IMPROVED DETECTION AND PREVENTION MECHANISMS FOR DDOS ATTACK IN CLOUD COMPUTING ENVIRONMENT

A PROJECT REPORT

Submitted by

AATHISH B	**(18CSR003)**
DHARSHAN G	**(18CSR019)**
KAVINKUMAR M	**(18CSR043)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

ERODE - 638 012

MAY 2022

BONAFIDE CERTIFICATE

Certified that this project report **"IMPROVED DETECTION AND PREVENTION MECHANISMS FOR DDOS ATTACK IN CLOUD COMPUTING ENVIRONMENT"** is the bonafide work of **"AATHISH B (18CSR003) & DHARSHAN G (18CSR019) & KAVIN KUMAR M (18CSR043)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Ms.M.Mohanasundari.,M.E.,	Dr.S.Jabeen Begum,M.E.,PhD.,
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Assistant Professor	Professor,
Department of CSE,	Department of CSE,
Velalar College of Engineering and Technology,	Velalar College of Engineering and Technology,
Erode-638 012	Erode-638 012

Submitted for Semester Project viva-voce examination held on _________

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We are greatly indebted to many people who have contributed to the progress of our project and have encouraged us right from the start to make our ideas a reality.

We are very privileged to express our heartfelt thanks to our honorable Secretary and Correspondent **Thiru. S. D .Chandrasekar, B.A.,** who provided all the facilities to build our project.

We express our deep sense of gratitude and sincere thanks to **Dr. M. Jayaraman, M.E., Ph.D., F I E** Principal, VCET for permitting us to undertake this project.

We are indebted to **Prof. P. Jayachandar, M.E.,** Dean, Velalar College of Engineering and Technology for giving us the opportunity and continuous inspiration to carry out the project.

We express our profound gratitude to **Dr. S. Jabeen Begum, M.E., Ph.D**., Head, Department of Computer Science and Engineering for her constant encouragement.

Our sincere thanks to **Ms. M. Mohanasundari, M.E.,** Assistant Professor for her valuable guidance, advice and help rendered whenever we approached her in times of need

We express our sincere thanks to our Project Coordinator **Ms. S. Subathra, M.E**., Assistant Professor (Sr.Gr.), for the support and guidance in all aspects throughout the work.

We express our sincere thanks to the teaching and non-teaching faculty of the Computer Science and Engineering Department and other Departments of **VCET** for their valuable suggestions to make our project successful.

We express our sincere thanks to our parents for their support to do our project successfully.

ABSTRACT

The massive diverse network which is connected over the cloud computing network produces various traffic patterns. These patterns are analyzed and identify the DDoS attacks based on their volume and the anomaly. Due to the cloud computing's widely distributed nature many challenges occur such as privacy, security and availability due to its dynamic resource provisioning based on the on-demand requests from cloud users. Among these various issues one of the important issues in cloud computing paradigm is DDoS attack. Distributed Denial of Service (DDoS) attack is a sub class of Denial of Service (DoS) attack. A DDoS attack involves multiple connected online devices, collectively known as a botnet, which are used to overwhelm a target website or server by increasing its traffic, thus denying the service to the users. This paper put in plain words the DDoS attack, its detection as well as prevention mechanisms in cloud computing environment. The inclusive study also explains about the effects of DDoS attack on cloud platform and the related defense mechanisms required to be considered. The purpose of this paper is to achieve data security of cloud storage and to formulate corresponding cloud storage security policy.

TABLE OF CONTENTS

CHAPTER NO.	CONTENTS	PAGE NO.
	ABSTRACT	iv
	LIST OF FIGURES	vi
	LIST OF ABBREVIATIONS	vii
1	**INTRODUCTION**	1
	1.1 PROBLEM STATEMENT	1
	1.2 OBJECTIVE	2
2	**SYSTEM SPECIFICATION**	3
	2.1 HARDWARE SPECIFICATION	3
	2.2 SOFTWARE SPECIFICATION	3
3	**TOOL DESCRIPTION**	4
	3.1 NETBEANS IDE	4
	3.2 WEKA	6
4	**SOFTWARE DESCRIPTION**	9
	4.1 JAVA	9
5	**PROJECT DESCRIPTION**	12
	5.1 OVERVIEW OF THE PROJECT	12
	5.2 MODULE DESCRIPTION	12
6	**SYSTEM TESTING**	14
	6.1 UNIT TESTING	14
	6.2 INTRGRATION TESTING	15
	6.3 USER ACCEPTANCE TESTING	15
7	**SYSTEM IMPLEMENTATION**	16
8	**CONCLUSION**	18
	8.1 FUTURE ENHANCEMENTS	18
9	**APPENDICES**	19
	9.1 SOURCE CODE	19
	9.2 SCREEN SHOTS	40
10	**REFERENCE**	42
	CERTIFICATES	44

LIST OF FIGURES

CHAPTER NO.	CONTENTS	PAGE NO.
7	7.1 FLOW CHART OF THE PROJECT	17

LIST OF ABBREVATIONS

AI	Artificial Intelligence
CGAN	Conditional Generative Adversarial Network
DDOS	Distributed Denial of Service
IDE	Integrated Development Environment
IT	Information Technology
JDK	Java Platform, Standard Edition Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
KDD	Knowledge Discovery From Data
LSTM	Long Short-Term Memory
ML	Machine Learning
PART	Partial Decision Tree Algorithm
RF	Random Forest
WEKA	Waikato Environment for Knowledge Analysis

CHAPTER 1

INTRODUCTION

There are many challenges that exist in cloud computing such as privacy, security and availability due to its dynamic resource provisioning based on the on-demand requests from cloud users. For effective use of cloud resources and to reduce the latency of cloud users, the cloud computing model extends the services such as networking facilities, computational capabilities and storage facilities based on the said on-demand requests. So, due to this widely distributed nature many challenges occur such as privacy, security and availability due to its dynamic resource provisioning based on the on-demand requests from cloud users.

Cloud computing environment depends on user services such as high-speed storage and retrieval provided by cloud computing system. Meanwhile data security is an important problem to solve urgently for cloud storage technology. Among these various issues one of the important issues in cloud computing paradigm is DDoS attack. There are many mechanisms available for handling these security and privacy issues in cloud computing. But these approaches cannot be directly applied to fog computing due to a mixture of features that treat it different form cloud.

Distributed computing is a mechanical development that offers the offices, stage, and programming of data innovation as Internet administrations. It is viewed as the transformation of a durable dream called computing for Use," and it is, in effect, step by step embraced by associations as private, public, or mixture Clouds. Its fundamental goal is to allow clients to utilize and pay for what they need, promising on-request benefits for their product or framework needs. Although Cloud registering is viewed as a significant and positive IT foundation shift, much security work is as yet expected to limit its deficiencies. Since a significant measure of individual and corporate data is put in the Cloud server farms, those Cloud security issues and weaknesses should be identified and forestalled. The partner editorial manager organizing the audit of this original copy and endorsing it for distribution was Ines Domingue

1.1 PROBLEM STATEMENT

A denial-of-service condition is accomplished by flooding the targeted host or network with traffic until the target cannot respond or simply crashes, preventing access for legitimate users. DoS attacks can cost an organization both time and money while their resources and services are inaccessible.

1.2 OBJECTIVE

Due to the diverse network of the cloud computing many challenges occur come to the light every day. The issues reflect in privacy, storage and availability of services. One of this challenge is a malicious attack called DDOS. So, our main objective of the project is to detect the DDoS attack on the cloud computing environment effectively in less time with C.45 algorithm.

The anomaly detected by the machine learning algorithm powered model in the internet traffic will be alerted.

CHAPTER- 2

SYSTEM SPECIFICATION

2.1 HARDWARE REQUIREMENTS

This section gives the detail and specification of the hardware on which the system is expected to work.

• Processor Type	`:	Pentium i3
• Speed	:	3.40GHZ
• RAM	:	8GB DD4 RAM
• Hard disk	:	500 GB
• Keyboard	:	101/102 Standard Keys
• Mouse	:	Optical Mouse

2.2 SOFTWARE REQUIREMENTS

This section gives the detail and specification of the software on which the system is expected to work.

• Operating System	:	Windows 10
• Frontend	:	NetBeans
• Coding Language	:	java
• Tools	:	KDD data set

CHAPTER-3

TOOL DESCRIPTION

3.1 NetBeans IDE

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++. The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems.

The IDE provides comprehensive support for JDK 7 technologies and the most recent Java enhancements. It is the first IDE that provides support for JDK 7, Java EE 7, and JavaFX 2. The IDE fully supports Java EE using the latest standards for Java, XML, Web services, and SQL and fully supports the Glassfish Server, the reference implementation of Java EE.

It is also the official Java 8 IDE that is packed with powerful converters, editors, and code analyzers. The system integrates with the widely used Find Bugs tool that enables users to quickly identify and solve common code problems.

NetBeans features

- Project management
- Visual debugger
- Static analysis tools
- Code converters
- NetBeans Profiler
- Batch code analysers
- Cross-platform support
- Multiple language support
- Code editor

Main Components of Java NetBeans IDE

Before somebody starts developing the applications, make sure that JAVA SDK along with NetBeans IDE is installed on our computer. After downloading you can install the software. Once the software is installed, now let us go through all 5 components after installation. After installation of the software, we can start working on it by following these steps as follows

1. Start
2. All Programs
3. NetBeans
4. NetBeans IDE

Initially, wait for a while till the program (NetBeans IDE) window opens in the fully functional mode. It consists of many features & by using it we are able to prepare all type of programs (Desktop Application & Web Application) and games also.

Let us discuss the components one by one which are as follows:

1. Title bar
2. Menu bar and pull-down menus
3. Toolbar
4. GUI building
5. Palette

Key Features of NetBeans IDE

For Java development, you can use NetBeans to develop almost any kind of Java applications, from Java desktop apps (Swing & JavaFX) to Java web apps (Servlet, JSP, and EJB). However, NetBeans doesn’t support developing Android mobile apps - You have to use Android Studio from Google. Before Android becomes popular, NetBeans has great support for developing mobile applications with JavaME. But since Android has dominated the market, JavaME became obsolete.

For Swing development, the GUI Builder helps you visually design layout and drag and drop user interface components. Likewise, the Scene Builder helps you design JavaFX applications quickly and easily. For developing Java web applications, NetBeans supports two types of servers: Glassfish and Apache Tomcat. By default, NetBeans uses Ant build for projects, and you can also create Maven ones. For version control, NetBeans is integrated with CVS, Subversion (SVN) and Mercurial. The IDE also comes with powerful refactoring capability which facilitates code restructuring. The visual debugger helps you debug running Java applications, and the Profiler tool helps you monitor Java applications to find memory leaks and optimize speed. For unit testing, NetBeans is integrated with JUnit and TestNG frameworks.

Pros :

- Good for Collaboration
- Open Source
- Good for Debugging and Testing Code

Cons :

- Have less add-ons.
- The software occupies lots of space and works very slow on computers having i3 processor
- NetBeans provide only one view unlike Eclipse.

3.2 Weka

Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization. Weka is an open-source software issued under the GNU General Public License.

All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining.

Weka’s Features

Weka features include

- Machine learning,
- Data mining,
- Pre-processing,
- Classification,
- Regression,
- Clustering,
- Association rules,
- Attribute selection,
- Experiments,
- Workflow,
- Visualization.

Components of Weka

Weka's main user interface is the Explorer, the same functionality also can be accessed through the component-based Knowledge Flow interface and from the command line. There is also the Experimenter, which allows the systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of datasets. The Explorer interface features several panels providing access to the main components of the workbench such as pre-process panel which facilities for importing data, classify panel enables the user to apply classification and regression algorithms, associate panel provides access to association rule learners cluster panel gives access to the clustering techniques, select attributes panel provides algorithms for identifying the most predictive attributes in a dataset, and visualize panel shows a scatter plot matrix.

Weka provides comprehensive set of data pre-processing tools, learning algorithms and evaluation methods, graphical user interfaces and an environment for comparing learning algorithms. The data can be imported from a file in various formats such as ARFF, CSV, C4.5, binary. Data can also be read from a URL or from an SQL database (using JDBC). Pre-processing tools in WEKA are called "filters" and there are filters available for Discretization, normalization, resampling, attribute selection, transforming and combining attributes.

The implemented learning schemes are decision trees and lists, instance-based classifiers, support vector machines, multi-layer perceptron’s, logistic regression, Bayes' nets. The meta classifiers included are bagging, boosting, stacking, error-correcting output codes, locally weighted learning. The implemented schemes are k-Means, EM, Cobweb, X-means, Farthest First. The Clusters can be visualized and compared to "true" clusters. Apriori can compute all rules that have a given minimum support and exceed a given confidence. In Weka, data sources, classifiers, etc. are beans and can be connected graphically.

Pros :

- It is an open-source software and platform friendly,
- It uses the GNU GPL version 2.0 or 3.0 depending on the version of WEKA installed, This means that the software's source code is available for review and that the software may be used freely and for any purpose.

Cons:

- Weka is memory intensive,
- While WEKA is open source, using it for commercial applications which require distributing the WEKA code as part of the application may require you to purchase licenses from one or more corporate entities,
- When you load a dataset into the WEKA Explorer (the graphical user interface), For very large datasets, Fortunately, this can be circumvented by using the Command-line interface (CLI) for loading and processing the large dataset.

CHAPTER- 4

SOFTWARE DESCRIPTION

4.1 JAVA

Java is a simple and yet powerful object-oriented programming language and it is in many respects similar to C++. Java originated at Sun Microsystems, Inc. in 1991. It was conceived by James Gosling, Patrick Naughton, Chris Wrath, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. It was developed to provide a platform-independent programming language.

Platform Independent

Unlike many other programming languages including C and C++ when Java is compiled, it is not compiled into platform *specific machine*, rather into platform independent byte code. This byte code is distributed over the web and interpreted by virtual Machine (JVM) on whichever platform it is being run.

Java Virtual Machine

Java was designed with a concept of ‘write once and run everywhere’. Java Virtual Machine plays the central role in this concept. The JVM is the environment in which Java programs execute. It is software that is implemented on top of real hardware and operating system. When the source code (.java files) is compiled, it is translated into byte codes and then placed into (.class) files. The JVM executes these bytecodes. So Java byte codes can be thought of as the machine language of the JVM. A JVM can either interpret the bytecode one instruction at a time or the bytecode can be compiled further for the real microprocessor using what is called a just-in-time compiler. The JVM must be implemented on a particular platform before compiled programs can run on that platform.

Object Oriented Programming

Java is an object oriented programming language it has following features:

- Reusability of Code
- Emphasis on data rather than procedure
- Data is hidden and cannot be accessed by external functions
- Objects can communicate with each other through functions

Object Oriented Programming is a method of implementation in which programs are organized as cooperative collection of objects, each of which represents an instance of a class, and whose classes are all members of a hierarchy of classes united via inheritance relationships.

OOP Concepts

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

Abstraction

Abstraction denotes essential characteristics of object that distinguish it from other kinds of objects, thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer.

Encapsulation

Encapsulation is the process of compartmentalizing the elements of an abstraction that constitute its structure and behavior ; encapsulation serves to separate the contractual interface of an abstraction and its implementation.

Inheritance

Inheritance is the process by which one object acquires the properties of another object.

Polymorphism

Polymorphism is the existence of the classes or methods in different forms or single name denoting different implementations.

Java is Distributed

With extensive set of routines to handle TCP/IP protocols like HTTP and FTP java can open and access the objects across net via URLs.

Java is Multithreaded

One of the powerful aspects of the Java language is that it allows multiple threads of execution to run concurrently within the same program A single Java program can have many different threads executing independently and continuously. Multiple Java applets can run on the browser at the same time sharing the CPU time.

Java is Secure

Java was designed to allow secure execution of code across network. To make Java secure many of the features of C and C++ were eliminated. Java does not use Pointers. Java programs cannot access arbitrary addresses in memory.

Garbage collection

Automatic garbage collection is another great feature of Java with which it prevents inadvertent corruption of memory. Similar to C++, Java has a new operator to allocate memory on the heap for a new object. But it does not use delete operator to free the memory as it is done in C++ to free the memory if the object is no longer needed. It is done automatically with garbage collector.

CHAPTER-5

PROJECT DESCRIPTION

5.1 OVERVIEW OF THE PROJECT

The main objective of the project is to detect the DDoS attack on the cloud computing environment effectively in less time with C.45 algorithm which will detect the anomaly on the server traffic and alert the admin and finally the results will be compared with other machine learning algorithms. The accuracy evaluated and the result will show its effectiveness compared to other algorithms.

5.2 MODULE DESCRIPTION

- Upload Files
- Node and Request
- Server Give Response
- DDOS Attack Detection

Upload Files

In computer networks, to upload can refer to the sending of data from a local system to a remote system such as a server or another client with the intent that the remote system should store a copy of the data being transferred, or the initiation of such a process. Where the connection to the remote computers is via a dial-up connection, the transfer time required to download locally and then upload again could increase from seconds to hours or days. A Server creates its own database then uploads its files.

Node Send Request

Often Nodes and servers communicate over a computer network on separate hardware, but both node and server may reside in the same system. A server host runs one or more server programs which share their resources with nodes. A node does not share any of its resources but requests a server's content or service function. Nodes therefore initiate communication sessions with servers which await incoming requests. The server component provides a function or service to one or many nodes, which initiate requests for such services.

A web server serves web pages, and a file server serves computer files. A shared resource may be any of the server computer's software and electronic components, from programs and data to processors and storage devices.

Server Give Response

Clients and servers exchange messages in a request-response messaging pattern. The client sends a request, and the server returns a response. This exchange of messages is an example of inter-process communication. To communicate, the computers must have a common language, and they must follow rules so that both the client and the server know what to expect. The language and rules of communication are defined in a communications protocol. Here the server extracts from its database and match the node request then send this particular file to node.

DDOS Attack Detection

In this module attacker intrudes server's response and generate huge number of fake packets to the victim. Then it can forward this fake packet through its attacking agents. These attacking agents may be zombies or puppets or root zombies. These attacking agents again forward this fake packet to victim through router. Here router act as BW-DDOS attack detector. First router checks packets size. If huge number of packets are detected it will give the defense against this BW-DDOS attacks. Else it will forward to the requested node. This router using four types of defense mechanisms: filtering, rate limiting, detouring, and absorbing, and breakthrough. It applies any one defense mechanism and get original packets. Finally, our router will forward the original packet to the requested node.

CHAPTER – 6

SYSTEM TESTING

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. In the testing process, first test the actual system in an organization and solve the gathered error and ensure that the portal will work in the new system tin full efficiency as stated.

System testing is the stage of implementation, which is aimed to ensure that the system works accurately and efficiently. In the testing process, it tests the actual system in an organization and gather errors from the new system and take initiatives to correct the error. All the front-end and back-end connectivity are tested to be sure that the portal will work in the new system full efficiency as stated. Inadequate testing or non-testing leads to errors, which may appear a few months later. This will create two problems:

- Time delay between the cause and appearance of the problem.
- There may be an error on files and records within the system.
- The purpose of the system testing is to consider all the likely variations to which sit will be suggested and push the system to its limits.

6.1 UNIT TESTING

Unit testing, also known as component testing, verifies the functionality of a specific section of code and usually the function level. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In an object-oriented environment, this is usually at the class level. The minimal unit tests include the constructors and destructor. These types of tests are usually written by developers as it works on code, to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner 24 cases or other branches in the code.

Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks in the software work independently.

6.2 INTEGRATION TESTING

Integration testing is a type of software testing that seeks to verify the interfaces between components against software design. Software components may be integrated in an iterative way or altogether. Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as Integration and Testing, 'String Testing' and sometimes 'Thread Testing'. Progressively, larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

6.3 USER ACCEPTANCE TESTING

Once a program or module has been unit tested, then the programmer will work to integrate it with other programs. Integration testing is a systematic technique for constructing tests to uncover errors associated within the interface. In this project, all the modules are combined and then the entire programs are tested.

The main Purpose of User Acceptance Testing is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. This testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

CHAPTER 7

SYSTEM IMPLEMENTATION

Implementation is the process of converting a new or revised system design into an operational one when the initial design was done by the system; a demonstration was given to the end user about the working system. This process is used to verify and identify any logical mess working of the system by feeding various combinations of test data.

Problem Definition

A denial-of-service condition is accomplished by flooding the targeted host or network with traffic until the target cannot respond or simply crashes, preventing access for legitimate users. DoS attacks can cost an organization both time and money while their resources and services are inaccessible

Problem Definition

Equip your network, applications, and infrastructure with multi-level protection strategies. This may include prevention management systems that combine firewalls, VPN, anti-spam, content filtering and other security layers to monitor activities and identity traffic inconsistencies that may be symptoms of DDoS attacks.

Proposed Methodology

By employing C.4.5 algorithm based on machine learning approach DDoS attacks can be efficiently handled in the cloud computing environment. When flooded traffic found in network then immediately it is been analyzed. The attacks will be identified, and immediate response will be taken to drop the malicious traffic. To adapt to these patterns, machine learning algorithms are used by the cloud networks. This algorithm makes use of various techniques based on signature which leads to the formation of decision tree to identify the attacks automatically.

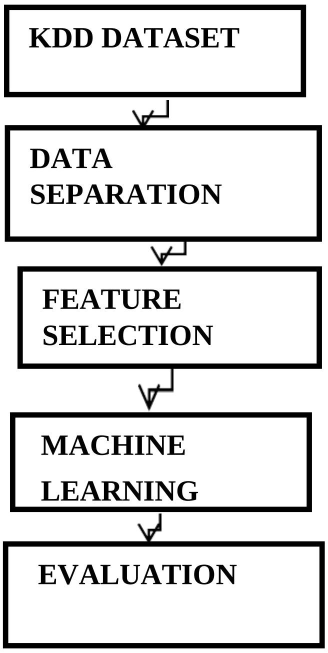


fig 7.1 flowchart of the project

CHAPTER – 8
CONCLUSION

We carried out a systematic literature review to analyze ML techniques used in Cloud security. The review investigated relevant studies that answered 3 RQs; Cloud security area, type of ML techniques used, and the accurate estimation of the ML model. Overall, our conclusions are summarized as follows: RQ1 findings are the 11 Cloud security areas identified; anomaly detection, attack detection, privacy preservation, security, vulnerability detection, the confidentiality of data, data privacy, DDoS, DoS, and intrusion detection (ID). DDoS and data privacy are analyzed the most, with a 16% frequency of usage and 14% respectively. RQ2 counted 30 ML techniques used, some used as hybrid and others as standalone. Identification of correctly classified instances and incorrectly classified instance are all segmented and the accuracy for each segmentation is identified.

8.1 FUTURE ENHANCEMENTS

Future work can be extended to propose a new efficient DDoS prevention mechanism to handle the attacks using the Deep Learning techniques which would be an effective implementation of both detection and prevention measures.

CHAPTER - 9

APPENDICES

9.1 SOURCE CODE

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package aodebasedids;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.Random;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import weka.attributeSelection.AttributeSelection;
import weka.attributeSelection.CfsSubsetEval;
import weka.attributeSelection.GreedyStepwise;
import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.classifiers.bayes.AODE;
import weka.classifiers.evaluation.NominalPrediction;
import weka.classifiers.meta.AttributeSelectedClassifier;
import weka.classifiers.rules.DecisionTable;
import weka.classifiers.rules.PART;
import weka.classifiers.trees.DecisionStump;
import weka.classifiers.trees.J48;
import weka.core.FastVector;
import weka.core.Instances;
import weka.core.Utils;
import weka.core.converters.ArffSaver;
import weka.core.converters.CSVLoader;
import weka.core.converters.ConverterUtils.DataSource;
import weka.filters.Filter;
/**
 *
 * @author Elcot
 */
public class AODEFrame extends javax.swing.JFrame {
    /**
     * Creates new form AODEFrame
     */
    String fn="";
    String fn5="";
    String fn4="";
ArrayList duration=new ArrayList();
ArrayListsrcbytes=new ArrayList();
ArrayListdestbytes=new ArrayList();
ArrayList val1=new ArrayList();
ArrayList val2=new ArrayList();
ArrayList val3=new ArrayList();
ArrayList nominal=new ArrayList();
ArrayListnominalval=new ArrayList();
    public AODEFrame() {
initComponents();
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jTabbedPane1 = new javax.swing.JTabbedPane();
        jPanel2 = new javax.swing.JPanel();
        jButton1 = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jScrollPane3 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jPanel3 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea2 = new javax.swing.JTextArea();
        jPanel5 = new javax.swing.JPanel();
        jButton2 = new javax.swing.JButton();
        jTextField2 = new javax.swing.JTextField();
        jButton3 = new javax.swing.JButton();
        jTextField3 = new javax.swing.JTextField();
        jButton4 = new javax.swing.JButton();
        jTextField4 = new javax.swing.JTextField();
        jButton5 = new javax.swing.JButton();
        jTextField5 = new javax.swing.JTextField();
        jPanel4 = new javax.swing.JPanel();
        jButton6 = new javax.swing.JButton();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTextArea3 = new javax.swing.JTextArea();
        jLabel2 = new javax.swing.JLabel();
        jTextField6 = new javax.swing.JTextField();
        jButton8 = new javax.swing.JButton();
        jPanel6 = new javax.swing.JPanel();
        jButton7 = new javax.swing.JButton();
        jScrollPane4 = new javax.swing.JScrollPane();
        jTextArea4 = new javax.swing.JTextArea();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Algerian", 0, 24)); // NOI18N
        jLabel1.setText(" DDOS DETECTION");

        jButton1.setText("Browse");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel3.setText("File path");

        jTextArea1.setColumns(20);
        jTextArea1.setRows(5);
        jScrollPane3.setViewportView(jTextArea1);

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel2Layout.createSequentialGroup()
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel2Layout.createSequentialGroup()
.addGap(20, 20, 20)
.addComponent(jLabel3)
.addGap(18, 18, 18)
.addComponent(jTextField1,              javax.swing.GroupLayout.PREFERRED_SIZE,              553,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(jPanel2Layout.createSequentialGroup()
.addGap(78, 78, 78)
.addComponent(jScrollPane3,             javax.swing.GroupLayout.PREFERRED_SIZE,             516,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(jPanel2Layout.createSequentialGroup()
.addGap(301, 301, 301)
.addComponent(jButton1,                 javax.swing.GroupLayout.PREFERRED_SIZE,               87,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addContainerGap(55, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(
          jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel2Layout.createSequentialGroup()
.addGap(24, 24, 24)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel3)
.addComponent(jTextField1,                    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addComponent(jButton1)
.addGap(15, 15, 15)
.addComponent(jScrollPane3,         javax.swing.GroupLayout.PREFERRED_SIZE,         204,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(33, Short.MAX_VALUE))
        );

        jTabbedPane1.addTab("NSL KDD Dataset", jPanel2);

        jTextArea2.setColumns(20);
        jTextArea2.setRows(5);
        jScrollPane1.setViewportView(jTextArea2);

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel3Layout.createSequentialGroup()
.addGap(52, 52, 52)
.addComponent(jScrollPane1,         javax.swing.GroupLayout.PREFERRED_SIZE,         575,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(60, Short.MAX_VALUE))
        );
        jPanel3Layout.setVerticalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel3Layout.createSequentialGroup()
.addGap(43, 43, 43)
.addComponent(jScrollPane1,         javax.swing.GroupLayout.PREFERRED_SIZE,         250,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(44, Short.MAX_VALUE))
        );

        jTabbedPane1.addTab("Numeric to Nominal Conversion", jPanel3);

        jButton2.setText("Feature Selection");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jButton2ActionPerformed(evt);
            }
        });

        jButton3.setText("Information Gain");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jButton3ActionPerformed(evt);
            }
        });

        jButton4.setText("Intrinsic Information");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jButton4ActionPerformed(evt);
            }
        });

        jButton5.setText("Gain Ratio");
        jButton5.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jButton5ActionPerformed(evt);
            }
        });

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
        jPanel5.setLayout(jPanel5Layout);
        jPanel5Layout.setHorizontalGroup(
            jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel5Layout.createSequentialGroup()
.addGap(153, 153, 153)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton2)
.addComponent(jButton3)
.addComponent(jButton4))
.addGap(61, 61, 61)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE, 248,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, 248,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
.addComponent(jTextField3, javax.swing.GroupLayout.DEFAULT_SIZE, 248,
Short.MAX_VALUE)
.addComponent(jTextField2, javax.swing.GroupLayout.DEFAULT_SIZE, 248,
Short.MAX_VALUE)))
.addContainerGap(96, Short.MAX_VALUE))
        );
        jPanel5Layout.setVerticalGroup(
            jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel5Layout.createSequentialGroup()
.addGap(81, 81, 81)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton2)
.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton3)
.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton4)
.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton5)
.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addContainerGap(110, Short.MAX_VALUE))
        );

        jTabbedPane1.addTab("Future Selection", jPanel5);

        jButton6.setText("Average One Dependence Estimator Classifier");
        jButton6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jButton6ActionPerformed(evt);
            }
        });
jTextArea3.setColumns(20);
        jTextArea3.setRows(5);
        jScrollPane2.setViewportView(jTextArea3);

        jLabel2.setText("Testing Data set");

        jTextField6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jTextField6ActionPerformed(evt);
            }
        });

        jButton8.setText("Browse");
        jButton8.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jButton8ActionPerformed(evt);
            }
        });

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
```

```
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createSequentialGroup()
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addGroup(jPanel4Layout.createSequentialGroup()
.addContainerGap()
.addComponent(jScrollPane2,            javax.swing.GroupLayout.PREFERRED_SIZE,            549,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel4Layout.createSequentialGroup()
.addGap(40, 40, 40)
.addComponent(jLabel2)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jTextField6,            javax.swing.GroupLayout.PREFERRED_SIZE,            401,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jButton8)))
.addGroup(jPanel4Layout.createSequentialGroup()
.addGap(195, 195, 195)
.addComponent(jButton6)))
.addContainerGap(86, Short.MAX_VALUE))
        );
        jPanel4Layout.setVerticalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createSequentialGroup()
.addGap(20, 20, 20)
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jTextField6,            javax.swing.GroupLayout.PREFERRED_SIZE,            28,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel2)
.addComponent(jButton8))
.addGap(18, 18, 18)
.addComponent(jButton6)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jScrollPane2,            javax.swing.GroupLayout.DEFAULT_SIZE,            237,
Short.MAX_VALUE))
        );

        jTabbedPane1.addTab("Machine Learning", jPanel4);

        jButton7.setText("Evaluation");
        jButton7.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEventevt) {
                jButton7ActionPerformed(evt);
            }
        });

        jTextArea4.setColumns(20);
```

```
    jTextArea4.setRows(5);
    jScrollPane4.setViewportView(jTextArea4);

javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
    jPanel6.setLayout(jPanel6Layout);
    jPanel6Layout.setHorizontalGroup(
      jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel6Layout.createSequentialGroup()
.addGap(280, 280, 280)
.addComponent(jButton7)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel6Layout.createSequentialGroup()
.addContainerGap(38, Short.MAX_VALUE)
.addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 608,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(41, 41, 41))
    );
    jPanel6Layout.setVerticalGroup(
      jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel6Layout.createSequentialGroup()
.addGap(27, 27, 27)
.addComponent(jButton7)
.addGap(18, 18, 18)
.addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 239,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(26, Short.MAX_VALUE))
    );

    jTabbedPane1.addTab("Evaluation", jPanel6);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addGap(83, 83, 83)
.addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 692,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(jPanel1Layout.createSequentialGroup()
.addGap(341, 341, 341)
.addComponent(jLabel1)))
.addContainerGap(74, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap(42, Short.MAX_VALUE)
```

```
.addComponent(jLabel1)
.addGap(31, 31, 31)
.addComponent(jTabbedPane1,           javax.swing.GroupLayout.PREFERRED_SIZE,           365,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap())
        );

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
.addContainerGap()
.addComponent(jPanel1,                                  javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addContainerGap())
        );
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
.addGap(0, 0, Short.MAX_VALUE)
.addComponent(jPanel1,                              javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );

pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:
        try
        {
JFileChooser fc=new JFileChooser();
            int result=fc.showOpenDialog(this);
            if (result == JFileChooser.APPROVE_OPTION)
            {
                File file=fc.getSelectedFile();
                jTextField1.setText(file.getAbsolutePath());
                //pa=file.getAbsolutePath();
                //fa=file.getName();

                String fname=file.getAbsolutePath();
                File fe=new File(fname);
FileInputStreamfis=new FileInputStream(fe);
                byte data[]=new byte[fis.available()];
fis.read(data);
fis.close();
                String str=new String(data);
                jTextArea1.setText(str);
System.out.println(str);
```

```
                String s[]=str.split("\n");
                String s1[]=s[0].split(",");
                String k1="";
for(int k=0;k<s1.length;k++)
                {
                    k1=k1+s1[k]+",";
                }
                String k2=k1.substring(0,k1.lastIndexOf(','));
nominalval.add(k2);

for(int i=1;i<s.length;i++)
                {
                    String s2[]=s[i].split(",");
                    //dm.addRow(s2);
duration.add(s2[0]);
srcbytes.add(s2[4]);
destbytes.add(s2[5]);
                }

                int h=0;
for(int g=0;g<srcbytes.size();g++)
                {
                    String g1=srcbytes.get(g).toString();
                    h=h+Integer.parseInt(g1);
                }
                int n=h/srcbytes.size();

                int h1=0;
for(int g=0;g<destbytes.size();g++)
                {
                String g1=destbytes.get(g).toString();
                h1=h1+Integer.parseInt(g1);
                }
                int n1=h1/destbytes.size();
                //String nominal=str;
for(int i=1;i<s.length;i++)
                {
                    //Vector v=new Vector();
                    String s2[]=s[i].split(",");
for(int j=0;j<s2.length;j++)
                    {
                        if(j==0)
                        {
                           //nominal.add(s2[0]+"#period1");
System.out.println("-----------------------");
System.out.println("s2[0] is "+s2[0]);
                            if(s2[0].equals("0"))
                            {
                                s2[0]="period 1";
nominal.add(s2[0]);
String st=nominal.get(j).toString();
                        st1=st1+st+",";
                    }
                    nominal=new ArrayList();
                    String st2=st1.substring(0,st1.lastIndexOf(','));
System.out.println("st2 is "+st2);
nominalval.add(st2);
                }
                String st3="";
for(int i=0;i<nominalval.size();i++)
                {
                    String nom=nominalval.get(i).toString();
                    st3=st3+nom+"\n";
                }
                String st4=st3.substring(0,st3.lastIndexOf('\n'));
                jTextArea2.setText(st4);
                String fname1="nominal.csv";
                File file1=new File(fname1);
FileOutputStreamfos=new FileOutputStream(file1);
fos.write(st4.getBytes());
fos.close();


  }
            else if (result == JFileChooser.CANCEL_OPTION)
            {
JOptionPane.showMessageDialog(this,"Open Command cancelled by user.");
            }
        }
catch(Exception e)
        {
e.printStackTrace();
        }
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEventevt) {
        try
        {
                        //Feature Selection

            String sou="nominal.csv";
            File file = new File(sou);
            int index = file.getName().lastIndexOf('.');
            String fname="";
            if (index>0&& index <= file.getName().length() - 2 )
            {
fname=file.getName().substring(0, index);
            }
System.out.println("fname is "+fname);
```

```
                //Convert this file CSV format to Arff format

            // load CSV
CSVLoader loader = new CSVLoader();
System.out.println("src1 is "+sou);
loader.setSource(new File(sou));
        Instances data1 = loader.getDataSet();

            // save ARFF
ArffSaver saver = new ArffSaver();
saver.setInstances(data1);
saver.setFile(new File(fname+".arff"));
fn=fname+".arff";
saver.writeBatch();

        String sou1=jTextField1.getText();
        File file1 = new File(sou1);
        int index1= file1.getName().lastIndexOf('.');
        String fname1="";
        if (index1>0&& index1 <= file1.getName().length() - 2 )
        {
           fname1=file1.getName().substring(0, index1);
        }
System.out.println("fname1 is "+fname1);

                //Convert this file CSV format to Arff format

            // load CSV
CSVLoader loader1 = new CSVLoader();
System.out.println("src1 is "+sou);
        loader1.setSource(new File(sou));
        Instances data = loader1.getDataSet();

            // save ARFF
ArffSaver saver1 = new ArffSaver();
        saver1.setInstances(data);
        saver1.setFile(new File(fname1+".arff"));
        fn4=fname1+".arff";
        //fn=fname+".arff";
        saver1.writeBatch();

System.out.println("\n0. Loading data");
DataSource source = new DataSource(fname+".arff");
        Instances data2 = source.getDataSet();
        if (data2.classIndex() == -1)
        data2.setClassIndex(data2.numAttributes() - 1);

            // 1. meta-classifier
useClassifier(data2);
```

```
                // 2. filter
useFilter(data2);

                // 3. low-level
useLowLevel(data2);
        }
catch(Exception e)
        {
e.printStackTrace();
        }
        fn5="t.arff";
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:
        String ig=jTextField2.getText();
        String s1[]=ig.split(",");
                                                              try
                                                              {
            try
            {
                double d1[]=new double[s1.length];
                double d2=0;
for(int i=0;i<s1.length;i++)
                {
                    d1[i]=new Integer(s1[i]).intValue();
                    d2=d2+d1[i];
                }
                double d5=0;
for(int i=0;i<d1.length;i++)
                {
                    double d3=d1[i]/d2;
                    double d4=Math.log(((d1[i]==0)?1:d1[i])/(d2));
                    d5=d5+(d3*d4);
                }
                double d6=d5*(-1);
                double iGain=d6/Math.log(2);
System.out.println(iGain);
                jTextField3.setText(""+iGain);
            }
catch(NumberFormatExceptionnfe)
            {
System.out.println("Error in input \""+nfe+"\"");
            }


        }
catch(Exception e)
                                                              {
System.out.println("Error in input \""+e+"\"");
                                                              }
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:
        String ig=jTextField2.getText();
        String s1[]=ig.split(",");
        double d1[]=new double[s1.length];
        double d2=0;
for(int i=0;i<s1.length;i++)
        {
            d1[i]=new Integer(s1[i]).intValue();
            d2=d2+d1[i];
        }
        double intrininfo=d2*((-1/d2)*Math.log(1/d2));
        jTextField4.setText(""+intrininfo);
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:
        double
gainratio=(Double.parseDouble(jTextField3.getText())/Double.parseDouble(jTextField4.getText()))
;
        jTextField5.setText(""+gainratio);
    }

    private void jButton6ActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:
        try
                                                    {
            //String fn1="iris.arff";
            //String fn2="iris.arff";
            String fn1=fn;
            String fn2=jTextField6.getText();
fn=jTextField6.getText();
            Instances train = new Instances(new BufferedReader(new FileReader(fn1)));
            Instances test = new Instances(new BufferedReader(new FileReader(fn2)));
            int cIdx=train.numAttributes()-1;
train.setClassIndex(cIdx);

            int cIdx1=test.numAttributes()-1;
test.setClassIndex(cIdx1);
            //Classifier cls = new J48();
            Classifier cls = new AODE();
cls.buildClassifier(train);
            // evaluate classifier and print some statistics
            Evaluation eval = new Evaluation(train);
eval.evaluateModel(cls, test);
System.out.println(eval.toSummaryString("\nResults\n======\n", false));
System.out.println("class "+eval.toClassDetailsString());
System.out.println("mat "+eval.toMatrixString());
```

```
          jTextArea3.setText("class
"+eval.toClassDetailsString()+"\n"+"\n"+"mat"+eval.toMatrixString());

                                                    }
                                                    catch(Exception e)
                                                    {
e.printStackTrace();
                                                    }

   }

   private void jButton7ActionPerformed(java.awt.event.ActionEventevt) {
       // TODO add your handling code here:
       String st="";
       String st1="";
       try
       {
System.out.println("fn4 is "+fn4);
System.out.println("fn5 is "+fn5);
BufferedReader datafile = readDataFile(fn5);

          Instances data = new Instances(datafile);
data.setClassIndex(data.numAttributes() - 1);

          // Do 10-split cross validation
Instances[][] split = crossValidationSplit(data, 10);

          // Separate split into training and testing arrays
Instances[] trainingSplits = split[0];
Instances[] testingSplits = split[1];
                                                    // Use a set of classifiers
Classifier[] models = {
             new J48(), // a decision tree
             new PART(),
             new DecisionTable(),//decision table majority classifier
             //new AODE(),
             new DecisionStump() //one-level decision tree
          };

          // Run for each model
          for (int j = 0; j <models.length; j++)
          {

             // Collect every group of predictions for current model in a FastVector
FastVector predictions = new FastVector();

             // For each training-testing split pair, train and test the classifier
             for (int i = 0; i<trainingSplits.length; i++)
             {
                Evaluation validation = classify(models[j], trainingSplits[i], testingSplits[i]);
predictions.appendElements(validation.predictions());

                                                  // Uncomment to see the summary for
each training-testing pair.

                                          //System.out.println(models[j].toString());
                                              }

                                                // Calculate overall accuracy of current classifier
on all splits
         double accuracy = calculateAccuracy(predictions);

                                                // Print current classifier's name and accuracy in
a complicated,
                                                // but nice-looking way.
System.out.println("Accuracy of " + models[j].getClass().getSimpleName() + ": "
                                                                +        String.format("%.2f%%",
accuracy)
                                                                +  "\n--------------------------------
");
st=st+"Accuracy of " + models[j].getClass().getSimpleName() + ": "
                                                                +        String.format("%.2f%%",
accuracy)
                                                                +  "\n--------------------------------
"+"\n";
       }
     }
catch(Exception e)
    {
e.printStackTrace();
    }
    //jTextArea4.setText(st);        /*    jTextArea4.setText("Accuracy    of    "    +
models[j].getClass().getSimpleName() + ": "
                                                                //+      String.format("%.2f%%",
accuracy)
                                                                //+ "\n-------------------------------
-");*/
     try
     {
BufferedReader datafile1 = readDataFile(fn);

      Instances data1 = new Instances(datafile1);
      data1.setClassIndex(data1.numAttributes() - 1);

      // Do 10-split cross validation
Instances[][] split1 = crossValidationSplit(data1, 10);

      // Separate split into training and testing arrays
Instances[] trainingSplits1 = split1[0];
Instances[] testingSplits1 = split1[1];
```

```
                                                   // Use a set of classifiers
            Classifier models1 =new AODE();
            // Collect every group of predictions for current model in a FastVector
FastVector predictions1 = new FastVector();

            // For each training-testing split pair, train and test the classifie
            int result=fc.showOpenDialog(this);
            if (result == JFileChooser.APPROVE_OPTION)
            {
                File file=fc.getSelectedFile();
                jTextField6.setText(file.getAbsolutePath());


            }
            else if (result == JFileChooser.CANCEL_OPTION)
            {
JOptionPane.showMessageDialog(this,"Open Command cancelled by user.");
            }
        }
catch(Exception e)
        {
e.printStackTrace();
        }
    }

    private void jTextField6ActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(AODEFrame.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (InstantiationException ex) {
```

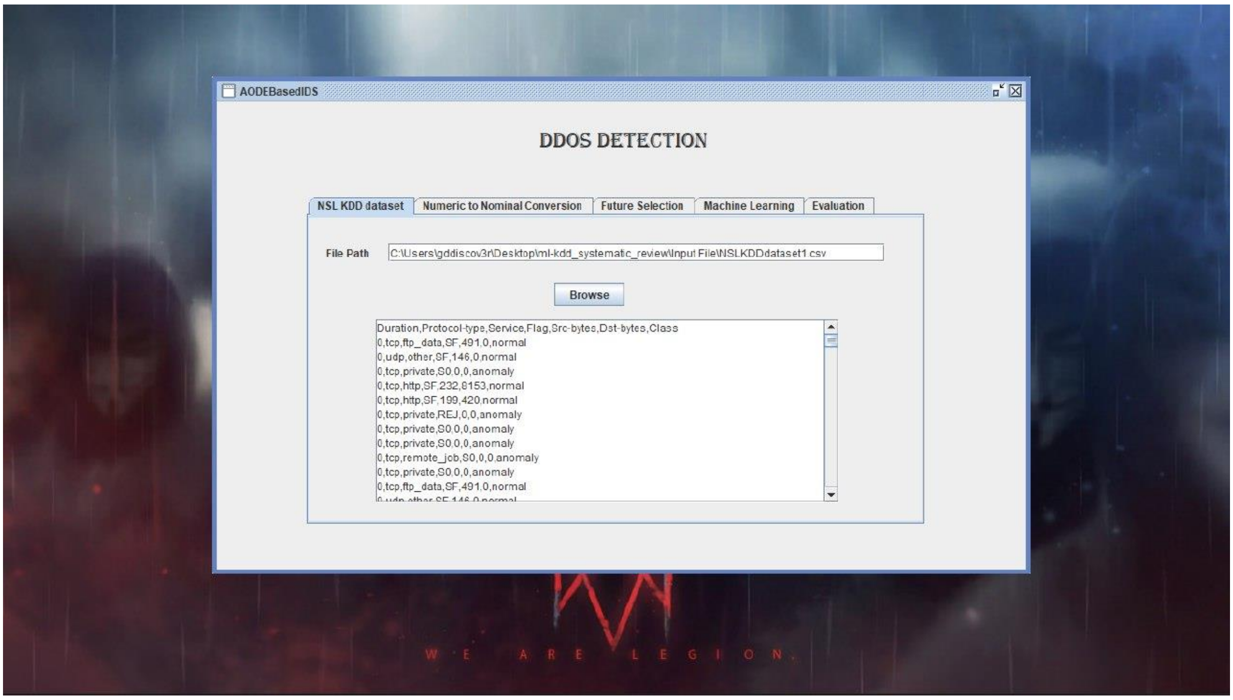
```
java.util.logging.Logger.getLogger(AODEFrame.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(AODEFrame.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(AODEFrame.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new AODEFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JButton jButton8;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JPanel jPanel6;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JScrollPane jScrollPane4;
    private javax.swing.JTabbedPane jTabbedPane1;
    private javax.swing.JTextArea jTextArea1;
    private javax.swing.JTextArea jTextArea2;
    private javax.swing.JTextArea jTextArea3;
    private javax.swing.JTextArea jTextArea4;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    // End of variables declaration

    private void useClassifier(Instances data) throws Exception {
System.out.println("\n1. Meta-classfier");
AttributeSelectedClassifier classifier = new AttributeSelectedClassifier();
CfsSubsetEval eval = new CfsSubsetEval();
GreedyStepwise search = new GreedyStepwise();
search.setSearchBackwards(true);
        J48 base = new J48();
classifier.setClassifier(base);
classifier.setEvaluator(eval);
classifier.setSearch(search);
        Evaluation evaluation = new Evaluation(data);
evaluation.crossValidateModel(classifier, data, 10, new Random(1));
System.out.println(evaluation.toSummaryString());
        //throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    private void useFilter(Instances data) throws Exception{
System.out.println("\n2. Filter");
weka.filters.supervised.attribute.AttributeSelection filter = new weka.filters.supervised.attribute.AttributeSelection();
CfsSubsetEval eval = new CfsSubsetEval();
GreedyStepwise search = new GreedyStepwise();
search.setSearchBackwards(true);
filter.setEvaluator(eval);
filter.setSearch(search);
filter.setInputFormat(data);
        Instances newData = Filter.useFilter(data, filter);
System.out.println(newData);
        //throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    private void useLowLevel(Instances data) throws Exception {
System.out.println("\n3. Low-level");
AttributeSelectionattsel = new AttributeSelection();
CfsSubsetEval eval = new CfsSubsetEval();
GreedyStepwise search = new GreedyStepwise();
search.setSearchBackwards(true);
attsel.setEvaluator(eval);
attsel.setSearch(search);
attsel.SelectAttributes(data);
int[] indices = attsel.selectedAttributes();
System.out.println("selected attribute indices (starting with 0):\n" + Utils.arrayToString(indices));
        jTextField2.setText(Utils.arrayToString(indices));
        //throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
```

```
   }

   private BufferedReaderreadDataFile(String fn) {
BufferedReaderinputReader = null;

                                                try
      {
inputReader = new BufferedReader(new FileReader(fn));
                                                }
      catch (FileNotFoundException ex)
      {
System.err.println("File not found: " + fn);
                                                }

                                                return inputReader;
      //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
   }

   private Instances[][] crossValidationSplit(Instances data, int numberOfFolds) {
Instances[][] split = new Instances[2][numberOfFolds];

                                                for (int i = 0; i<numberOfFolds; i++)
      {
         split[0][i] = data.trainCV(numberOfFolds, i);
         split[1][i] = data.testCV(numberOfFolds, i);
                                                }
                                                return split;
      //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
   }

   private Evaluation classify(Classifier model, Instances trainingSet, Instances testingSet) throws
Exception {
      Evaluation evaluation = new Evaluation(trainingSet);

                                                model.buildClassifier(trainingSet);
                                                evaluation.evaluateModel(model, testingSet);

                                                return evaluation;
      //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
   }

   private double calculateAccuracy(FastVector predictions) {
      double correct = 0;

                                                for (int i = 0; i<predictions.size(); i++)
      {
NominalPrediction np = (NominalPrediction) predictions.elementAt(i);
```

```
          if (np.predicted() == np.actual())
          {
             correct++;
          }
                                                 }
                                                 return 100 * correct / predictions.size();
       //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
   }}
```

9.2 SCREEN SHOTS

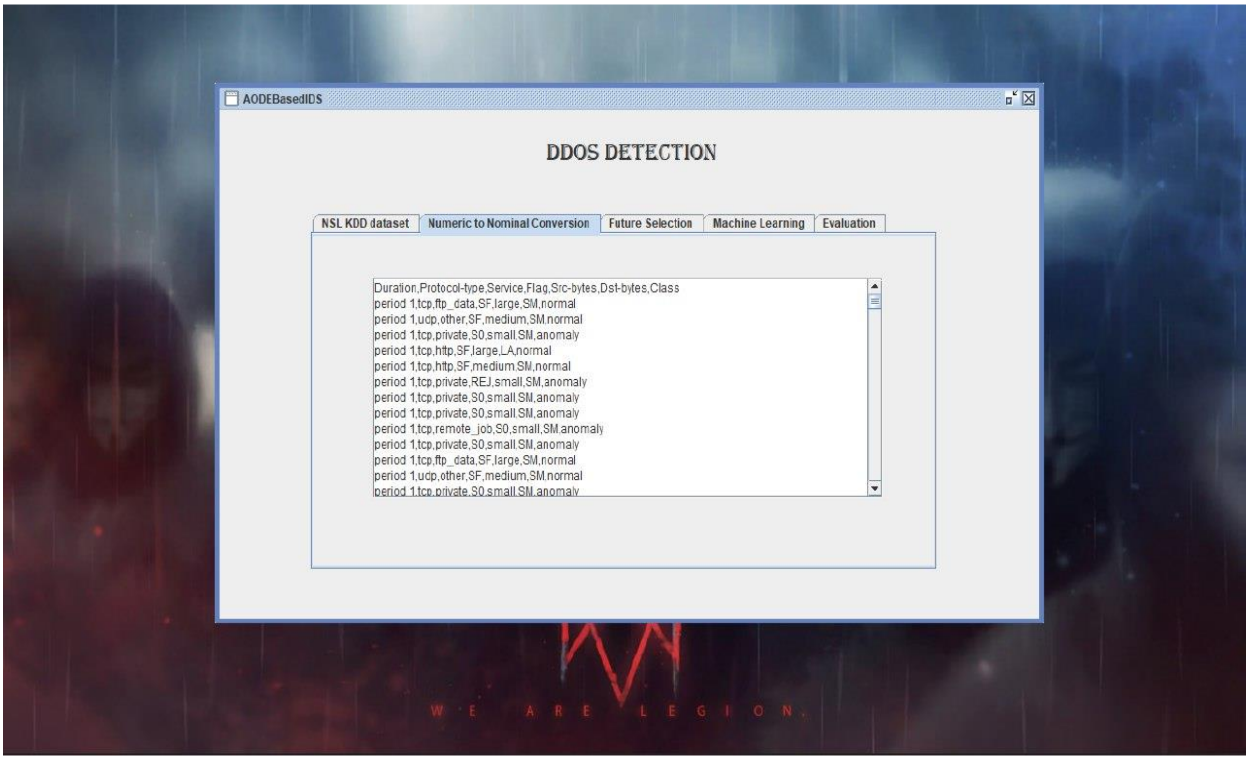
KDD DATASET



File Path: C:\Users\gddiscov3r\Desktop\ml-kdd_systematic_review\Input File\NSLKDDdataset1.csv

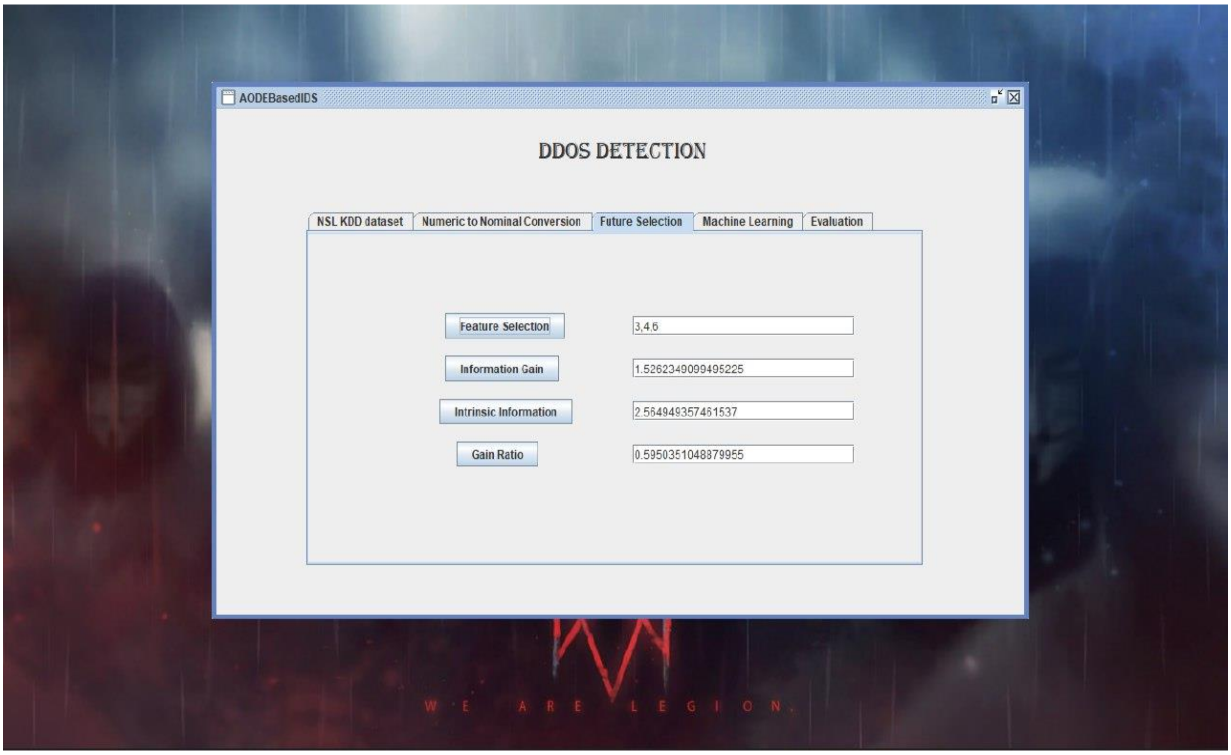
```
Duration,Protocol-type,Service,Flag,Src-bytes,Dst-bytes,Class
0,tcp,ftp_data,SF,491,0,normal
0,udp,other,SF,146,0,normal
0,tcp,private,S0,0,0,anomaly
0,tcp,http,SF,232,8153,normal
0,tcp,http,SF,199,420,normal
0,tcp,private,REJ,0,0,anomaly
0,tcp,private,S0,0,0,anomaly
0,tcp,private,S0,0,0,anomaly
0,tcp,remote_job,S0,0,0,anomaly
0,tcp,private,S0,0,0,anomaly
0,tcp,ftp_data,SF,491,0,normal
0,udp,other,SF,146,0,normal
```

NUMERIC TO NOMINAL CONVERSION



```
Duration,Protocol-type,Service,Flag,Src-bytes,Dst-bytes,Class
period 1,tcp,ftp_data,SF,large,SM,normal
period 1,udp,other,SF,medium,SM,normal
period 1,tcp,private,S0,small,SM,anomaly
period 1,tcp,http,SF,large,LA,normal
period 1,tcp,http,SF,medium,SM,normal
period 1,tcp,private,REJ,small,SM,anomaly
period 1,tcp,private,S0,small,SM,anomaly
period 1,tcp,private,S0,small,SM,anomaly
period 1,tcp,remote_job,S0,small,SM,anomaly
period 1,tcp,private,S0,small,SM,anomaly
period 1,tcp,ftp_data,SF,large,SM,normal
period 1,udp,other,SF,medium,SM,normal
period 1,tcp,private,S0,small,SM,anomaly
```

FEATURE SELECTION



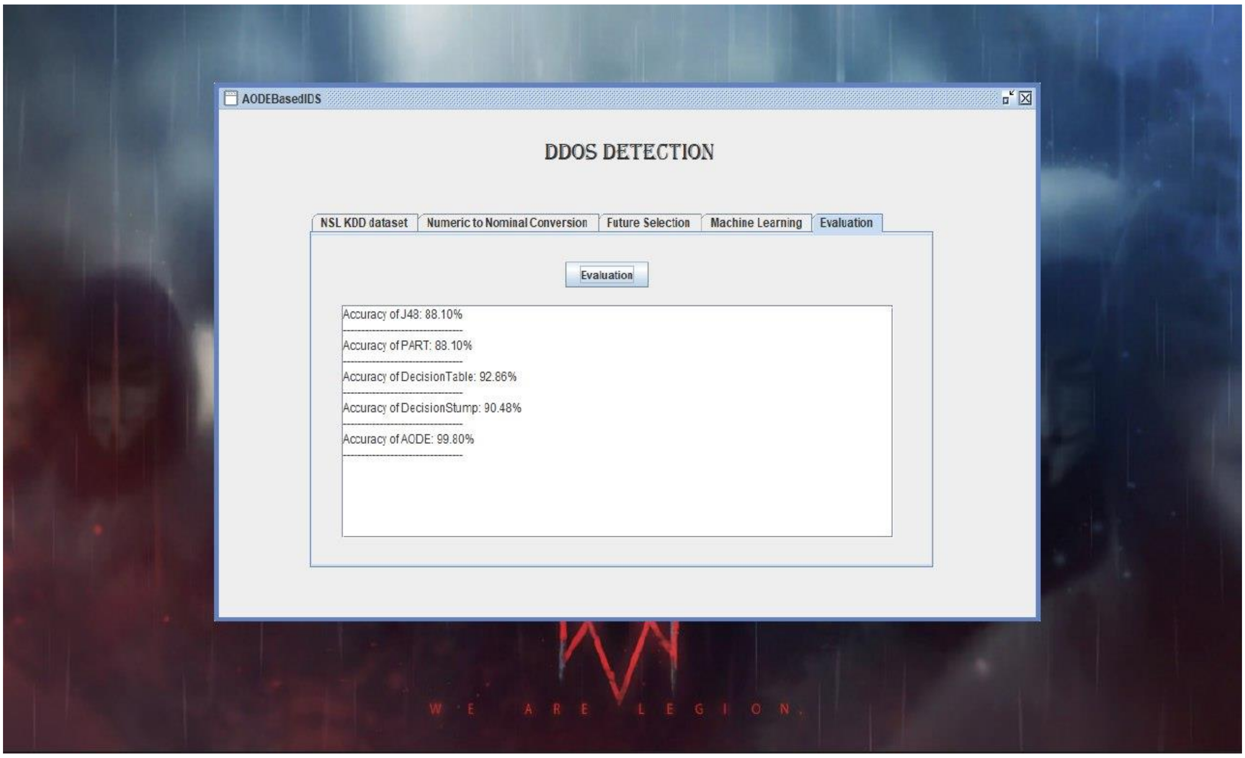
Feature Selection: 3,4,6

Information Gain: 1.5262349099495225

Intrinsic Information: 2.564949357461537

Gain Ratio: 0.5950351048879955

MACHINE LEARNING AND EVALUATION



```
Accuracy of J48: 88.10%
--------------------------------
Accuracy of PART: 88.10%
--------------------------------
Accuracy of DecisionTable: 92.86%
--------------------------------
Accuracy of DecisionStump: 90.48%
--------------------------------
Accuracy of AODE: 99.80%
--------------------------------
```

CHAPTER 10

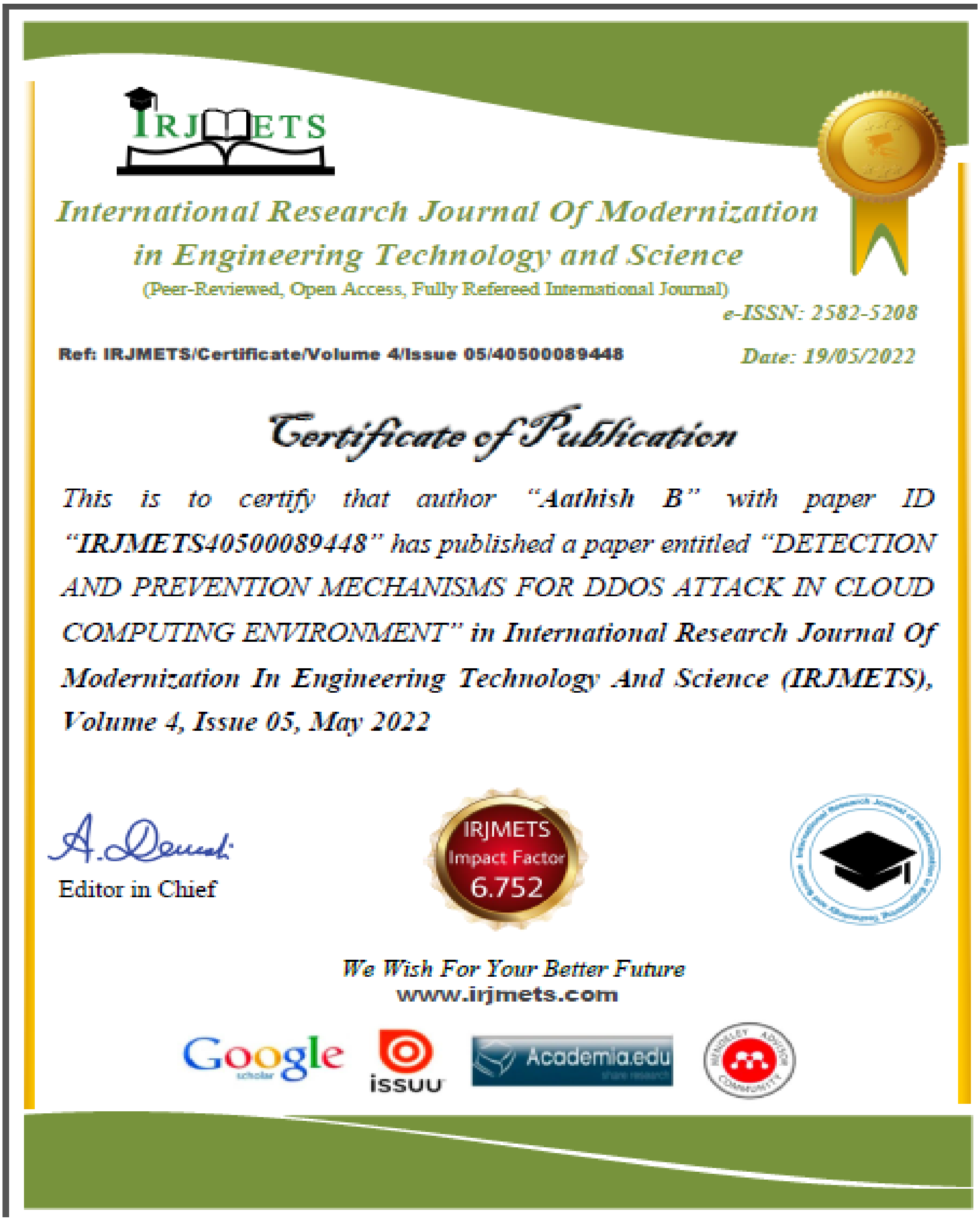
REFERENCES

1. A. Singh, U. Chandra, S. Kumar and K. Chatterjee, "A Secure Access Control Model for E-health Cloud," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 2329-2334.
2. P. J. Sun, "Privacy Protection and Data Security in Cloud Computing: A Survey, Challenges, and Solutions," in IEEE Access, vol. 7, pp. 147420-147452, 2019.
3. W. Ahmad, S. Wang, A. Ullah, Sheharyar, Z. Mahmood, "ReputationAware Trust and Privacy-Preservation for Mobile Cloud Computing," in IEEE Access, vol. 6, pp. 46363-46381, 2018.
4. W. Jiannan, W.Xiaojie, L. Nan, Y. Guomin, M. Yi, "A PrivacyPreserving Fog Computing Framework for Vehicular Crowdsensing Networks" (2018). Faculty of Engineering and Information Sciences, vol. 6, pp. 43776-43784, 2018 .
5. M. Zekri, S. E. Kafhali, N. Aboutabit and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, 2017, pp. 1-7.
6. H. Tabriz chi and M. Kuchaki Rafsanjani, ‘‘A survey on security challenges in cloud computing: Issues, threats, and solutions,’’ J. Supercomput., vol. 76, no. 12, pp. 9493–9532, Dec. 2020, doi: 10.1007/s11227-020- 03213-1.Gyusoo Kim and Seulgi Lee, “2014 Payment Research”, Bank of Korea, Vol. 2015, No. 1, Jan. 2015.
7. Abusitta, M. Bellaiche, M. Dagenais, and T. Halabi, ‘‘A deep learning approach for proactive multi-cloud cooperative intrusion detection system,’’ Future Gener. Comput. Syst., vol. 98, pp. 308–318, Sep. 2019, doi:10.1016/j.future.2019.03.043.
8. P. Achilleos, K. Kritikos, A. Rossini, G. M. Kapitsaki, J. Domaschka, M Orzechowski, D. Seybold, F. Griesinger, N.Nikolov, D. Romero, and G. A.Papadopoulos, ‘‘The cloud application modelling and execution language,’’ J. Cloud Comput., vol. 8, no. 1, p. 20, Dec. 2019, doi: 10.1186/s13677-019-0138-7.
9. L. Alhenaki, A. Alwatban, B. Alamri, and N. Alarifi, ‘‘A survey on the security of cloud computing,’’ in Proc. 2nd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS), May 2019, pp. 1–7, doi: 10.1109/CAIS.2019.8769497.

10. R. Kumar and R. Goyal, ‘‘On cloud security requirements, threats, vulnerabilities and countermeasures: A survey,’’ Comput. Sci. Rev., vol. 33, pp. 1–48, Aug. 2019, doi:10.1016/j.cosrev.2019.05.002.
11. L. B. Bhajantri and T. Mujawar, ‘‘A survey of cloud computing security challenges, issues and their countermeasures,’’ in Proc. 3rd Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC), Dec. 2019, pp. 376–380, doi: 10.1109/I-SMAC47947.2019.9032545.
12. K. Vijayakumar and C. Arun, ‘‘Continuous security assessment of cloud based applications using distributed hashing algorithm in SDLC,’’ Cluster Comput., vol. 22, no. S5, pp. 10789–10800, Sep. 2019, doi: 10.1007/s10586-017-1176-x.
13. A. Singh, U. Chandra, S. Kumar and K. Chatterjee, "A Secure Access Control Model for E-health Cloud," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 2329-2334.
14. W. Ahmad, S. Wang, A. Ullah, Sheharyar, Z. Mahmood, "ReputationAware Trust and Privacy-Preservation for Mobile Cloud Computing," in IEEE Access, vol. 6, pp. 46363-46381, 2018.
15. W. Jiannan, W.Xiaojie, L. Nan, Y. Guomin, M. Yi, "A PrivacyPreserving Fog Computing Framework for Vehicular Crowdsensing Networks" (2018). Faculty of Engineering and Information Sciences, vol. 6, pp. 43776-43784, 2018
16. M. Zekri, S. E. Kafhali, N. Aboutabit and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, 2017, pp. 1-7.

CERTIFICATES

JOURNAL CERTIFICATES :



IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 05/40500089448

Date: 19/05/2022

Certificate of Publication

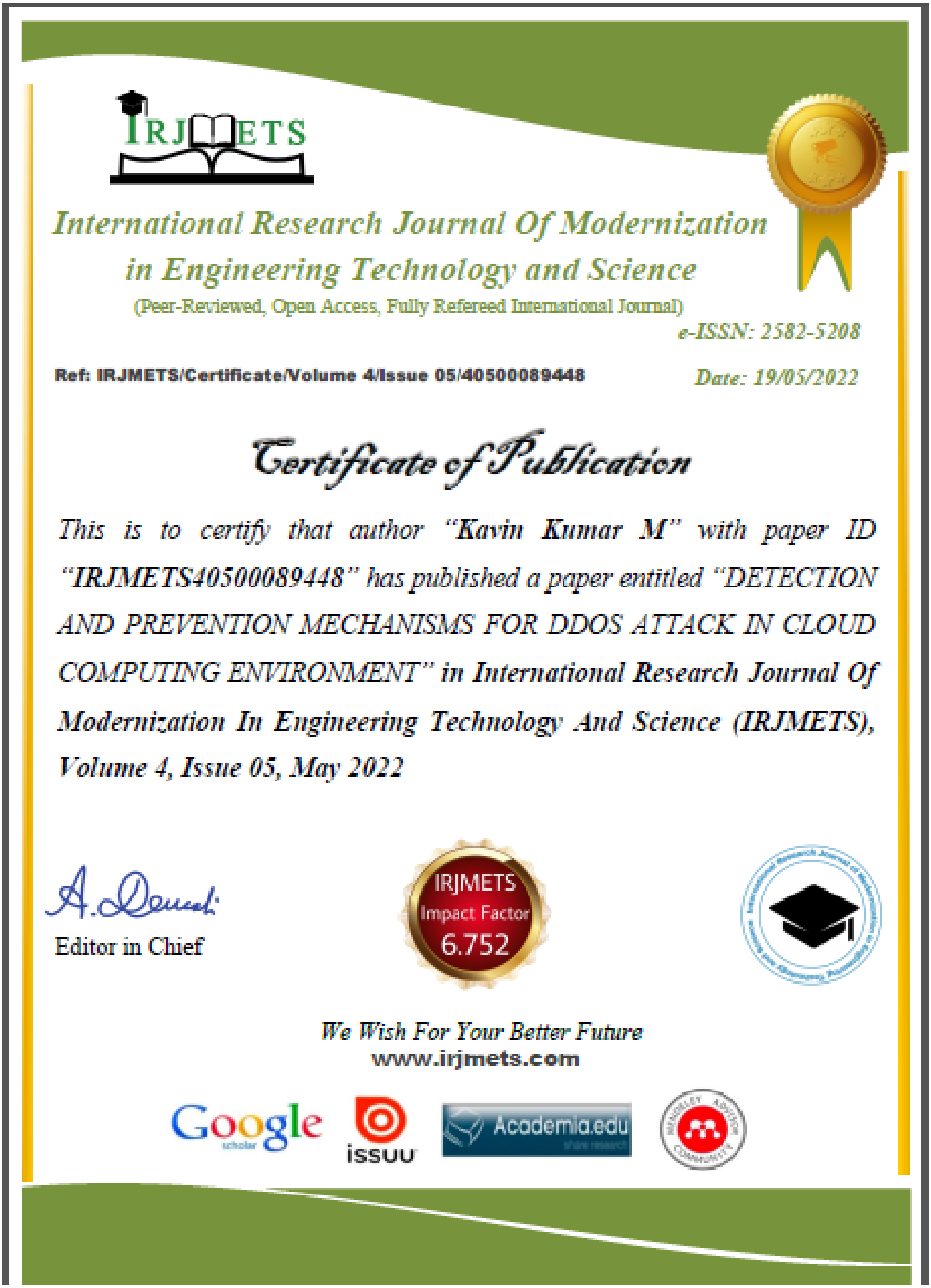
This is to certify that author "Aathish B" with paper ID "IRJMETS40500089448" has published a paper entitled "DETECTION AND PREVENTION MECHANISMS FOR DDOS ATTACK IN CLOUD COMPUTING ENVIRONMENT" in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 05, May 2022

Editor in Chief

IRJMETS Impact Factor 6.752

We Wish For Your Better Future
www.irjmets.com

Google scholar issuu Academia.edu



IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 05/40500089448 *Date: 19/05/2022*

Certificate of Publication

*This is to certify that author "**Kavin Kumar M**" with paper ID "**IRJMETS40500089448**" has published a paper entitled "DETECTION AND PREVENTION MECHANISMS FOR DDOS ATTACK IN CLOUD COMPUTING ENVIRONMENT" in **International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 05, May 2022***

A. Deual

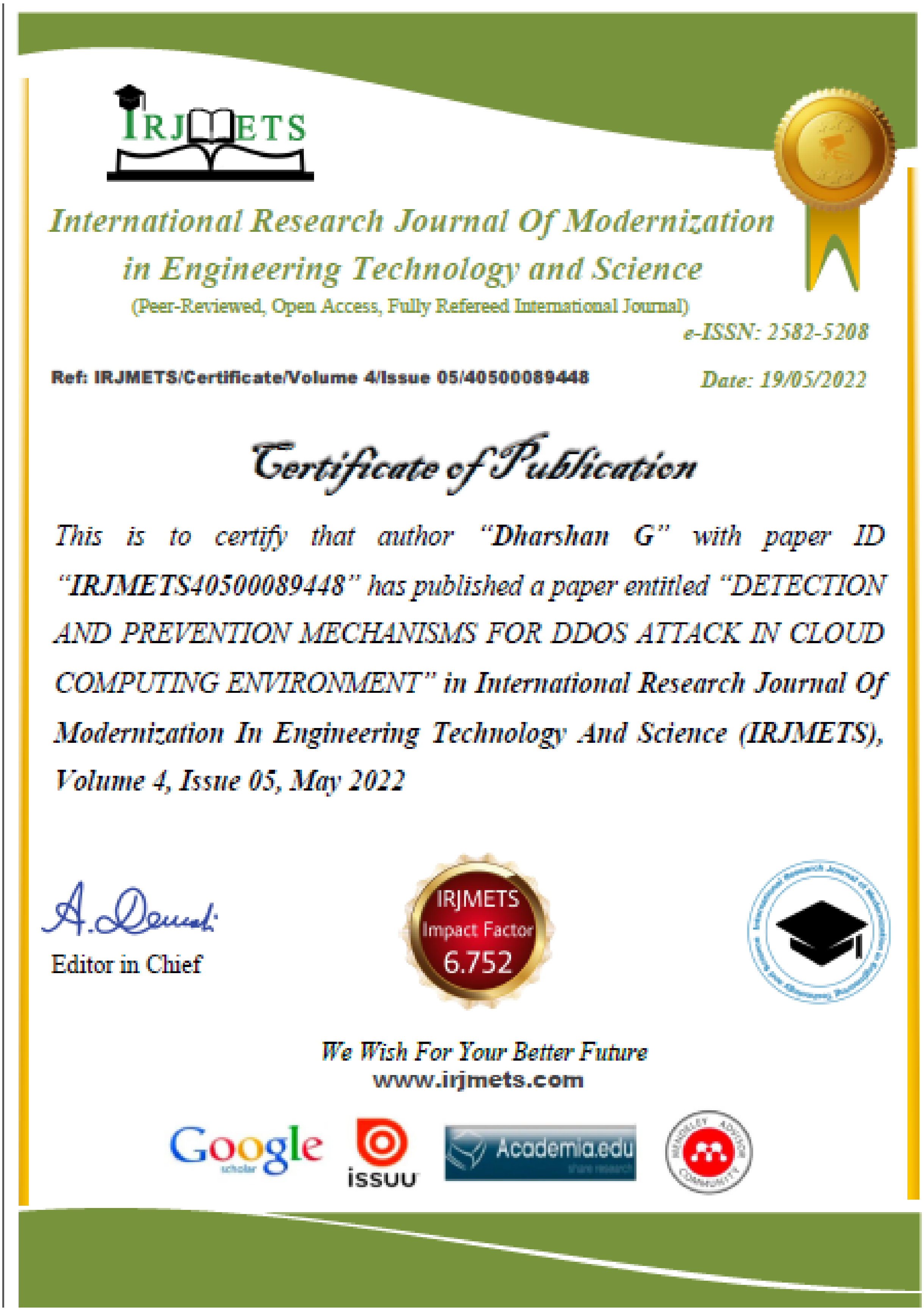
Editor in Chief

IRJMETS Impact Factor 6.752

We Wish For Your Better Future

www.irjmets.com

Google scholar issuu Academia.edu



International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 05/40500089448

Date: 19/05/2022

Certificate of Publication

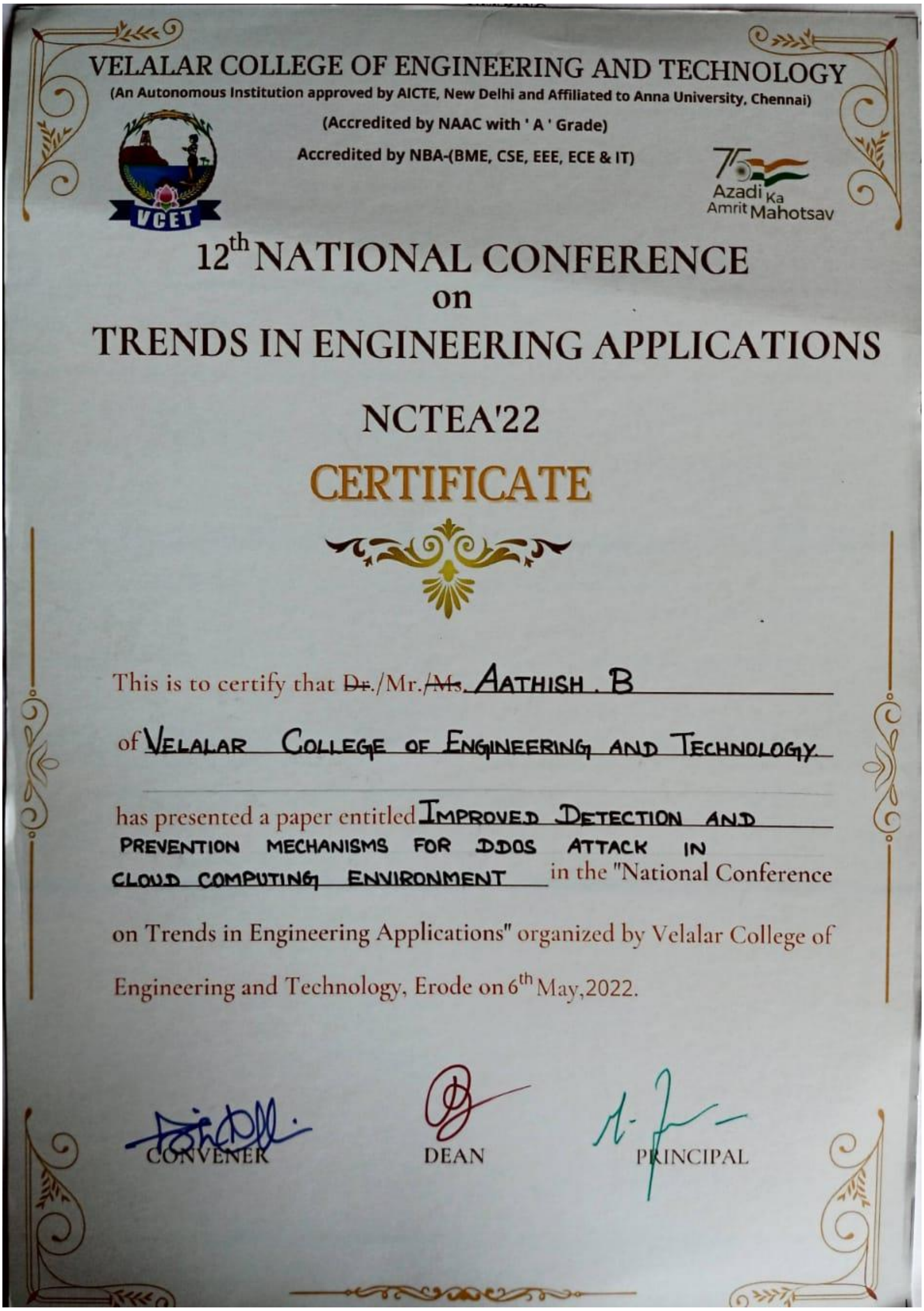
This is to certify that author "Dharshan G" with paper ID "IRJMETS40500089448" has published a paper entitled "DETECTION AND PREVENTION MECHANISMS FOR DDOS ATTACK IN CLOUD COMPUTING ENVIRONMENT" in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 05, May 2022

Editor in Chief

We Wish For Your Better Future

www.irjmets.com

CONFERENCE CERTIFICATES :



VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution approved by AICTE, New Delhi and Affiliated to Anna University, Chennai)

(Accredited by NAAC with ' A ' Grade)

Accredited by NBA-(BME, CSE, EEE, ECE & IT)

12th NATIONAL CONFERENCE

on

TRENDS IN ENGINEERING APPLICATIONS

NCTEA'22

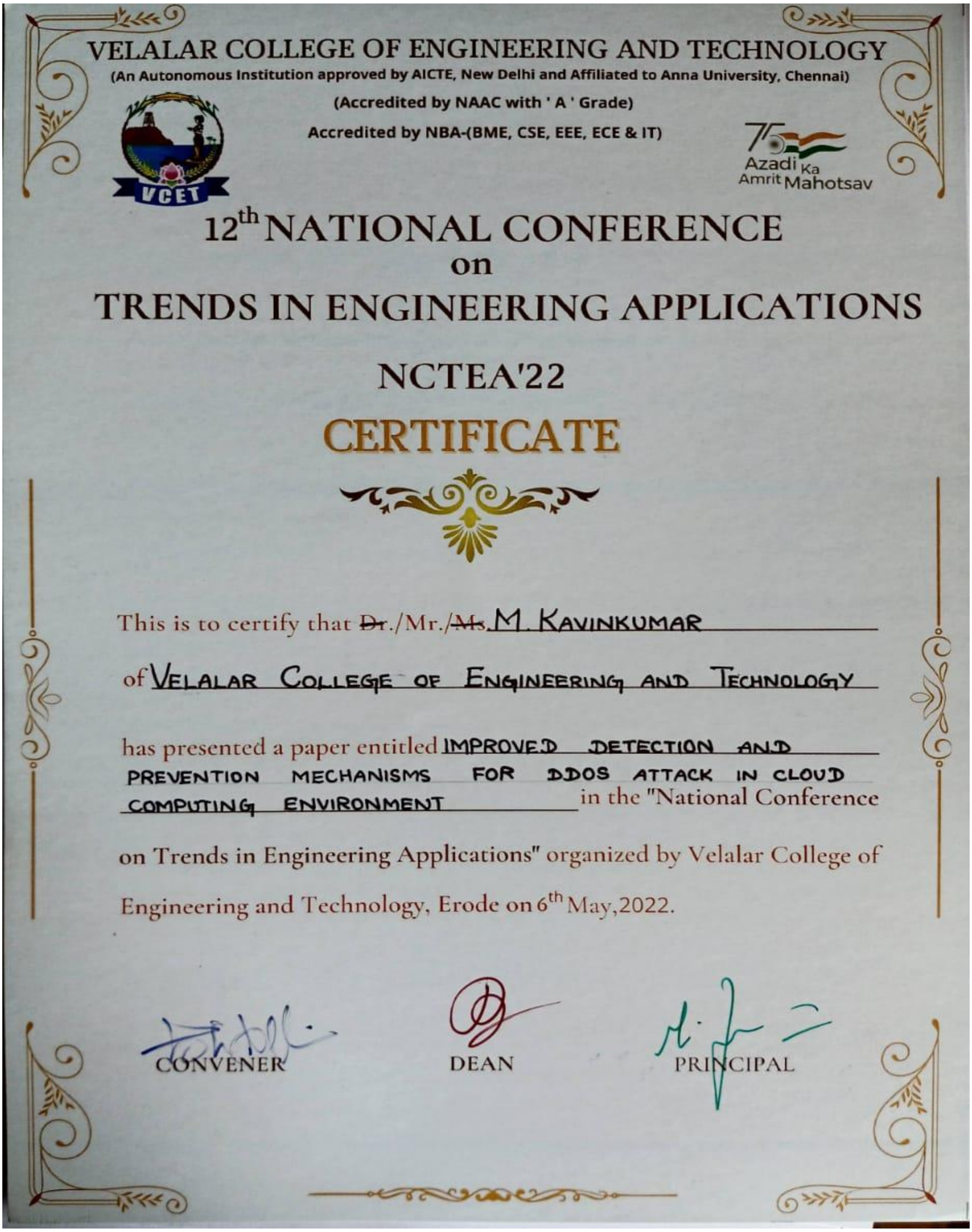
CERTIFICATE

This is to certify that ~~Dr.~~/Mr./~~Ms.~~ AATHISH . B

of VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY.

has presented a paper entitled IMPROVED DETECTION AND PREVENTION MECHANISMS FOR DDOS ATTACK IN CLOUD COMPUTING ENVIRONMENT in the "National Conference on Trends in Engineering Applications" organized by Velalar College of Engineering and Technology, Erode on 6th May,2022.

CONVENER DEAN PRINCIPAL



VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution approved by AICTE, New Delhi and Affiliated to Anna University, Chennai)

(Accredited by NAAC with ' A ' Grade)

Accredited by NBA-(BME, CSE, EEE, ECE & IT)

VCET

75 Azadi Ka Amrit Mahotsav

12th NATIONAL CONFERENCE on TRENDS IN ENGINEERING APPLICATIONS

NCTEA'22

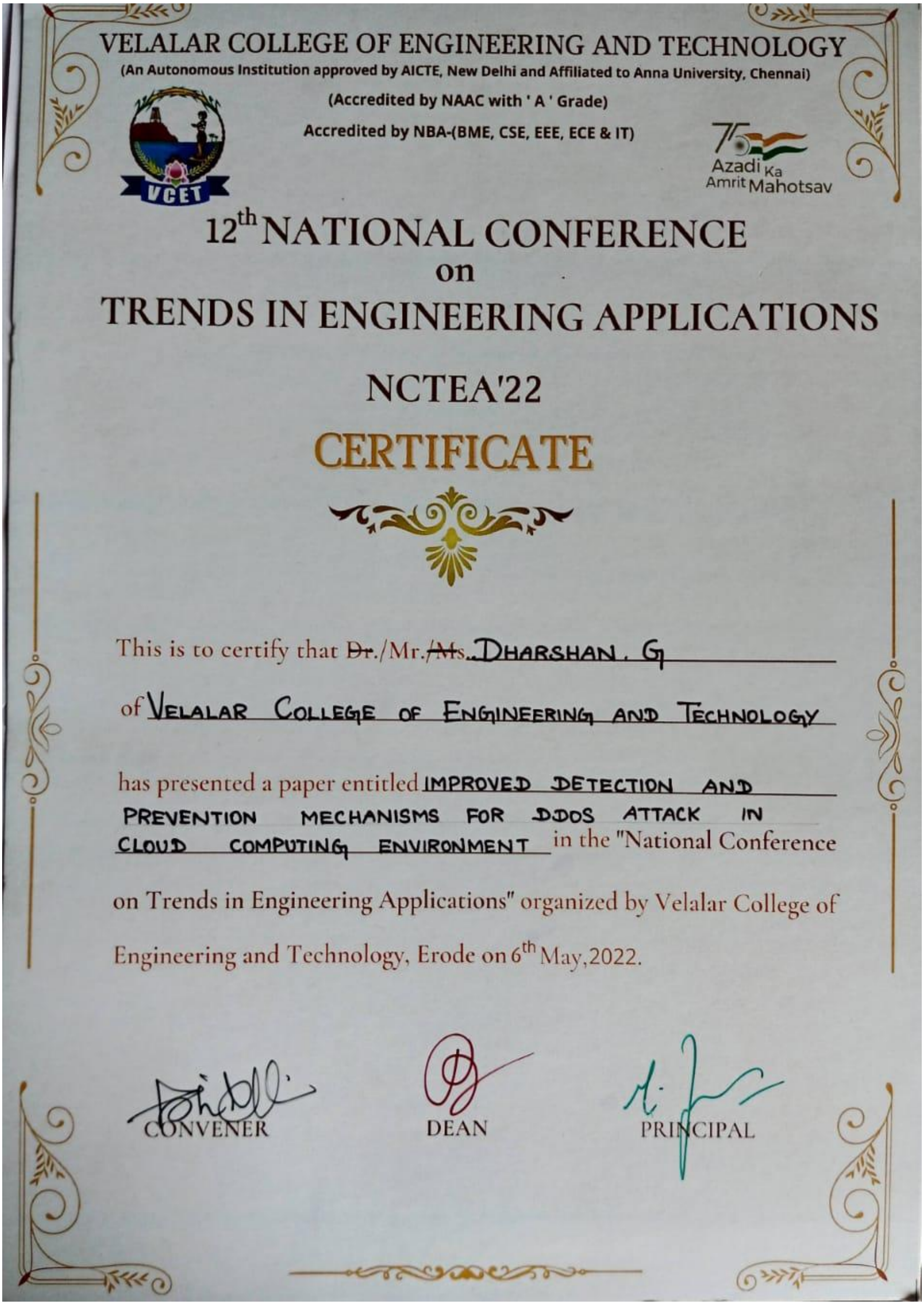
CERTIFICATE

This is to certify that ~~Dr.~~/Mr./~~Ms.~~ M. KAVINKUMAR

of VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

has presented a paper entitled IMPROVED DETECTION AND PREVENTION MECHANISMS FOR DDOS ATTACK IN CLOUD COMPUTING ENVIRONMENT in the "National Conference on Trends in Engineering Applications" organized by Velalar College of Engineering and Technology, Erode on 6th May,2022.

CONVENER DEAN PRINCIPAL



VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution approved by AICTE, New Delhi and Affiliated to Anna University, Chennai)

(Accredited by NAAC with ' A ' Grade)

Accredited by NBA-(BME, CSE, EEE, ECE & IT)

VCET

75 Azadi Ka Amrit Mahotsav

12th NATIONAL CONFERENCE on TRENDS IN ENGINEERING APPLICATIONS

NCTEA'22

CERTIFICATE

This is to certify that ~~Dr.~~/Mr./~~Ms.~~ DHARSHAN . G

of VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

has presented a paper entitled IMPROVED DETECTION AND PREVENTION MECHANISMS FOR DDOS ATTACK IN CLOUD COMPUTING ENVIRONMENT in the "National Conference on Trends in Engineering Applications" organized by Velalar College of Engineering and Technology, Erode on 6th May,2022.

CONVENER DEAN PRINCIPAL