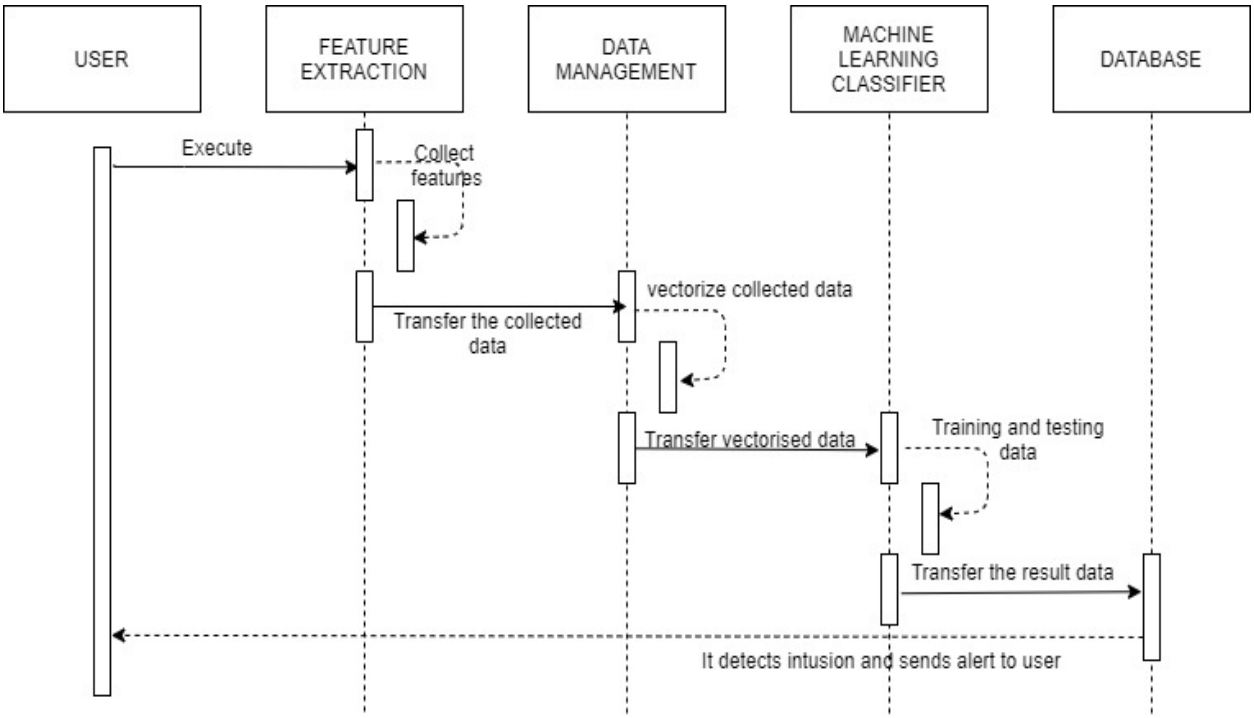

Fig:4.3.3 Sequence diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. The sequence of object interactions are

- Input
- Training
- Store in database
- Take user input
- Search for response
- Gives response

5. IMPLEMENTATION

5.1 SOFTWARE ENVIRONMENT:

MACHINE LEARNING USING PYTHON

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine Learning vs. Traditional Programming

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

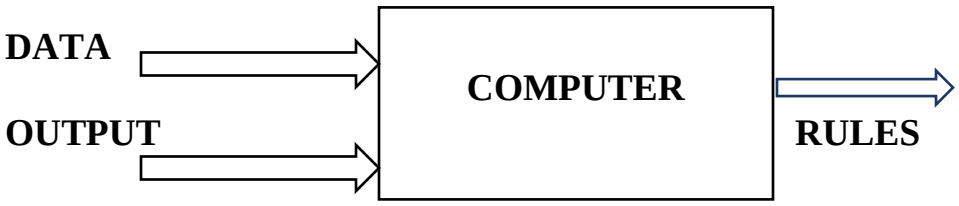

Fig:5.1.1 Machine Learning Model

How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.

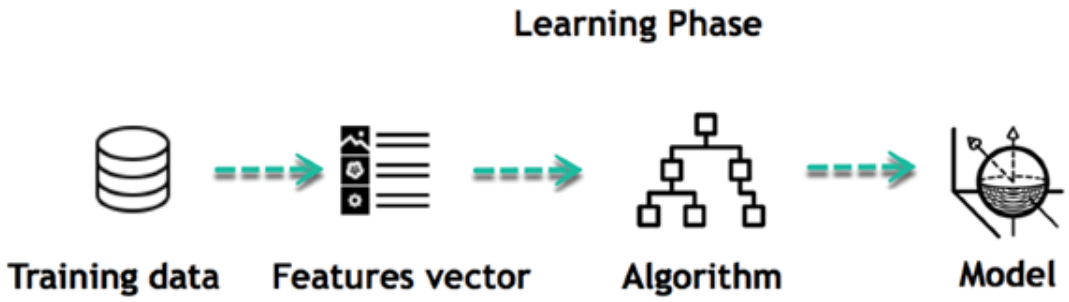

Fig:5.1.2 Learning phase

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant

Inferring:

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

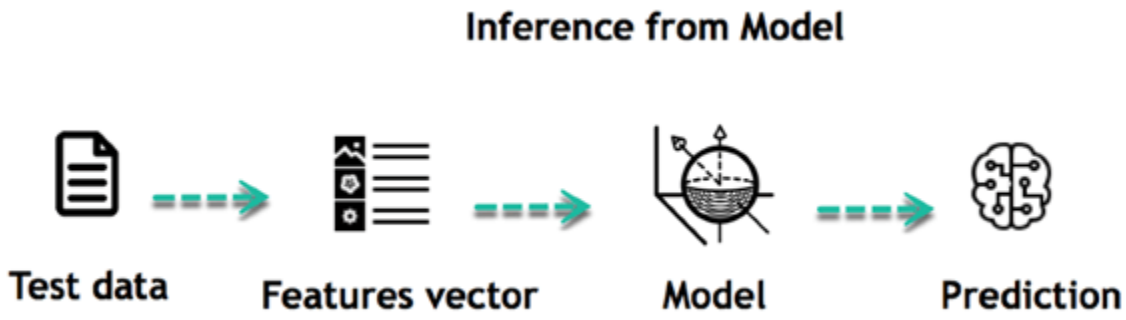

Fig:5.1.2 Inference from model

The life of Machine Learning programs is straight forward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

5.2 SOURCE CODE:

MAIN.PY:

```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from FS.woa import jfs
import matplotlib.pyplot as plt
# load data
data  = pd.read_csv('iscxdata2.csv')
data  = data.values
feat  = np.asarray(data[:, 0:-1])
label = np.asarray(data[:, -1])
# split data into train & validation (70 -- 30)
xtrain, xtest, ytrain, ytest = train_test_split(feat, label, test_size=0.3, stratify=label)
fold = {'xt':xtrain, 'yt':ytrain, 'xv':xtest, 'yv':ytest}
# parameter
k    = 5     # k-value in KNN
N    = 10    # number of particles
T    = 100   # maximum number of iterations
opts = {'k':k, 'fold':fold, 'N':N, 'T':T}
# perform feature selection
fmdl = jfs(feat, label, opts)
sf   = fmdl['sf']
# model with selected features
```

```
num_train = np.size(xtrain, 0)
num_valid = np.size(xtest, 0)
x_train   = xtrain[:, sf]
y_train   = ytrain.reshape(num_train)  # Solve bug
x_valid   = xtest[:, sf]
y_valid   = ytest.reshape(num_valid)  # Solve bug
mdl       = KNeighborsClassifier(n_neighbors = k)
mdl.fit(x_train, y_train)
# accuracy
y_pred    = mdl.predict(x_valid)
Acc       = np.sum(y_valid == y_pred)  / num_valid
print("Accuracy:", 100 * Acc)
# number of selected features
num_feat = fmdl['nf']
print("Feature Size:", num_feat)
# plot convergence
curve   = fmdl['c']
curve   = curve.reshape(np.size(curve,1))
x       = np.arange(0, opts['T'], 1.0) + 1.0
fig, ax = plt.subplots()
ax.plot(x, curve, 'o-')
ax.set_xlabel('Number of Iterations')
ax.set_ylabel('Fitness')
ax.set_title('WOA')
ax.grid()
```

```
plt.show()
```

WOA.PY:

```
import numpy as np

from numpy.random import rand

from FS.functionHO import Fun

def init_position(lb, ub, N, dim):

X = np.zeros([N, dim], dtype='float')

for i in range(N):

for d in range(dim):

X[i,d] = lb[0,d] + (ub[0,d] - lb[0,d]) * rand()

return X

def binary_conversion(X, thres, N, dim):

Xbin = np.zeros([N, dim], dtype='int')

for i in range(N):

for d in range(dim):

if X[i,d] > thres:

Xbin[i,d] = 1

else:

Xbin[i,d] = 0

return Xbin

def boundary(x, lb, ub):
```

```
if x < lb:

x = lb

if x > ub:

x = ub

return x

def jfs(xtrain, ytrain, opts):

# Parameters

ub    = 1

lb    = 0

thres = 0.5

b     = 1       # constant

N       = opts['N']

max_iter = opts['T']

if 'b' in opts:

b    = opts['b']

# Dimension

dim = np.size(xtrain, 1)

if np.size(lb) == 1:

ub = ub * np.ones([1, dim], dtype='float')

lb = lb * np.ones([1, dim], dtype='float')

# Initialize position
```

```
X    = init_position(lb, ub, N, dim)

# Binary conversion

Xbin = binary_conversion(X, thres, N, dim)

# Fitness at first iteration

fit  = np.zeros([N, 1], dtype='float')

Xgb  = np.zeros([1, dim], dtype='float')

fitG = float('inf')

for i in range(N):

fit[i,0] = Fun(xtrain, ytrain, Xbin[i,:], opts)

if fit[i,0] < fitG:

Xgb[0,:] = X[i,:]

fitG     = fit[i,0]

# Pre

curve = np.zeros([1, max_iter], dtype='float')

t     = 0

curve[0,t] = fitG.copy()

print("Generation:", t + 1)

print("Best (WOA):", curve[0,t])

t += 1

while t < max_iter:

# Define a, linearly decreases from 2 to 0
```

```
a = 2 - t * (2 / max_iter)

for i in range(N):

# Parameter A (2.3)

A = 2 * a * rand() - a

# Paramater C (2.4)

C = 2 * rand()

# Parameter p, random number in [0,1]

p = rand()

# Parameter l, random number in [-1,1]

l = -1 + 2 * rand()

# Whale position update (2.6)

if p  < 0.5:

# {1} Encircling prey

if abs(A) < 1:

for d in range(dim):

# Compute D (2.1)

Dx     = abs(C * Xgb[0,d] - X[i,d])

# Position update (2.2)

X[i,d] = Xgb[0,d] - A * Dx

# Boundary

X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])
```

```
# {2} Search for prey

elif abs(A) >= 1:

for d in range(dim):

# Select a random whale

k      = np.random.randint(low = 0, high = N)

# Compute D (2.7)

Dx     = abs(C * X[k,d] - X[i,d])

# Position update (2.8)

X[i,d] = X[k,d] - A * Dx

# Boundary

X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])

# {3} Bubble-net attacking

elif p >= 0.5:

for d in range(dim):

# Distance of whale to prey

dist   = abs(Xgb[0,d] - X[i,d])

# Position update (2.5)

X[i,d] = dist * np.exp(b * l) * np.cos(2 * np.pi * l) + Xgb[0,d]

# Boundary

X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])

# Binary conversion
```

```
Xbin = binary_conversion(X, thres, N, dim)

# Fitness

for i in range(N):

fit[i,0] = Fun(xtrain, ytrain, Xbin[i,:], opts)

if fit[i,0] < fitG:

Xgb[0,:] = X[i,:]

fitG     = fit[i,0]

# Store result

curve[0,t] = fitG.copy()

print("Generation:", t + 1)

print("Best (WOA):", curve[0,t])

t += 1

# Best feature subset

Gbin       = binary_conversion(Xgb, thres, 1, dim)

Gbin       = Gbin.reshape(dim)

pos        = np.asarray(range(0, dim))

sel_index  = pos[Gbin == 1]

num_feat   = len(sel_index)

# Create dictionary

woa_data = {'sf': sel_index, 'c': curve, 'nf': num_feat}

return woa_data
```

FUNCTIONHO.PY:

```
import numpy as np

from sklearn.neighbors import KNeighborsClassifier

# error rate

def error_rate(xtrain, ytrain, x, opts):

# parameters

k     = opts['k']

fold  = opts['fold']

xt    = fold['xt']

yt    = fold['yt']

xv    = fold['xv']

yv    = fold['yv']

# Number of instances

num_train = np.size(xt, 0)

num_valid = np.size(xv, 0)

# Define selected features

xtrain  = xt[:, x == 1]

ytrain  = yt.reshape(num_train)  # Solve bug

xvalid  = xv[:, x == 1]

yvalid  = yv.reshape(num_valid)  # Solve bug

# Training
```

```
mdl     = KNeighborsClassifier(n_neighbors = k)
mdl.fit(xtrain, ytrain)
# Prediction
ypred   = mdl.predict(xvalid)
acc     = np.sum(yvalid == ypred) / num_valid
error   = 1 - acc
return error
# Error rate & Feature size
def Fun(xtrain, ytrain, x, opts):
# Parameters
alpha    = 0.99
beta     = 1 - alpha
# Original feature size
max_feat = len(x)
# Number of selected features
num_feat = np.sum(x == 1)
# Solve if no feature selected
if num_feat == 0:
cost  = 1
else:
# Get error rate
```

```
error = error_rate(xtrain, ytrain, x, opts)

# Objective function

cost  = alpha * error + beta * (num_feat / max_feat)

return cost
```

6. TESTING

Testing is the process of finding differences between the expected behavior specified by system models and the observed behavior Testing is the process of executing a program with the intent of finding any errors. Testing is vital to the success of the system. Without proper testing hide errors will surface after sometime of use and perhaps irreversible damage has been done to valuable data. A series of tests like responsiveness, its value, stress and security are performed before the system is ready for user acceptance testing. System testing follows the logical conclusion that is all part of the system are tested and found to be working properly under all kinds of situations, and then the system is achieving its goal of processing the data perfectly according to user rules and requirements. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TESTING METHODOLOGIES:

Different levels of testing are used in the testing process, each level of testing aims to test different aspects of the system. The basic levels are

- Unit testing
- Integration testing
- System testing
- Acceptance testing

UNIT TESTING:

Unit testing focuses on the building blocks of the software system, that is, objects and sub system. There are three motivations behind focusing on components. First, unit testing reduces the complexity of the overall tests activities, allowing us to focus on smaller units of the system. Second, unit testing makes it easier to pinpoint and correct faults given that few components are

involved in the test. Third, Unit testing allows parallelism in the testing activities that is each component can be tested.

INTEGRATION TESTING:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

Top Down Integration:

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module which are incorporated into the structure in either a depth first or a breadth first manner. In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

Bottom-up Integration:

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.

SYSTEM TESTING:

In system testing the entire software is tested. The reference document for the process is the requirement document and the goal is to see whether the software meets its requirements. The system was tested for various test cases with various inputs.

ACCEPTANCE TESTING:

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

There are various forms of acceptance testing:

- User acceptance Testing
- Business acceptance Testing
- Alpha Testing
- Beta Testing

6.2 TEST CASES:

TEST CASE ID	TEST CASE	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Input dataset	Dataset given successful	Dataset not given	False negative
2	Input dataset	Dataset given successful	Dataset given successful	True positive
3	Dataset compilation	Successfully compiled	Errors occured	False negative
4	Dataset compilation	Successfully compiled	Successfully compiled	True positive
5	Runtime compilation	Output displayed	Output error	False negative
6	Runtime compilation	Output displayed	Output displayed	True positive

7. OUTPUT SCREEN:

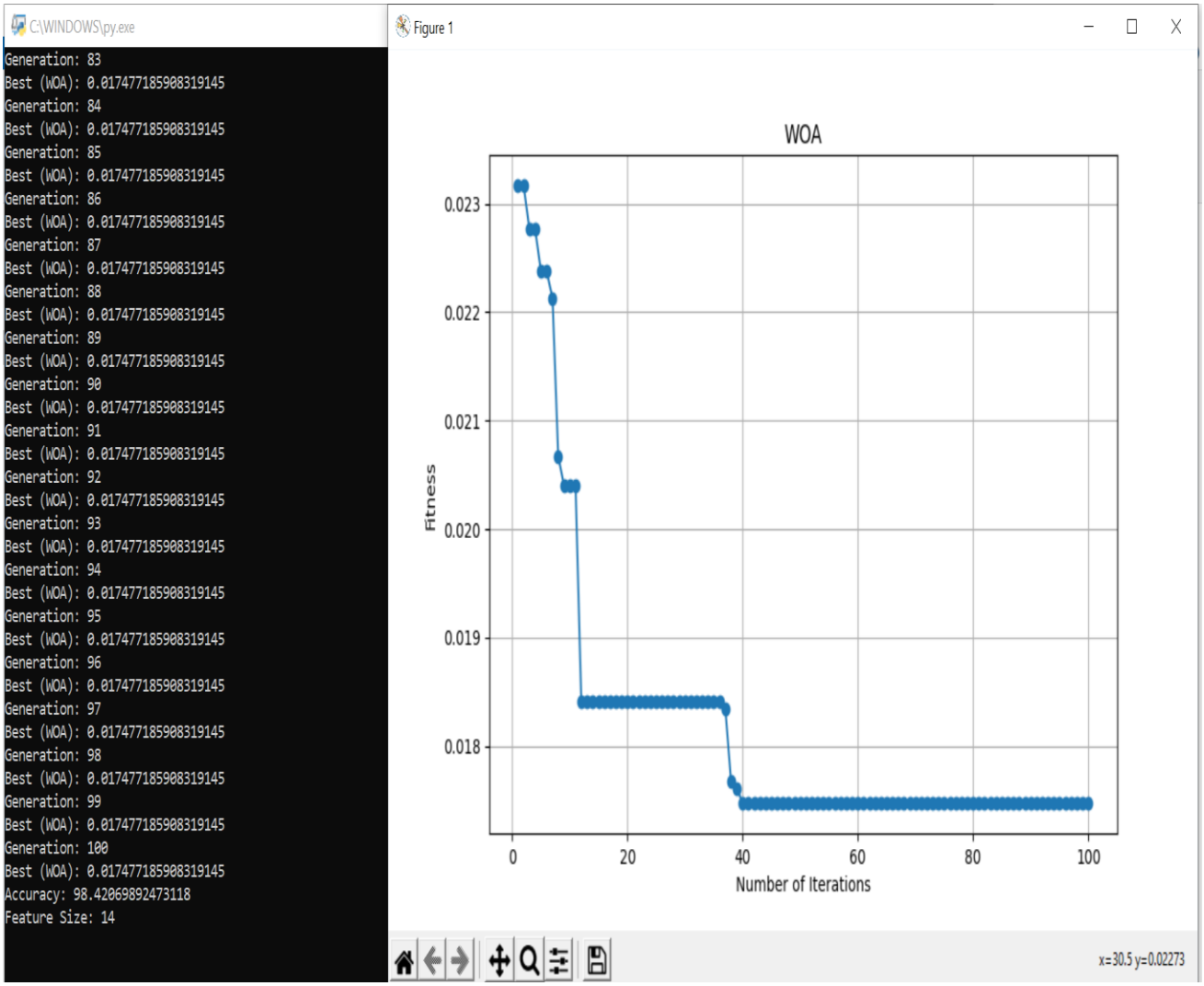

Fig:7.1 Graph showing the change of fitness values for number of iteration using WOA and K-NN.

8. CONCLUSION

In this project, we proposed a novel feature selection method using WOA and genetic operators for improving the performance of IDS. The proposed method selects the most informative features from the network data. The selected informative features help to improve the accuracy of the KNN-based IDS.

We evaluated the performance of the proposed method by using network-intrusion datasets. A comparison of the proposed method with the other feature selection algorithms on the basis of detection rate, execution time, and computational complexity proved the efficiency of the proposed method. The results clearly indicated that the inclusion of the crossover and mutation operators enhances the global search space of the whales and overcomes the problem of being stuck in the local optimum. The simulation results proved the suitability of the improved WOA-based feature selection method for the IDS.

9.FUTURE ENHANCEMENTS

The threat landscape is constantly changing, and now security vendors are focusing on high-fidelity machine learning, which uses algorithms to analyze files, and uses noise cancellation techniques like census and whitelist checking. As machine-learning grows, attackers will discover ways to get by them too, and IPS vendors will be off to create the next security antidote.

As we look toward the future, it's likely that more organizations will be looking at ways to future-proof their environments so they will have access to the latest technology and security professionals to discover, analyze and thwart the latest threats. The human factor will grow in all aspects of cyber security. More and more human analysis, incident-response teams and forensic crews will be involved in almost every company to complement the traditional intrusion-detection and intrusion-prevention systems. The automotive industry will need to incorporate defensive intrusion prevention and detection solutions into their cars, as the impact of attacks could be disastrous.

10. REFERENCES

[1] Ahmad, I.; Abdullah, A.; Alghamdi, A.; Alnfajan, K.; Hussain, M. Intrusion detection using feature subset selection based on MLP. Sci. Res. Essays 2011, 6, 6804–6810.

[2] Ghanem, W.; Jantan, A. Novel multi-objective artificial bee colony optimization for wrapper based feature selection in intrusion detection. Int. J. Adv. Soft Comput. Appl. 2016, 8, 70–81.

[3] Zaman, S.; El-Abed, M.; Karray, F. Features selection approaches for intrusion detection systems based on evolution algorithms. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, Kota Kinabalu, Malaysia, 17–19 January 2013; pp. 1–5.

[4] Chung, Y.Y.; Wahid, N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). Appl. Soft Comput. 2012, 12, 3014–3022.

[5] Syarif, I. Feature selection of network intrusion data using genetic algorithm and particle swarm optimization. EMITTER Int. J. Eng. Technol. 2016, 4, 277–290.

[6] Devi, E.M.; Suganthe, R.C. Feature selection in intrusion detection grey wolf optimizer. Asian J. Res. Soc. Sci. Humanit. 2017, 7, 671–682.

[7] Selvakumar, B.; Muneeswaran, K. Firefly algorithm based feature selection for network intrusion detection. Comput. Secur. 2019, 81, 148–155.

[8] S. Mirjalili and A. Lewis, "The whale optimization algorithm", *Adv. Eng. Softw.*, vol. 95, pp. 51-67, May 2016.

[9] D. E. Goldberg, Genetic algorithms in Search Optimization and Machine Learning, Reading, MA, USA:Addison-Wesley, 1989.

[10] B. Mukherjee, L. T. Heberlein and K. N. Levitt, "Network intrusion detection", *IEEE Netw.*, vol. 8, no. 3, pp. 26-41, May 1994.

[11] Chen, Y., Li, Y., Cheng, X., and Guo, L.:.Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection System. Information Security and Cryptology, Lecture notes in Computer science, 4318, pp. 153-167, (2006).

[12]Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, P., and Kannan, A.:Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. Journal on Wireless Communications and Networking, pp. 1-16, (2013).

[13]Gne Kayack, H., and Zincir-Heywood, N.: Analysis of Three Intrusion Detection System Benchmark Datasets Using Machine Learning Algorithms. Proceedings of IEEE international conference on Intelligence and Security Informatics, pp. 362- 367, 2005.

[14]Gnes Kayack, H., Nur Zincir-Heywood, A., and Heywood, M. I.: Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets. Third Annual Conference on Privacy, Security and Trust, (2005).

[15]Jalill, K. A., Kamarudin, M. H., and Masrek, M.N.: Comparison of Machine Learning Algorithms Performance in Detecting Network Intrusion. International Conference on Networking and Information Technology, pp. 221-226, (2010).