**DDOS ATTACK DETECTION**

**Abstract**

Distributed denial of service (DDoS) attacks pose an increasing threat to businesses and government agencies. They harm internet businesses, limit access to information and services, and damage corporate brands. Attackers use application layer DDoS attacks that are not easily detectable because of impersonating authentic users. In this study, we address novel application layer DDoS attacks by analysing the characteristics of incoming packets, including the size of HTTP frame packets, the number of Internet Protocol (IP) addresses sent, constant mappings of ports, and the number of IP addresses using proxy IP. We analysed client behaviour in public attacks using standard datasets, the CTU-13 dataset, real weblogs (dataset) from our organisation, and experimentally created datasets from DDoS attack tools: Slow Lairs, Hulk, Golden Eyes, and Xerex. A multilayer perceptron (MLP), a deep learning algorithm, is used to evaluate the effectiveness of metrics-based attack detection. Simulation results show that the proposed MLP classification algorithm has an efficiency of 98.99% in detecting DDoS attacks. The performance of our proposed technique provided the lowest value of false positives of 2.11% compared to conventional classifiers, i.e., Naïve Bayes, Decision Stump, Logistic Model Tree, Naïve Bayes Updateable, Naïve Bayes Multinomial Text, AdaBoostM1, Attribute Selected Classifier, Iterative Classifier, and OneR.

Keywords : ML , Model , Script, Ddos, Traffic snippets.

1. **INTRODUCTION**

In today’s fast-paced world, where the number of internet-connected devices is increasing and online applications are growing at a rapid pace, information security is becoming an absolute necessity. Since the beginning of the World Wide Web, 1.2 billion websites have been developed, and a huge number and variety of online applications are integrated with various web services, such as e-commerce, online banking, online shopping, online education, e-healthcare, and industrial control systems (ICS) for critical infrastructure, etc. Nowadays, cyber attackers are highly skilled and well-equipped to carry out successful attacks on businesses and governments. Cybercrime is big business today, and the volume of stolen information is enormous. There are many different categories of malware. This poses a huge risk to governments, businesses, and consumers around the world. We do not have to go far back in time to remember the massive attack on a bank in Bangladesh, where USD 81 million was reportedly stolen. This is a constant reminder of how effective these attacks can be; the bank’s own computers were used to transfer large sums of money. No business is safe, no matter how large. Statistics show that 20% of affected businesses fall into the small business category, 33% into the SME category, and 41% into the large business category. The more widespread the threat, the more important it becomes to be aware of the issues and protect the important information. Eighty-two percent of organisations have been exposed to at least one or more attacks in which data are stolen and used to cripple the victim’s services. The organisations that were affected by DDoS attacks reported a 26% drop in performance of their services and 41% reported an outage of the affected services . Figure 1 shows an environment of DDoS attacks.

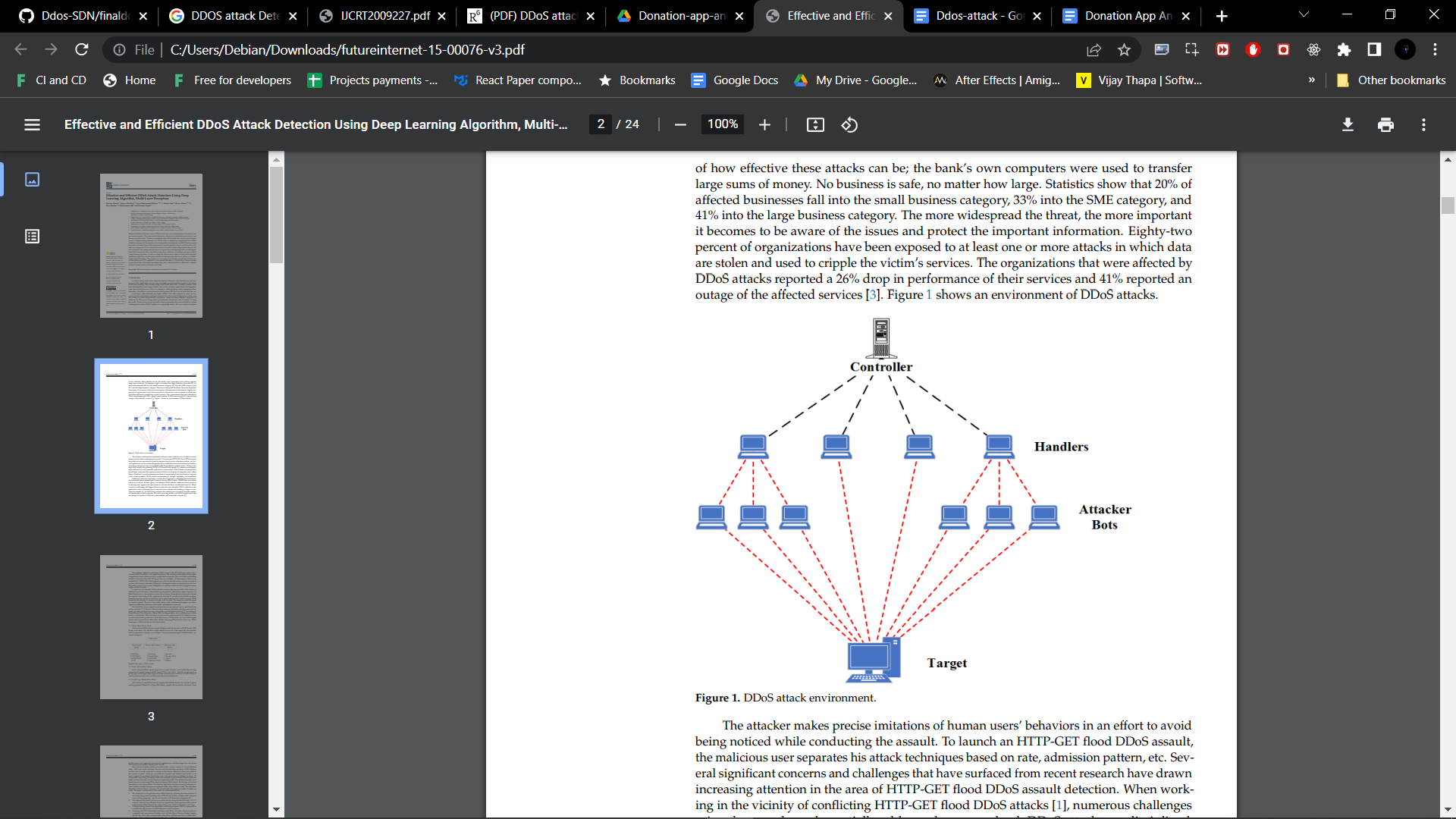


Fig. 1 Ddos Attack Environment.

The attacker makes precise imitations of human users’ behaviours in an effort to avoid being noticed while conducting the assault. To launch an HTTP-GET flood DDoS assault, the malicious user separates his attack techniques based on rate, admission pattern, etc. Several significant concerns and challenges that have surfaced from recent research have drawn increasing attention in the area of HTTP-GET flood DDoS assault detection. When working in the vicinity of conflicting HTTP-GET flood DDoS attacks, numerous challenges arise, that are also only partially addressed or unresolved. DDoS attacks are disciplined, distributed, and remotely organised networks that use deployed computers (also called Bots or Zombies) to send an immense number of uninterrupted and synchronous requests to the victim system(s). DDoS attacks are increasing in strength, regularity, and complexity. Malicious users are constantly evolving their experience, adapting their techniques, and using advanced technologies to launch various DDoS attacks. While there are various solutions to detect, defend against, or mitigate DDoS attacks, malicious users continue to develop new approaches and means to circumvent these countermeasures . DDoS events are still among the biggest threats to the network. Recently, DDoS outbreaks at the application layer of internet servers have become widespread, resulting in huge revenue losses for targets. In TCP/IP layer attacks, the online server is crushed and the number of requests per second is limited. Slowloris, zero-day attacks, and DDoS assaults that take advantage of Apache or Windows vulnerabilities fall under this category.

The solutions offered to understand DDoS attacks at the TCP/IP layer capture only a subset of DDoS incidents at the application layer. The resolutions that detect entire types of application-layer attacks are very complicated in formula. One set of tasks in detecting a DDoS outbreak at the TCP/IP layer is the unavailability of landscapes to detect such incidents. HTTP-GET DDoS attacks are a risk for all web servers, as bots are able to impersonate humans and make it difficult to distinguish malicious requests from real ones. Regardless of industry or scale, enterprises around the world are increasingly becoming targets of DDoS attacks. Complexity and strength of these attacks are increasing exponentially as the number of admitted systems increase, vulnerabilities go un-patched, and business impact increases. DDoS attacks have a strong impact on the cyber domain. Cyber attacks are feared to disrupt the regular functioning of the organization through IP overflow, bandwidth spoofing, intensive memory resources, and root sane or mouse damage. A slow-moving DDoS attack has the capacity to mimic real traffic with its traffic. It is simple to avoid detection by current systems. Based on their rank values, rank correlation techniques can detect significant differences between attack traffic and legitimate traffic. DoS attacks has serious impacts on information servers, internet servers, and cloud computing servers. Botnets, DDoS, hacking, malware, pharming, phishing, ransomware, spam, spoofing, and spyware are some of the most frequent hazards. According to Ginni Rometty, Chief Executive Officer IBM, the biggest risk to any or all businesses worldwide is a cyberattack. With that, there is an increase in cybercriminals. Malicious users use numerous hacking methods to hack client servers. DDoS attacks are very wide-ranging attacks and occur between other cyber attacks; detecting DDoS attacks is not easy. Three basic types of DDoS attacks are described below.

**1.1 Volume Based Ddos Attack**

Volume based DDoS attacks consist of faked packet floods such as ICMP floods, UDP floods, and others. The objective of this attack is to use all of the target site’s bandwidth, and it is measured in bits per second (bps). Various prominent types of DDoS attacks are shown in Figure 2.

**1.2 Protocol Based DDoS Attack**

Protocol based DDoS attacks appear in a variety of forms, such as SYN floods, fragmented pack attacks, ping of death, smurf DDoS, and others. Attacks are measured in packets per second (pps). Such types of attacks use real server resources, as well as those of central communications devices like firewalls and load balancers.

**1.3. TCP/IP Layer Based DDoS Attack**

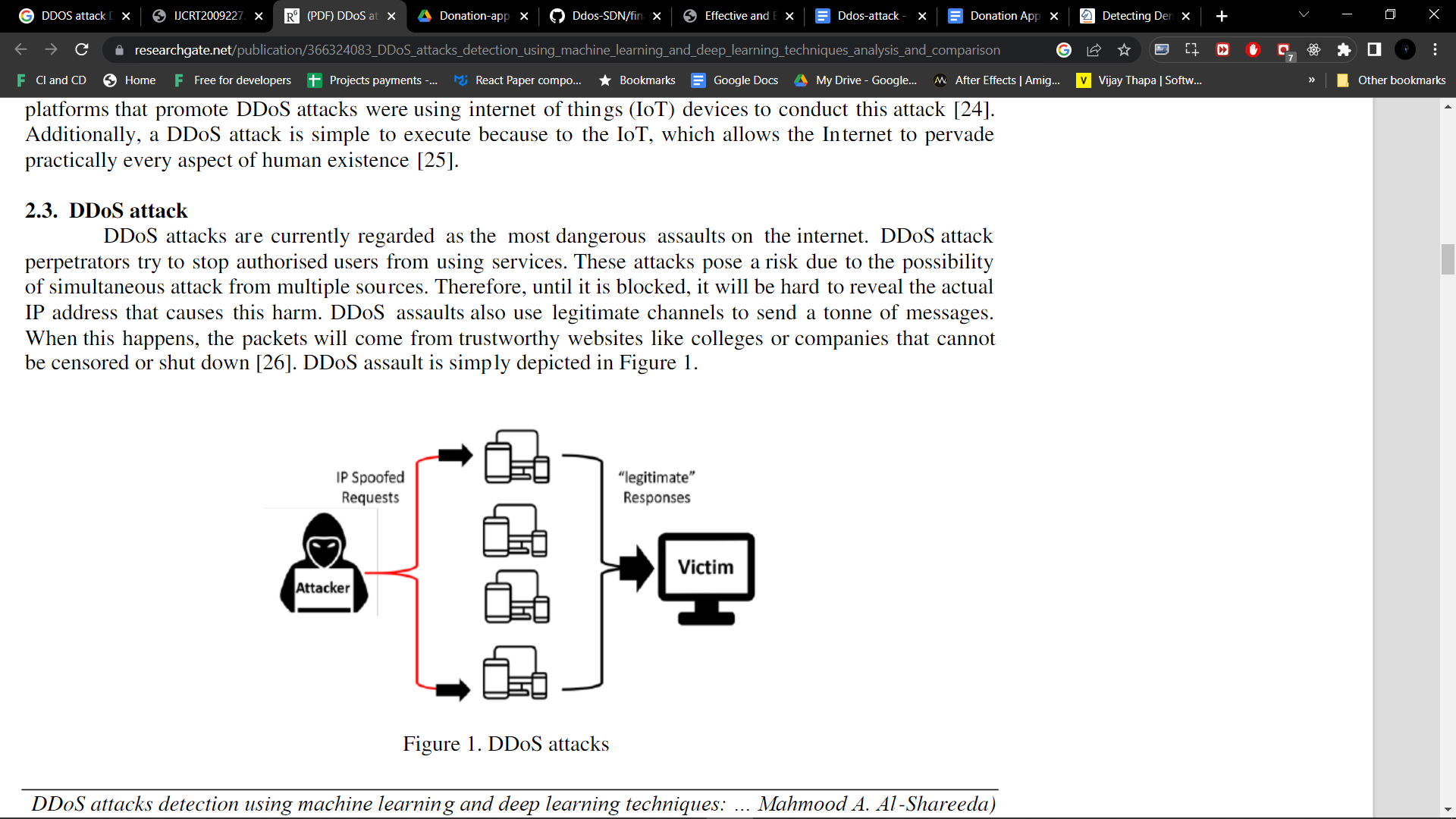
TCP/IP layer based DDoS attack comprise GET/POST floods, low and slow-speed attacks, potential Windows or OpenBSD attacks, Apache-driven attacks, and more. Such attacks seem to be legitimate and innocent applications, and they target the web server. The extent is measured by requests per second. The number of attacks and the associated traffic volume continue to increase dramatically. With such traffic intensity, the network infrastructure upstream of the intended victim is also severely impacted, so attack traffic must be filtered as close as possible to the sources of attack. However, it is difficult to predict and identify such nodes, as attacks originate from widely distributed nodes and spread across multiple locations. To successfully respond by disrupting traffic, the mitigation approach must detect malicious traffic and respond with minimal impact on legitimate traffic. The attacker launches a new attack, known as increasing DDoS attack and proxy DDoS attack. We develop a detection algorithm to solve this problem. The detection algorithm uses deep-learning techniques to detect malicious traffic and separate legitimate traffic from malicious traffic. The algorithm classifies traffic into three categories: (1) normal traffic (2) suspicious traffic malicious traffic. The main contributions of this study are summarised below.

1. We addressed novel application layer DDoS attacks by analysing the characteristics of incoming data packets including size of HTTP frame packets, number of IP addresses sent, constant mappings of ports, and number of IP addresses using proxy IP.
2. We analysed the client’s behaviour in public attacks using standard datasets, CTU-13 dataset, real web logs (dataset) from our organisation, and experimentally created datasets from DDoS attack tools such as Slow Lairs, Hulk, Golden Eyes, and Xerex.
3. A deep learning classification algorithm, multilayer perceptron (MLP), is proposed to evaluate the effectiveness of attack detection based on metrics. 4. Our proposed MLP classification model provided the lowest value of false positives as compared with conventional classifiers such as Linear Preceptron,

Multilayer perceptron, KNN, SVM, Random Forest,

The rest of the article is organised as follows. Section 2 briefly describes the literature review; the problem of motivation is discussed in Section 3. Chart flow and research methodology are presented in Section 4. The proposed attack classification model is briefly described in Section 5 and simulation results are elaborated upon in Section 6. Finally, Section 7 concludes this study along with future work.

**1.4 Objective of the Project**

****

**Fig.3 System Architecture**

**2. REQUIREMENT ANALYSIS**

**2.1 Introduction of Requirement Analysis**

This project has basic requirements to build and enhance the Jupyter ID E App complete development which takes place on Jupyter Notebook (Expected - Python Programming Language) built with neat and enhanced design based user experience.

**2.2 Hardware & Software Requirements**

| **Software Requirements** | **Hardware Requirements** |
| --- | --- |
| | Framework | Jupyter Notebook | | --- | --- | | Programming Language | Python | | Editor | VS Code | | | Processor | I5 else Ryzen 5000 series or Above | | --- | --- | | RAM | 8Gb or Above | | SSD | 128Gb | | GPU | (Optional) | |

**2.3 Software Requirement Specifications**

SRS is a document created by system analysts after the requirements are collected from various stakeholders. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, and response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, security, quality, limitations etc. The requirement received from the client is written in natural language.

It is the responsibility of system analysts to document the requirements in technical language so that they can be comprehended and useful by the software development team.

**SRS should come up with following features:**

User requirement are expressed in natural language.

Technical requirement are expressed in structured language, which is used inside the organisation.

**System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

**Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as unit testing.

**Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaw or errors. Testing is done so that the client doesn‟t face any problem during the installation of the software.

**Deployment of system:** Once the functional and non- functional testing is done, the product is deployed in the customer environment or released into the market.

**Maintenance:** This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. This modification arises either due to change requests initiated by the customer, or defects uncovered during live use of the system. Clients are provided with regular maintenance and support for the developed software.

**2.4 Review Software Analysis Literature**

1. **Jupyter Notebook**

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet.

1. **Python**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991. Designed to be easy as well as fun, the name "Python" is a nod to the British comedy group Monty Python.

**3. LITERATURE SURVEY**

**3.1 REVIEW OF LITERATURE**

Machine learning algorithms are being widely used by research community in every field of life. Prominent application areas of machine learning algorithms include image processing, forecasting, recommendation systems, healthcare, banking system, defence, education, robotics, etc.

Deep learning is a subset of machine learning. In this study, we have used a deep learning algorithm, namely, multilayer perceptron (MLP), for effective and efficient detection of DDoS attacks. State-of-art literature on DDoS attack detection is summarised in the following. Authors of have focused on mitigating multi-page HTTP DDoS attacks with slow-moving targets that target public servers. The conceptual proof model was used in a simple and validated argument.

In, the authors compared the probability similarity between cyber attack, DDoS, and mathematical prototypical probability, Levy Walks. This variation aimed to determine the suitability of Levy walk as prototypical similarity with DDoS potential features. In, the authors experimented with the clever subject of comedy measurement that utilises a conference seeking philosophy and a brilliant channel that sets shares in the traditional way.

Multilayer perceptron with a genetic algorithm (MLPGA) is proposed in to detect DDoS attacks. The authors examined the areas of incoming pockets as well. It is assumed that the non-receiver of an unusual collection returns once at the time of publication. However, the authors of provided a sequence of events for experimental distribution to test the capabilities. The authors did not show a positive impact on stock recovery, but in cases where DDoS attacks cause disruptions within the services sent by the client, the study experienced a strong negative impact. T

he current unit of current methods was created due to the actual malfunction of DDoS attack detection in the application layer. The authors developed a phase-based system with downloading local packets, fine-field extraction of these local units needed for detection, and the use of a separator for attack detection. The study at examined the impact of a DDoS attack on a state-of-the art gift network and evaluated network security mechanisms such as a router protection system and network servers. In, the authors presented a solution for such a type of DDoS attack.

When the server exceeded the limit of its application, the author then proposed a solution and sent a random number, which can be selected at an unconsidered time value, to the requesting client. Research provided a design that increases resilience to DDoS attacks by upgrading the roles of a virtual network and the software that defines a network. In the first phase, the proposed design defines the roles of the virtual network by solving the linear system. In phase two, to increase the previous protection against DDoS attacks, special VNF filters and a second path through these VNF filters were established by solving another linear system.

SDN controller switches routes with a second attack to DDoS traffic filtering methods to prevent congestion under DDoS attacks. In, the authors provided a flexible identifier that is set periodically in the background and can make additional data selections. The authors provided applications related to the occurrence of a DDoS-based attack group and a metal folding model that combines two orthogonal oddity-based attack modes. In, the authors provided a DDoS detection combining a fully based standard and an exceptionally dependent method in which three types of machine applications are found. The author first studied the performance of the proposed system under conditions enforced by normal saturation and TFN2K attacks. Then, the authors apply small costs, such as a saturation period with key traffic attack points, to soak the victim. The authors of investigated our hypotheses about the problem in the existing diagnosis method of the attack on the DoS application base with a strong attack on the algorithm of the CUSUM system.

In, the researchers developed bio-roused conflicts, based largely on the DDoS Assault framework, with the goal of achieving a faster space. The given prototype can be a bio-roused bat algorithm system, which usually handles the fast and timely location of a DDoS application over HTTP floods. The authors of proposed a cloud-based firewall to reduce DDoS attacks on the smartest grid network AMI. The Promoted Firewall is not only able to reduce the impact of DDoS attacks, but can also prevent attacks before they start.

In, the authors demonstrated another planning phase to detect and prevent multiple DDoS TCP (CS\_DDoS) attacks during the day. The proposed CS\_DDoS framework provides response protection for deleted records. In [29], the authors provided an event detection module to limit the proliferation of internet of things (IoT) services. It was modified from the current monitoring modules with information-based filters. The proposed module focuses on system behaviour during DDoS attacks and detects them using NTP-collected information used in the synchronisation service.

The author performed a demo test with an advanced module that generates a fake DDoS attack. The result showed that the deployed modules obtain high memory and accurate values, which show their effectiveness in capturing real-time events in IoT. A study done by presented exponentially weighted moving normal (EWMA) search for amazing mine learning and DDoS base discovery attacking internet of things (IoT). The authors investigated the tradeoff between statistical detection rate, warning, and localization delay. In, the authors narrow down the classification of DDoS threats that support unusual behaviour in the application layer and provide elliptical data on various DDoS tools. In addition, the author distinguishes methods of DDoS detective work based on viewing, blocking, detecting, and minimising comments. In, a step-by-step approach to DDoS attack mitigation was presented, where the entire process of mitigating DDoS attacks was forced to a single layer or multiple layers. To increase the security of DDoS attacks, the go-layer process has become a useful solution. The authors of presented a new plastic strategy for detecting Al-DDoS attacks.

Their aforementioned work differs from the previous method by considering the detection of Al-DDoS attacks in critical spine motions. A distributed, useable, automated, and interactive ISP standard was presented by the study’s authors in. It not only distributes computing complexity and storage to adjacent places, but also facilitates the early identification of DDoS attacks and flash occurrences. Using an independent multi-agent system and agents that depend on particle evolution to facilitate effective communication and precise decision-making, the authors of [35] present a unique DDoS attack detection and prevention technique. Multiple intermediate agents are used to detect DDoS attacks, and the coordinating agent is updated.

A secure root system and an access system that can identify nearby attacks on the RPL protocol have been suggested by experts of in order to mitigate the effects of such attacks. To find the malicious node, the IDS is developed, taking into account the location data and the received signal strength. Researchers discovered perplexing real-time blocking DDoS application layer assaults on the web inthat seek to be discovered quickly and quickly. ARTP is a machine learning technique for quick and accurate app DDoS detection using multiple flood requests. The work’s goal was achieved by measuring LLDoS databases through tests, and the findings showed how valuable the proposed model is.

A hybrid protocol proposed by the authors of is the best suited protocol for cloud computing to detect DDoS attacks. The authors of provide a new approach presented in this study. With the presence of these types of malicious nodes, attacks can be classified as active and inactive. In, the authors propose to identify the DDoS attack and mitigation model using the feature selection method. In the presented study, the network traffic is primarily analyzed according to the Hellinger degree. When a certain distance is detected, all data packets are analyzed and classified into two categories based on the selected segmentation factor, such as DDoS and official application groups.

The authors have addressed the problem and developed a secure system for these programs. The experts of proposed a botnet detection method that can manage multiple datasets and also detect botnets in the network. In, researchers addressed the need to prevent DDoS attacks by defining and demonstrating a mixed identification model by introducing an advanced and effective method to identify and effectively distinguish flooding in a hot crowd. In addition to introducing a multi-level classification method based on the presented set of entropy-based features with machine learning divisors to improve the low visibility and accuracy, the authors of also introduced a set of novel entropy-based symbols to help reliably detect DDoS attacks. In, researchers discussed four important network protection schemes against end-to-end network attacks, end-to-end, victim, and distributed schemes with a focus on two innovative models, Gossip and D- WARD.

In, the authors introduced a reduction method based on the fuzzy control system. It looks like inserting two new matrices. In, the researchers presented a novel selection algorithm, Dynamic Ant Colony System, with a choice of three levels of renewal function. The presented method uses different levels of pheromones to make the ants stronger. The proposed method by the authors of is contrasted with a different hybrid algorithm that is provided with 10- fold cross-validation.

The proposed method outperforms existing methods in terms of accuracy, detection rate, and false alarm rate, according to the database-based test results of KDD CUP 99. In, the experts proposed a new method to mark a packet that could be forwarded from the attacker’s side to the victim’s side. It allows the victim to ensure the necessary protection for internet service providers (ISPs). In the manuscript, the authors propose a defence system called SkyShield. This scheme uses a graphical data framework to identify and mitigate DDoS attacks at the application level. First, they proposed new split calculations in two graphs that improve the effect of network dynamics and increase the accuracy of detection. Second, they used an atypical graph to help identify the malicious predators of a persistent attack. In, the researchers proposed the concept of a system of experts. This program automatically resets security apps about incoming traffic. To achieve this, it is proposed to use a model, reasoning, and performance-based loop (LRA-M). In this case, it describes the structures of the corresponding system and defines its building blocks. In, the authors used a state-of-the-art SDN model, employing a new method for DDoS detection and mitigation known as State Sec. They demonstrated the benefits of this type of method, as shown in Figure 4

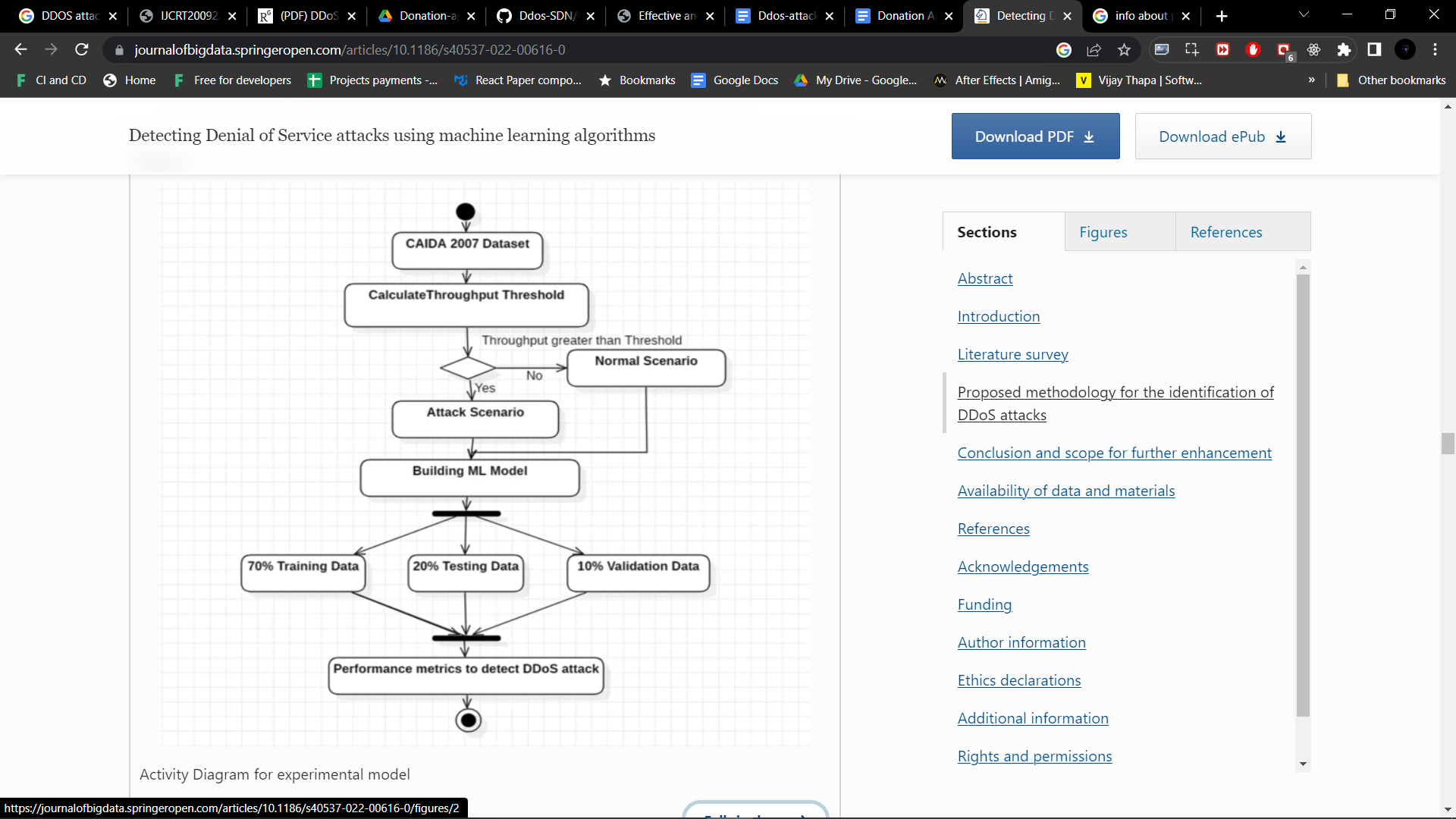
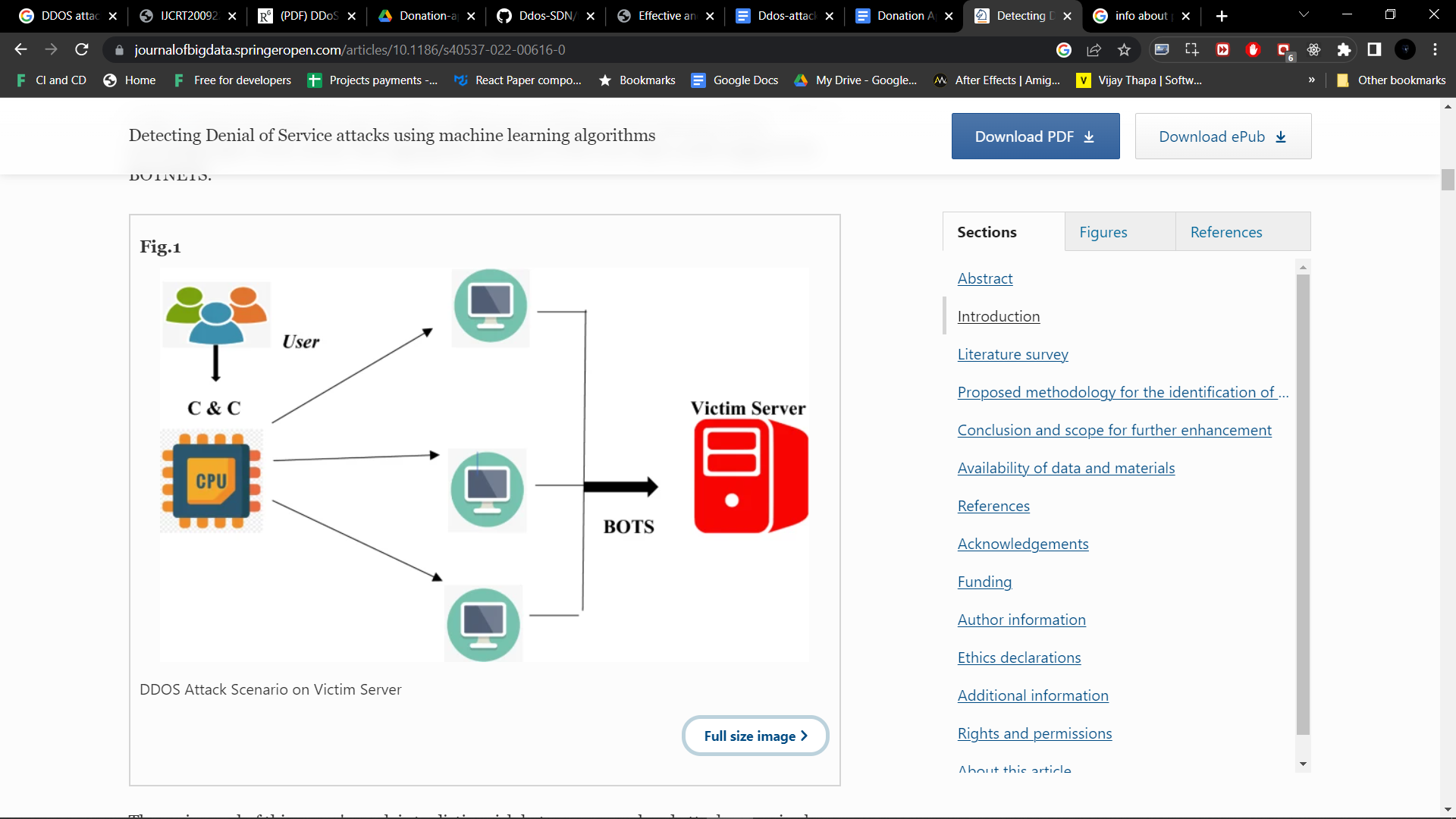


Fig.4 Experimental Model

**3.2 System Overview**

****

**Fig.5 Ddos Attack Scenario on Victim Server**

**3.3 Algorithms**

1. **SVM**
2. **KNN**
3. **Random Forest**
4. SVM

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

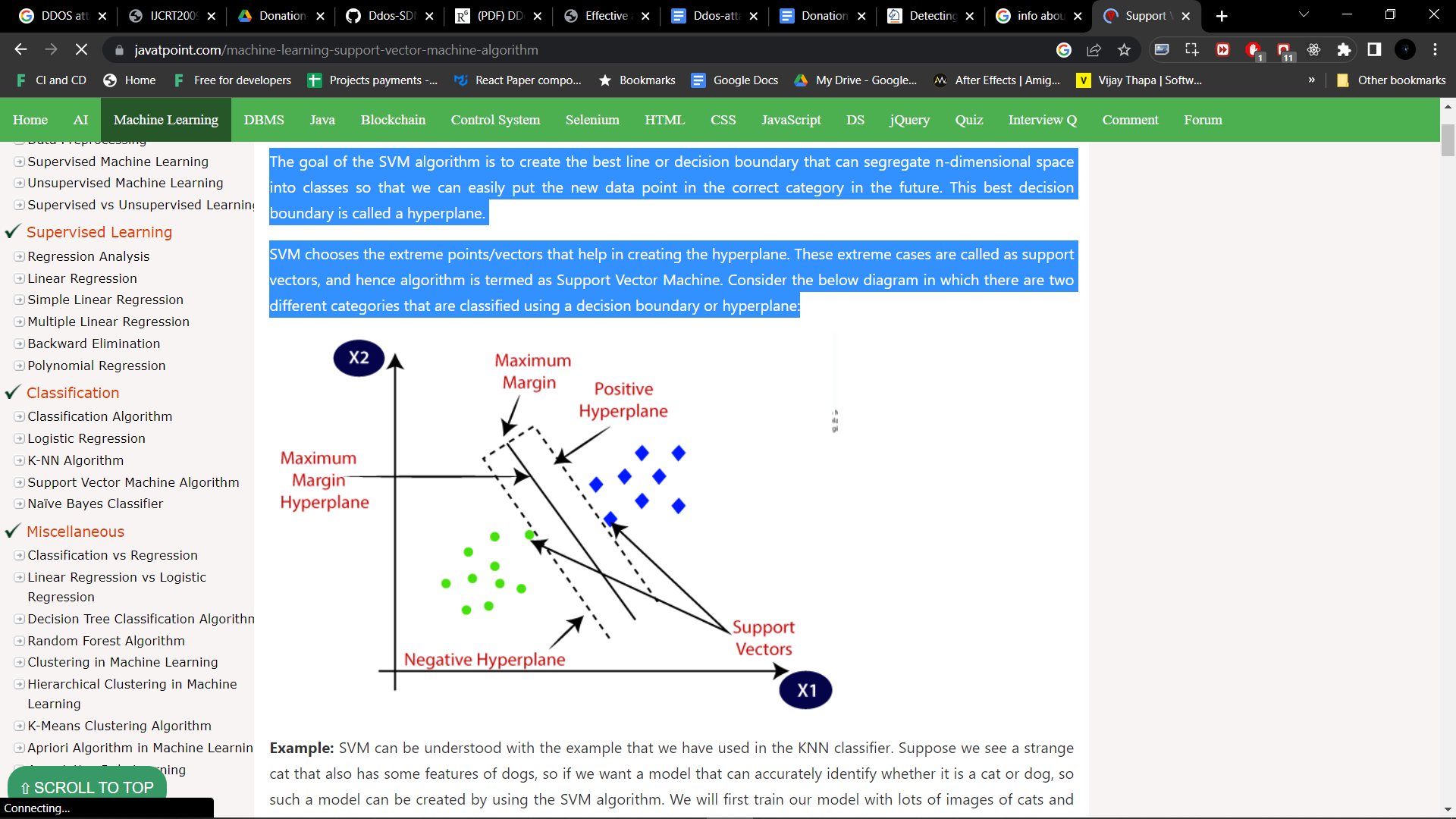
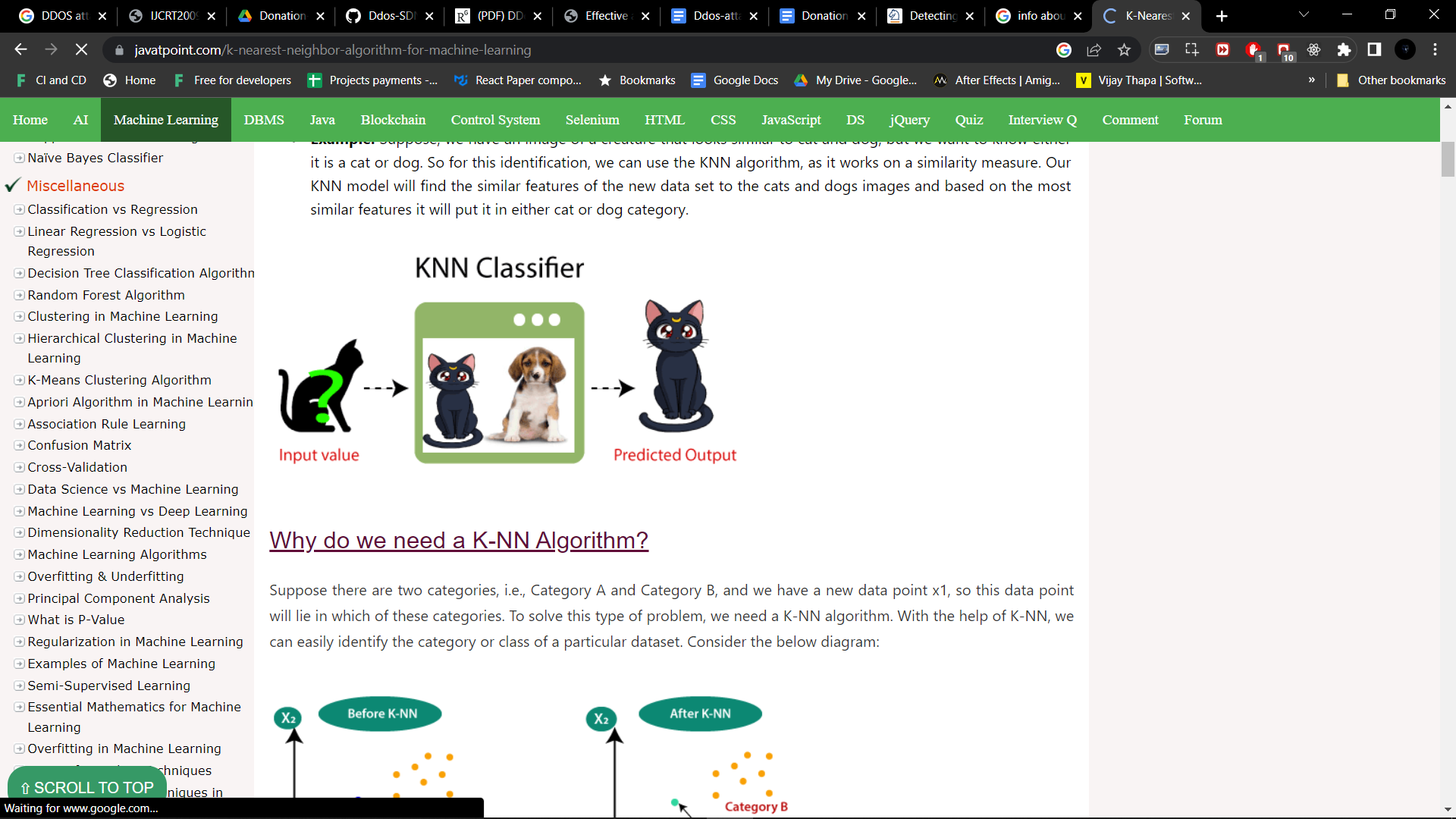


Fig. 6 SVM

1. KNN

One of the most popular methods for dividing a dataset into K groups is clustering. This approach refines the K initial cluster centers in a data set by each case that will enter the nearest cluster center after first identifying the initial cluster centres. To identify DDoS attacks of unknown sessions, developed a detection algorithm. Suggested a method for identifying DDoS attacks using the clustering algorithm of K-means, and they attained a 97.83% accuracy rate.



KNN Classifier

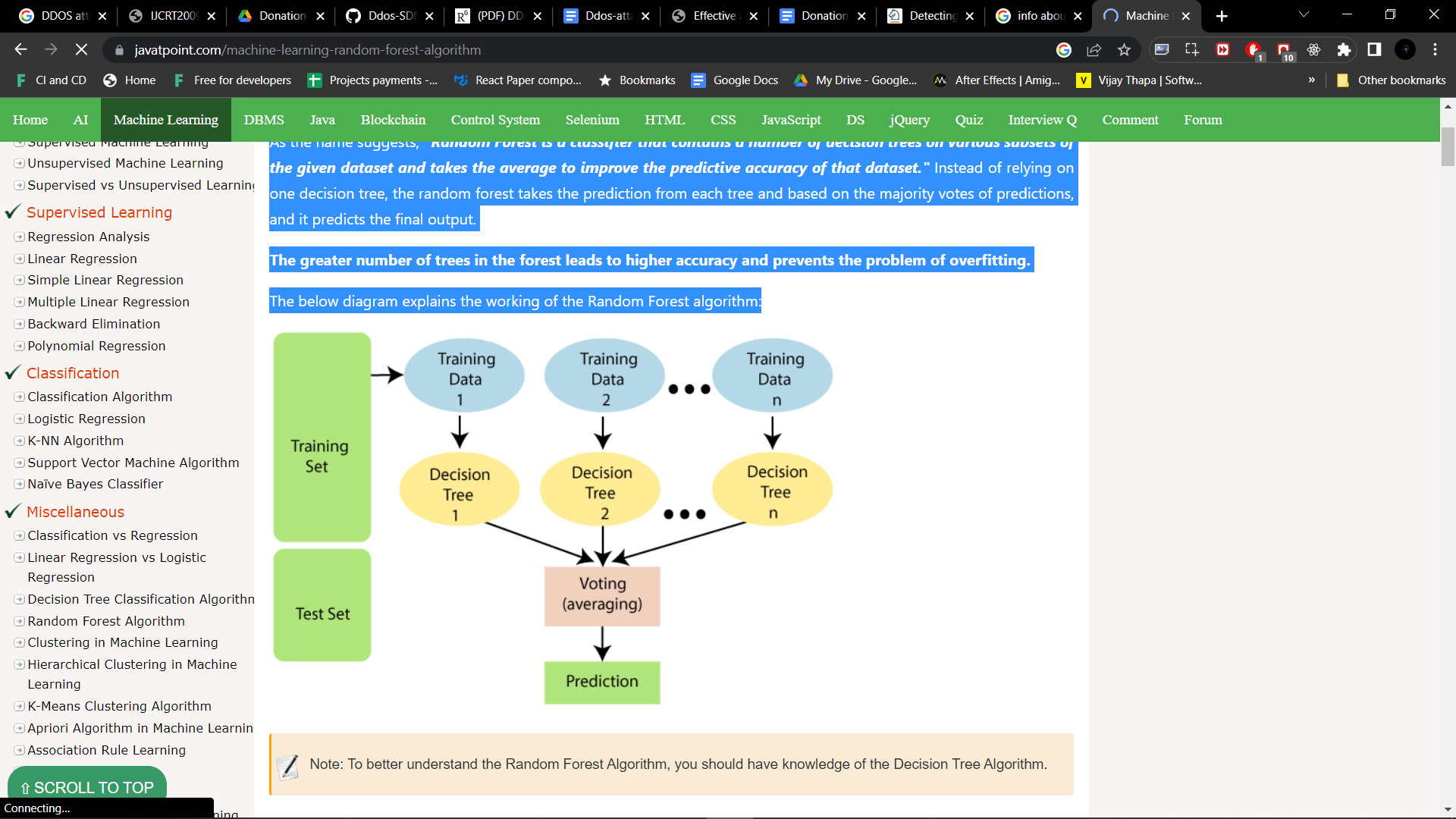
1. Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:



Random Forest Classifier

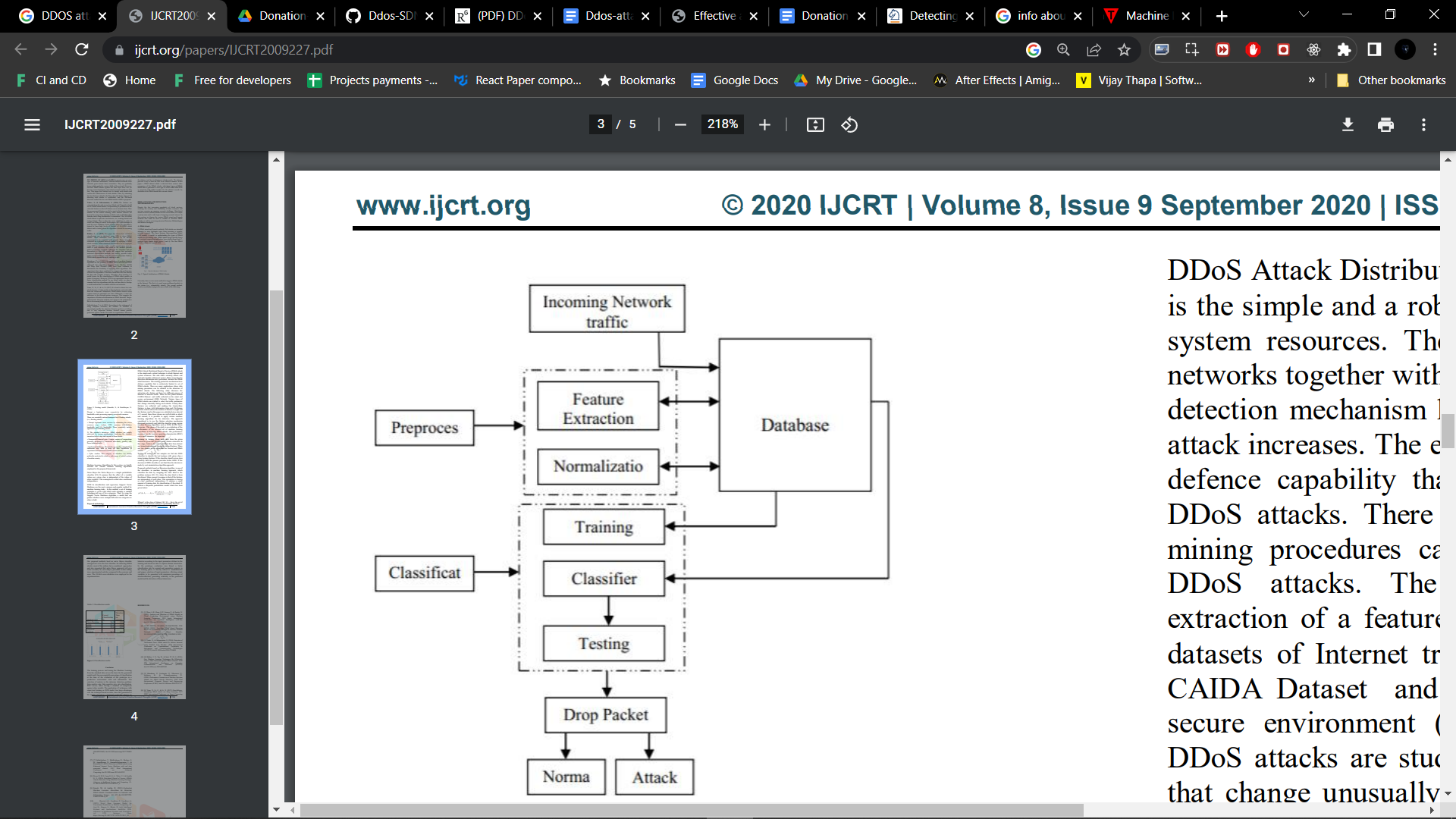
**4. SYSTEM DESIGN**

## **INTRODUCTION**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analysed, system design is the first of the three technical activities -design, code and test that is required to build and verify software. The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

## **UML DIAGRAM**

Unified Modelling Language is popular for its diagrammatic notations. Any complex system can be easily understandable by making some kind of pictures or diagrams. These diagrams have a better impact on our understanding in a better and simple way. To understand the UML, you need to form a conceptual model of the language, and this requires learning three major elements: the UML’s basic building blocks, the rules that dictate how those building blocks may be put together, and some common mechanisms that apply throughout blocks may be put together, and some common mechanisms that apply throughout the UML. Once you have grasped these ideas, you will be able to read UML models and create some basic ones. As you gain more experience in applying the UML. You can build on this conceptual model, using more advanced features of the language.



**4.2 Deep Learning based for IDS**

**4.2.3 Methodology**

Despite the fast increasing popularity of cloud services, ensuring the security and availability of data, resources and services remains an ongoing research challenge. Distributed denial of service (DDoS) attacks are not a new threat, It is major security issue and a wide topic of ongoing research interest. In this section we discuss the various DDoS intend and Launch methods that could be used to conduct or facilitate DDoS attacks, as well as reviewing intrusion Detection Methodologies and defence strategie .

**5.IMPLEMENTATION AND RESULTS**

**5.1 Introduction:**

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and changeover methods.

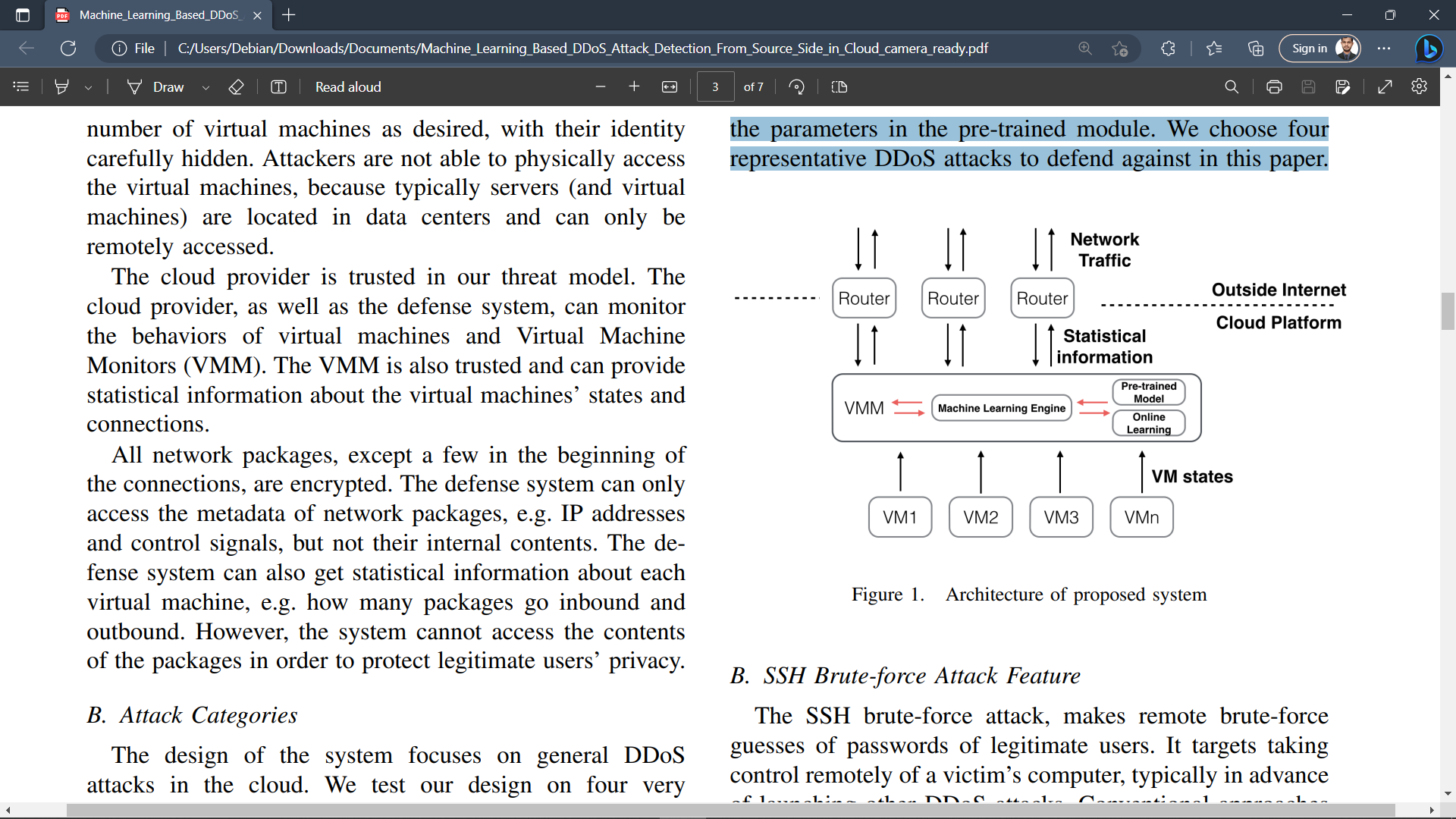
**5.2 Method of Implementation:**

This section contains the results of the experimental model using machine learning algorithms such as Random Forest, KNN, SVM algorithms. Jupyter tool is used for fetching the experimented results. Additionally, by subsampling the data, non-incremental learning methods can be used on large datasets. Weka also provides Hadoop and Spark compatibility as options for distributed data mining.

The distributed Jupyter Notebook Base package contains base "map" and "reduce" activities that are not dependent on any particular distributed platform.At a high level, the major distinction between Weka and the others is flexibility. Weka is a plug-and-play machine learning solution; it's packed in a.jar file and comes with a graphical user interface (GUI) via which you can perform most basic analyses and model development. Weka provides more instruction than the others, which are interactive shell languages, and running ML through Jupyter appears to be rather magical. As a result, Weka is less flexible than the others for statistical analysis and data exploration. The difference is that the others are programming languages with ML packages and libraries that you can import, whereas the others are not. On the other hand, Weka is a machine learning package. As you might expect, this implies that the others give you a lot more freedom to clean, analyse, and alter your data sets, as well as a lot more ability to control and tweak the underlying algorithms. Further, the performance of the experimental model is analysed. Jupyter is licensed under the GNU General Public License, making it free to use. Because it is written entirely in Python, it can run on nearly any modern computing platform. It is a collection of data preparation and modelling methods. Weka has some disadvantages in the sense that it is only capable of handling small datasets. An Out Of Memory problem happens whenever a set is larger than a few megabytes.

**5.3 Attack Features Selection and Extraction**

Before considering the feature selection and extraction, we first show the architecture of our system in Figure 1. TheVMM monitors the status of the virtual machines and gathers statistical network traffic information. The VMM inputs the gathered information to a machine learning engine. The machine learning engine feeds back whether a suspicion sanction is detected. If the suspicious behaviour is only detected on a single VM, terminating this VM is a reasonable way to mitigate the attack. If the suspicious behaviours are detected among multiple VMs, it is highly possible that a distributed DoS attack is ongoing. Therefore, cutting off the networkconnections of the suspicious servers can be done to defend against the attack.The machine learning engine has two modules: the pretrained module and the online learning module. The pretrained model is trained in advance to determine whether the virtual machine's actions are suspicious. On the other hand, the online learning module is trained in the background to update the pre-trained module. The online learning module takes monitored data as training samples to modify the parameters in the pre-trained module. We choose four representative DDoS attacks to defend against in this paper.



Architecture of Proposed Method

**5.3 Data Collection**

In our experiments, we collect network packages coming in and going out of the attacker virtual machine(s) for 2 hours. Four kinds of attacks are programmed to randomly start and end. Some of them may be started simultaneously. Our goal is detecting attacks, no matter which category an attack falls into. We evaluate both supervised learning and unsupervised learning algorithms. For supervised classification, we evaluate Linear Regression (LR), SVM (with linear, RBF or polynomial kernels), Random Forest algorithms. We also test unsupervised learning algorithms, k-means and GaussianMixture Model for Expectation-Maximization (GMM-EM). The time interval for collecting statistical features is 60 seconds. Table I shows the results of monitoring a single virtual machine with a pretrained learning module. Table II shows the results of simultaneously monitoring three virtual machines on three servers.

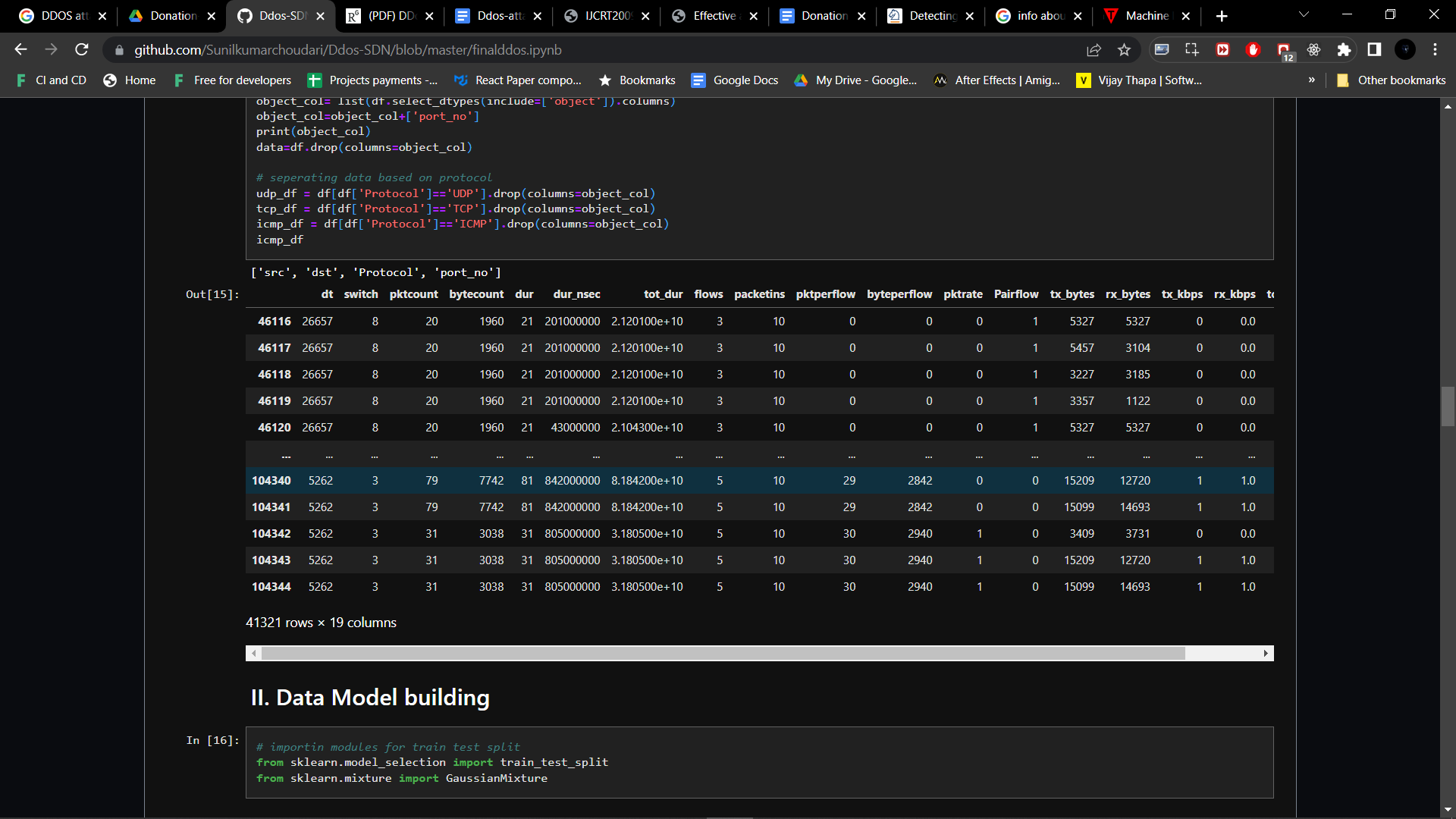


Table. 1

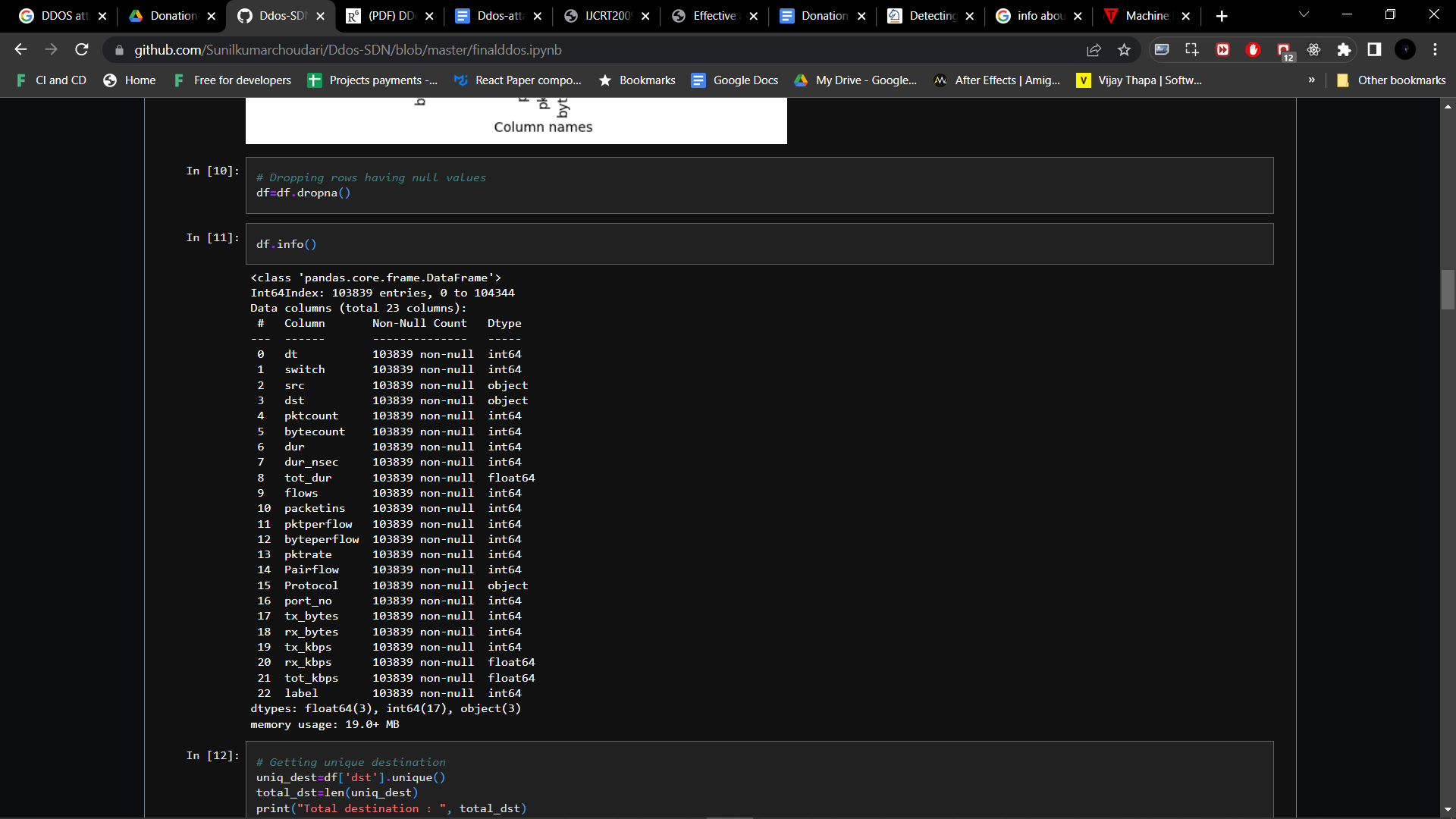


Table .2

**5.4 Data Processing**

dt switch pktcount bytecount dur dur\_nsec tot\_dur flows packetins pktperflow byteperflow pktrate Pairflow port\_no tx\_bytes rx\_bytes tx\_kbps rx\_kbps tot\_kbps label

count 104345.000000 104345.000000 104345.000000 1.043450e+05 104345.000000 1.043450e+05 1.043450e+05 104345.000000 104345.000000 104345.000000 1.043450e+05 104345.000000 104345.000000 104345.000000 1.043450e+05 1.043450e+05 104345.000000 103839.000000 103839.000000 104345.000000

mean 17927.514169 4.214260 52860.954746 3.818660e+07 321.497398 4.613880e+08 3.218865e+11 5.654234 5200.383468 6381.715291 4.716150e+06 212.210676 0.600987 2.331094 9.325264e+07 9.328039e+07 998.899756 1003.811420 2007.578742 0.390857

std 11977.642655 1.956327 52023.241460 4.877748e+07 283.518232 2.770019e+08 2.834029e+11 2.950036 5257.001450 7404.777808 7.560116e+06 246.855123 0.489698 1.084333 1.519380e+08 1.330004e+08 2423.471618 2054.887034 3144.437173 0.487945

min 2488.000000 1.000000 0.000000 0.000000e+00 0.000000 0.000000e+00 0.000000e+00 2.000000 4.000000 -130933.000000 -1.464426e+08 -4365.000000 0.000000 1.000000 2.527000e+03 8.560000e+02 0.000000 0.000000 0.000000 0.000000

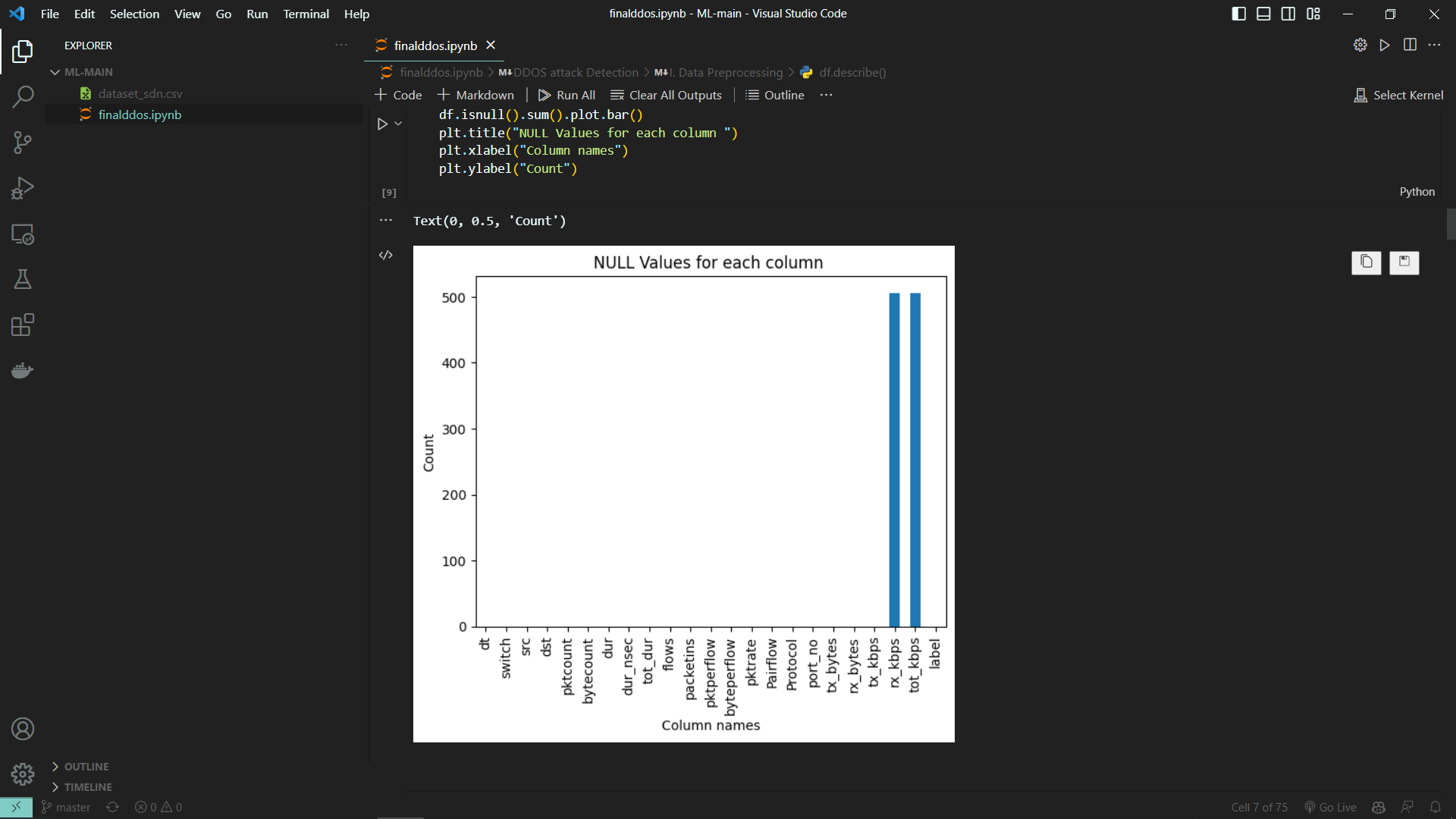
25% 7098.000000 3.000000 808.000000 7.957600e+04 127.000000 2.340000e+08 1.270000e+11 3.000000 1943.000000 29.000000 2.842000e+03 0.000000 0.000000 1.000000 4.743000e+03 3.539000e+03 0.000000 0.000000 0.000000 0.000000

50% 11905.000000 4.000000 42828.000000 6.471930e+06 251.000000 4.180000e+08 2.520000e+11 5.000000 3024.000000 8305.000000 5.521680e+05 276.000000 1.000000 2.000000 4.219610e+06 1.338339e+07 0.000000 0.000000 4.000000 0.000000

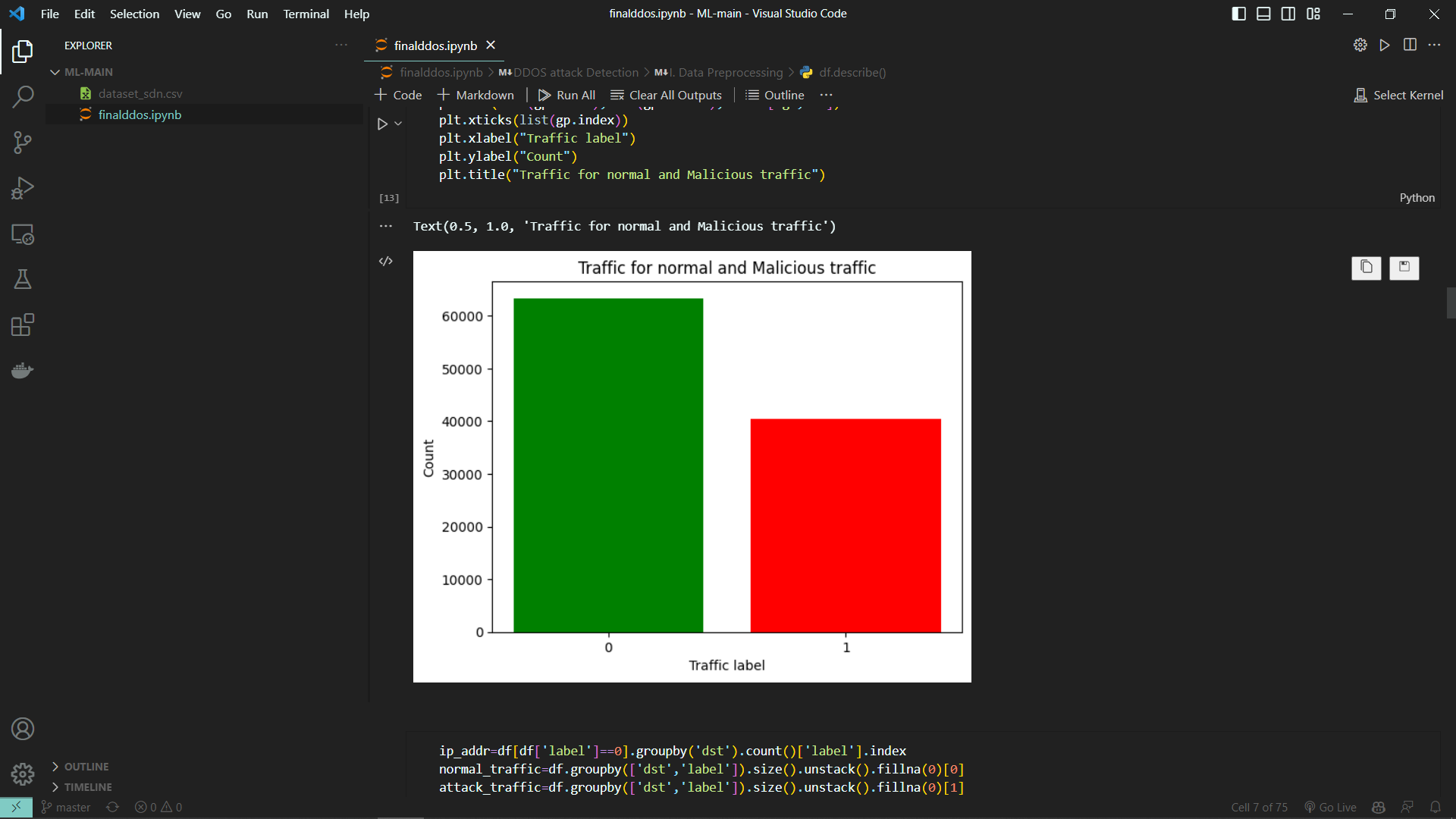
75% 29952.000000 5.000000 94796.000000 7.620354e+07 412.000000 7.030000e+08 4.130000e+11 7.000000 7462.000000 10017.000000 9.728112e+06 333.000000 1.000000 3.000000 1.356398e+08 1.439277e+08 251.000000 557.000000 3838.000000 1.000000

max 42935.000000 10.000000 260006.000000 1.471280e+08 1881.000000 9.990000e+08 1.880000e+12 17.000000 25224.000000 19190.000000 1.495387e+07 639.000000 1.000000 5.000000 1.269982e+09 9.905962e+08 20580.000000 16577.000000 20580.000000 1.000000

**5.5 Null Values Extraction**

****

**5.6 Traffic for Normal and Malicious Traffic**

****

ip\_addr=df[df['label']==0].groupby('dst').count()['label'].index

normal\_traffic=df.groupby(['dst','label']).size().unstack().fillna(0)[0]

attack\_traffic=df.groupby(['dst','label']).size().unstack().fillna(0)[1]

plt.barh(ip\_addr,normal\_traffic,color='g', label='Normal Traffic')

plt.barh(ip\_addr,attack\_traffic,color='r', label='Attack Traffic')

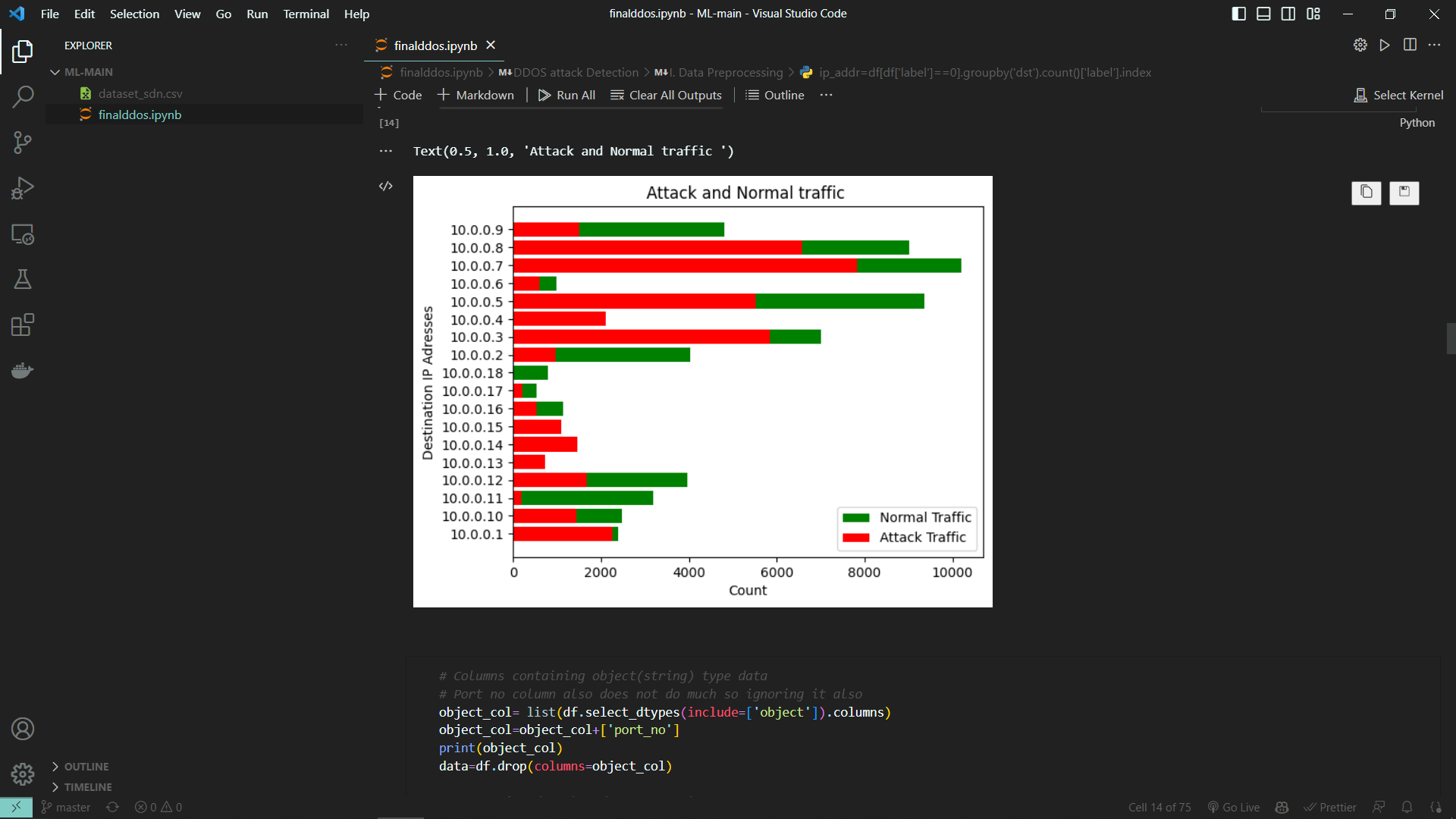
plt.legend()

plt.xlabel("Count")

plt.ylabel("Destination IP Adresses")

plt.title("Attack and Normal traffic ")

**5.7 Attack and Normal Traffic**

****

**5.9 Data Model Building**

object\_col= list(df.select\_dtypes(include=['object']).columns)

object\_col=object\_col+['port\_no']

print(object\_col)

data=df.drop(columns=object\_col)

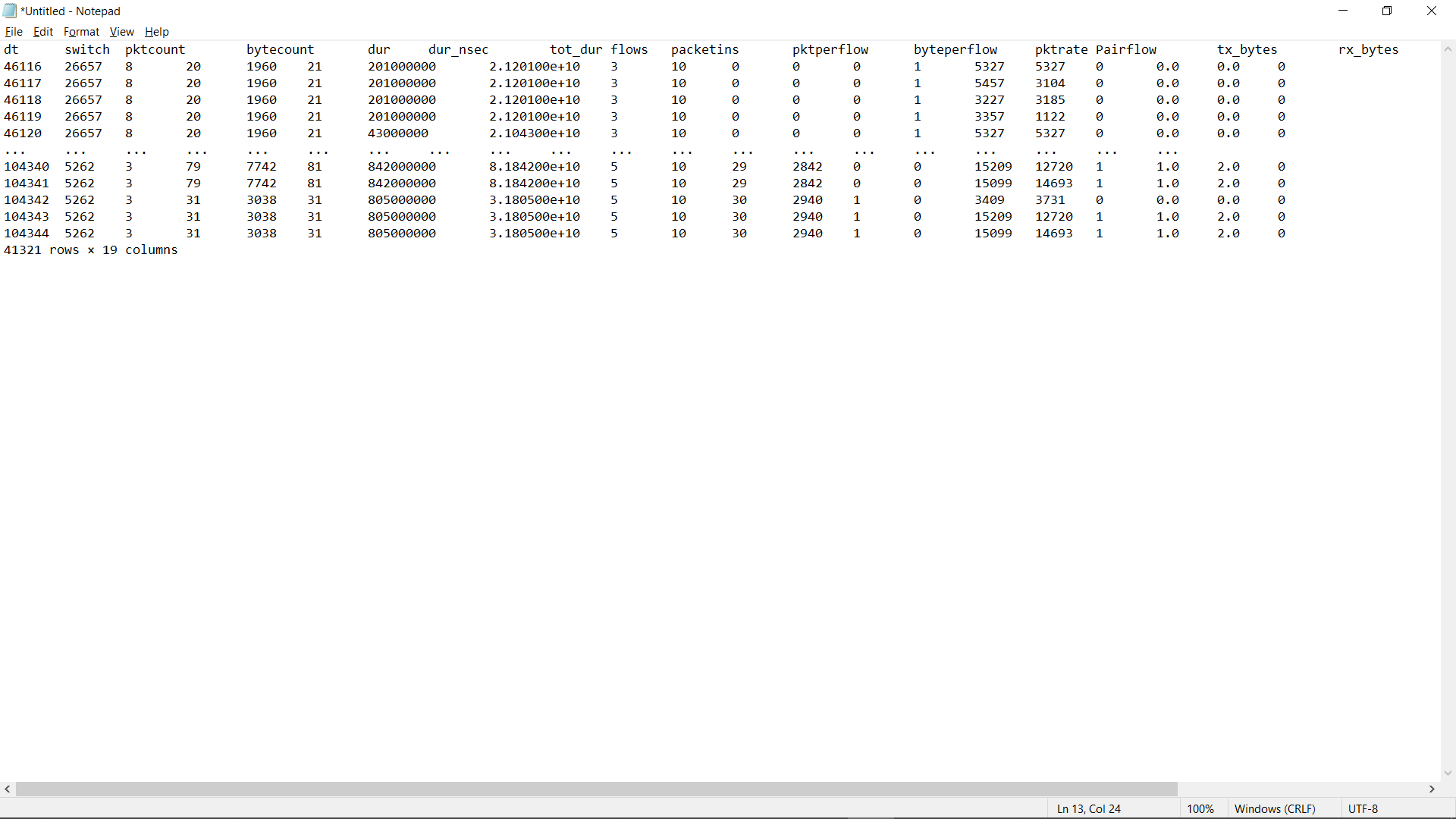
# seperating data based on protocol

udp\_df = df[df['Protocol']=='UDP'].drop(columns=object\_col)

tcp\_df = df[df['Protocol']=='TCP'].drop(columns=object\_col)

icmp\_df = df[df['Protocol']=='ICMP'].drop(columns=object\_col)

icmp\_df

****

**5.10 Linear Perceptron**

1. **UDP**

model = Perceptron(random\_state=67)

model.fit(udp\_train,udp\_train\_label)

model.score(udp\_train,udp\_train\_label)

1. **TCP**

model = Perceptron(random\_state=1)

model.fit(tcp\_train,tcp\_train\_label)

model.score(tcp\_train,tcp\_train\_label)

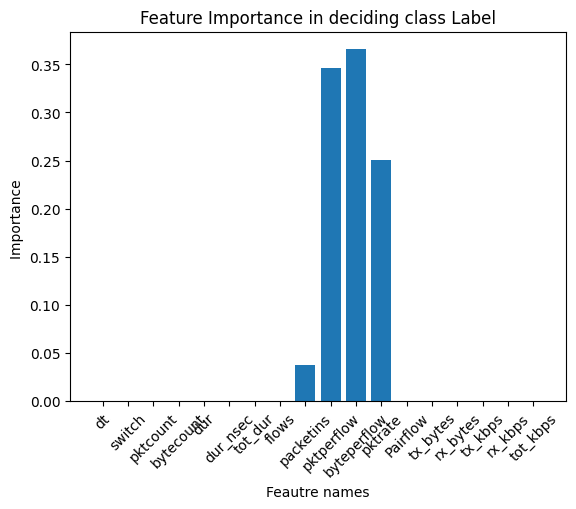
1. **ICMP**

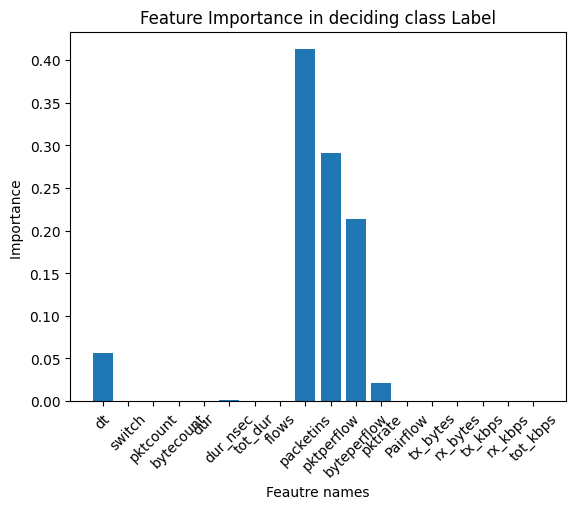
model = Perceptron(random\_state=1)

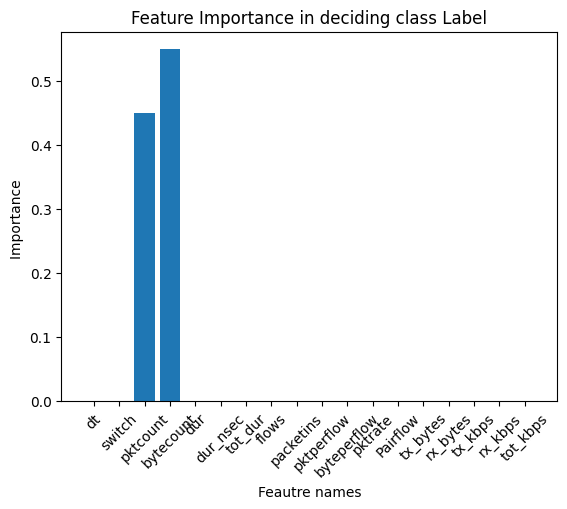
model.fit(icmp\_train,icmp\_train\_label)

model.score(icmp\_train,icmp\_train\_label)

**5.11 Feature importance in deciding class Label**

****

****

****

**5.12 Saving Random Forest Model**

**icmp\_file='icmp\_rf\_model.sav'**

**udp\_file='udp\_rf\_model.sav'**

**tcp\_file='tcp\_rf\_model.sav'**

**pickle.dump(udp\_rf, open(udp\_file, 'wb'))**

**pickle.dump(tcp\_rf, open(tcp\_file, 'wb'))**

**pickle.dump(icmp\_rf, open(icmp\_file, 'wb'))**

**6. CONCLUSION AND FUTURE WORK**

In this paper, we propose a DDoS attack detection system based on machine learning to prevent attacks on the source side in the cloud. We extract statistical features of four DDoS attacks and launch real attacks in lab settings for evaluation. Our proposed system is able to detect attacks with high accuracy (99.7%) and low false positives (< 0.07%). By detecting DDoS attacks at the source virtual machines in the cloud, we can ”nip the attacks in the bud” and also protect the cloud provider’s reputation.

We give some directions for future work:

• Combine different machine learning algorithms for better performance, especially unsupervised learning performance.

• Investigate more DDoS attacks and integrate their features into the current system

**REFERENCES**

[1] S. Ranjan, R. Swaminathan, M. Uysal, and E. W. Knightly, “Ddos-resilient scheduling to counter application layer attacks under imperfect detection.” in INFOCOM. Citeseer, 2006.

[2] Mcafee lab threat report. [Online]. Available: http://www.mcafee.com/us/resources/reports/ rp-quarterly-threat-q1-2015.pdf

[3] Worldwide infrastructure security report. [Online]. Available: https://www.arbornetworks.com/images/ documents/WISR2016\ EN\ Web.pdf

[4] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: architecture, applications, and approaches,” Wireless communications and mobile computing, vol. 13, no. 18, pp. 1587–1611, 2013.

[5] R. Miao, R. Potharaju, M. Yu, and N. Jain, “The dark menace: Characterizing network-based attacks in the cloud,” in Proceedings of the 2015 ACM Conference on Internet Measurement Conference. ACM, 2015, pp. 169–182.

[6] Ddos attack against servers accross the east coast usa. [Online]. Available: http://www.cnbc.com/2016/10/21/ major-websites-across-east-coast-knocked-out-in-apparent-ddos-attack. html

[7] S. Noh, C. Lee, K. Choi, and G. Jung, “Detecting distributed denial of service (ddos) attacks through inductive learning,” in International Conference on Intelligent Data Engineering and Automated Learning. Springer, 2003, pp. 286–295.

[8] T. Shon and J. Moon, “A hybrid machine learning approach to network anomaly detection,” Information Sciences, vol. 177, no. 18, pp. 3799–3821, 2007.

[9] N. B. Amor, S. Benferhat, and Z. Elouedi, “Naive bayes vs decision trees in intrusion detection systems,” in Proceedings of the 2004 ACM symposium on Applied computing. ACM, 2004, pp. 420–424.

[10] S. T. Zargar, J. Joshi, and D. Tipper, “A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks,” IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013.

[11] J. B. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra, “Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study,” in Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on. IEEE, 2001, pp. 609–622.

[12] T. Peng, C. Leckie, and K. Ramamohanarao, “Protection from distributed denial of service attacks using history-based ip filtering,” in Communications, 2003. ICC’03. IEEE International Conference on, vol. 1. IEEE, 2003, pp. 482–486.

[13] A. Yaar, A. Perrig, and D. Song, “Pi: A path identification mechanism to defend against ddos attacks,” in Security and Privacy, 2003. Proceedings. 2003 Symposium on. IEEE, 2003, pp. 93–107.

[14] J. Mirkovic, G. Prier, and P. Reiher, “Attacking ddos at the source,” in Network Protocols, 2002. Proceedings. 10th IEEE International Conference on. IEEE, 2002, pp. 312–321.

[15] ——, “Source-end ddos defense,” in Network Computing and Applications, 2003. NCA 2003. Second IEEE International Symposium on. IEEE, 2003, pp. 171–178. [16] T. M. Gil and M. Poletto, “Multops: A data-structure for bandwidth attack detection.” in USENIX Security Symposium, 2001, pp. 23–38.

[17] Mananet reverse firewall. [Online]. Available: http://www. cs3-inc.com/MANAnet.htm