**NETWORK INTRUSION DETECTION SYSTEM**

**PROJECT REPORT SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER ENGINEERING**

**(2010 - 2014)**

**SUBMITTED BY:**

**AKASH AHUJA 211/CO/10**

**AKHILESH CHAUDHARY 214/CO/10**

**AMIT KUMAR 220/CO/10**

****

**UNDER THE GUIDANCE OF**

**PROF SATISH CHAND**

**DEPARTMENT OF COMPUTER ENGINERING**

**NETAJI SUBHAS INSTITUTE OF TECHNOLOGY**

**UNIVERSITY OF DELHI, 2014**



**CERTIFICATE**

This is to certify that the dissertation entitled “Network Intrusion Detection System” being submitted by AKASH AHUJA, AKHILESH CHAUDHARY and AMIT KUMAR in the Department of Computer Engineering, Netaji Subhas Institute of Technology, Delhi, for the award of the degree of “Bachelor of Engineering” is a bona fide record of the work carried out by them. They have worked under my guidance and supervision and have fulfilled the requirements for the submission of this report, which has reached the requisite standard.

**Prof. SATISH CHAND**

Department of Computer Engineering

Netaji Subhas Institute of Technology

**ACKNOWLEDGEMENT**

We take this unique opportunity to express our heartfelt thanks and gratitude to our respected guide, **Prof SATISH CHAND**, Department of Computer Engineering, NSIT who kindly consented to be our guide for this project.

We thank him for the precious time he devoted to us, for his expert guidance from the commencement, for the kind attitude and the resources he arranged for us. It is only because of him that we have been able to successfully complete this project.

We also owe our thanks to all the faculty members for their constant support and encouragement.

We are grateful to all our friends for providing critical feedback & support whenever required.

AKASH AHUJA AKHILESH CHAUDHARY AMIT KUMAR

211/CO/10 214/CO/10 220/CO/10

**ABSTRACT**

There are many risks of network attacks under the Internet environment. Internet Security is a vital issue and therefore, the intrusion detection is one major research problem for business and personal networks to resist external attacks. A Network Intrusion Detection System (NIDS) is a software application that monitors the network or system activities for malicious activities and unauthorized access to devices. The goal of designing a NIDS is to protect data’s confidentiality and integrity. Our project focuses on these issues with the help of Machine Learning.

This project includes the implementation of different machine learning algorithms including Linear regression, K-Means Clustering and Artificial Neural Networks to automatically generate the rules for classify network activities. A comparative analysis of these techniques to detect intrusions has also been made. To learn the patterns of the attacks, NSL-KDD dataset has been used.

**KEYWORDS**: Network Intrusion Detection System (IDS), Machine Learning Techniques

**INTRODUCTION**

* An Intrusion Detection System (IDS) is a software application that monitors network or system activities for malicious activities and unauthorized access to devices.
* IDS come in a variety of “flavors” and approach the goal of detecting suspicious traffic in different ways. There are network based (NIDS) and host based (HIDS) intrusion detection systems
* There are IDS that detect based on comparing traffic patterns against a baseline and looking for anomalies. There are IDS that simply monitor and alert and there are IDS that perform an action or actions in response to a detected threat. We’ll cover each of these briefly.
* Network Intrusion Detection Systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. Ideally you would scan all inbound and outbound traffic , however doing so might create a bottleneck that would impair the overall speed of the network.
* When designing a IDS, the mission is to protect the data’s
  + Confidentiality – read
  + Integrity – read/write
* Two major levels/types
  + Misuse-based IDS
  + Anomaly-based IDS

**TYPES OF INTRUSIONS**

**Eavesdropping**

In general, the majority of network communications occur in an unsecured or

"cleartext" format, which allows an attacker who has gained access to data paths in your network to "listen in" or interpret (read) the traffic. When an attacker is eavesdropping on your communications, it is referred to as sniffing or snooping. The ability of an eavesdropper to monitor the network is generally the biggest security problem that administrators face in an enterprise. Without strong encryption services that are based on cryptography, your data can be read by others as it traverses the network.

### Data Modification

After an attacker has read your data, the next logical step is to alter it. An attacker can modify the data in the packet without the knowledge of the sender or receiver. Even if you do not require confidentiality for all communications, you do not want any of your messages to be modified in transit. For example, if you are exchanging purchase requisitions, you do not want the items, amounts, or billing information to be modified.

### Identity Spoofing (IP Address Spoofing)

Most networks and operating systems use the IP address of a computer to identify a valid entity. In certain cases, it is possible for an IP address to be falsely assumed— identity spoofing. An attacker might also use special programs to construct IP packets that appear to originate from valid addresses inside the corporate intranet.

### Compromised-Key Attack

A key is a secret code or number necessary to interpret secured information. Although obtaining a key is a difficult and resource-intensive process for an attacker, it is possible. After an attacker obtains a key, that key is referred to as a compromised key.

An attacker uses the compromised key to gain access to a secured communication without the sender or receiver being aware of the attack. With the compromised key, the attacker can decrypt or modify data, and try to use the compromised key to compute additional keys, which might allow the attacker access to other secured communications.

### Sniffer Attack

A sniffer is an application or device that can read, monitor, and capture network data exchanges and read network packets. If the packets are not encrypted, a sniffer provides a full view of the data inside the packet. Even encapsulated (tunneled) packets can be broken open and read unless they are encrypted and the attacker does not have access to the key.

Using a sniffer, an attacker can do any of the following:

* Analyze your network and gain information to eventually cause your network to crash or to become corrupted.
* Read your communications.

### Application-Layer Attack

An application-layer attack targets application servers by deliberately causing a fault in a server's operating system or applications. This results in the attacker gaining the ability to bypass normal access controls. The attacker takes advantage of this situation, gaining control of your application, system, or network, and can do any of the following:

* Read, add, delete, or modify your data or operating system.
* Introduce a virus program that uses your computers and software applications to copy viruses throughout your network.
* Introduce a sniffer program to analyze your network and gain information that can eventually be used to crash or to corrupt your systems and network.
* Abnormally terminate your data applications or operating systems.
* Disable other security controls to enable future attacks.

### Password-Based Attacks

A common denominator of most operating system and network security plans is password-based access control. This means your access rights to a computer and network resources are determined by who you are, that is, your user name and your password.

Older applications do not always protect identity information as it is passed through the network for validation. This might allow an eavesdropper to gain access to the network by posing as a valid user.

When an attacker finds a valid user account, the attacker has the same rights as the real user. Therefore, if the user has administrator-level rights, the attacker also can create accounts for subsequent access at a later time.

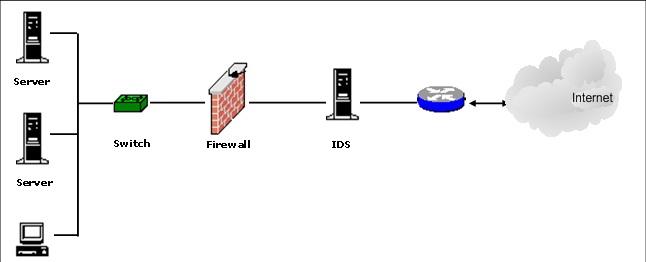
After gaining access to your network with a valid account, an attacker can do any of the following:

* Obtain lists of valid user and computer names and network information.
* Modify server and network configurations, including access controls and routing tables.
* Modify, reroute, or delete your data.

### Man-in-the-Middle Attack

As the name indicates, a man-in-the-middle attack occurs when someone between you and the person with whom you are communicating is actively monitoring, capturing, and controlling your communication transparently. For example, the attacker can re-route a data exchange. When computers are communicating at low levels of the network layer, the computers might not be able to determine with whom they are exchanging data.

Man-in-the-middle attacks are like someone assuming your identity in order to read your message. The person on the other end might believe it is you because the attacker might be actively replying as you to keep the exchange going and gain more information. This attack is capable of the same damage as an application-layer attack, described later in this section.



**TYPES OF CURRENT IDS**

**Misuse Detection**

* Record the specific patterns of intrusions.
* Monitor current event sequences and pattern matching
* Report the matched events as intrusions
* The IDS analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS look for a specific attack that has already been documented.
* Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against.
* Examples: Expert Systems, Signature Analysis, State Transition Analysis

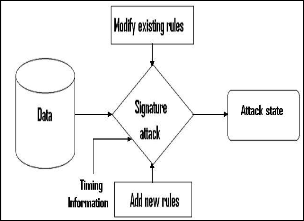
**Anomaly Detection**

* Establish the normal behavior profiles.
* Compare current activities with normal profiles.
* Report significant deviations as intrusions.
* The system administrator defines the baseline, or normal, state of the networks traffic load, breakdown, protocol, and typical packet size.
* The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies.
* Examples: Statistical Analysis, Expert System

**PROBLEM WITH CURRENT IDS**

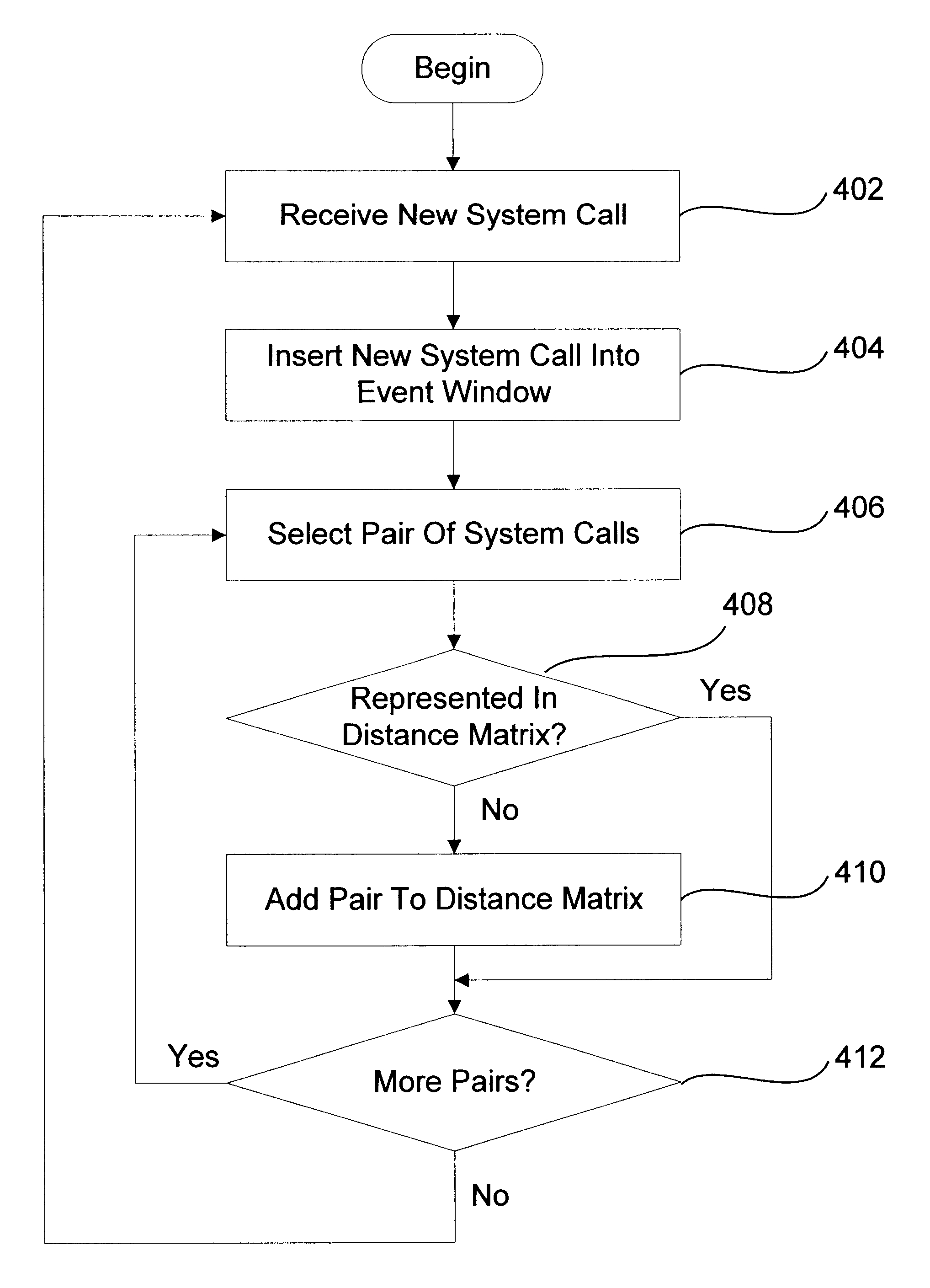
**Misuse Detection**

* **The known patterns have to be hand coded.**
* **Unable to detect unknown intrusions.**

****

**Anomaly Detection**

* **It relies upon in the selecting the system features.**
* **The sequential interrelation between transactions has to be studied.**

****

**WHAT IS MACHINE LEARNING**

**Machine learning**, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data. For example, a machine learning system could be trained on email messages to learn to distinguish between spam and non-spam messages. After learning, it can then be used to classify new email messages into spam and non-spam folders.

The core of machine learning deals with representation and generalization. Representation of data instances and functions evaluated on these instances are part of all machine learning systems. Generalization is the property that the system will perform well on unseen data instances; the conditions under which this can be guaranteed are a key object of study in the subfield of computational learning theory.

There are a wide variety of machine learning tasks and successful applications. Optical character recognition, in which printed characters are recognized automatically based on previous examples, is a classic example of machine learning.

A core objective of a learner is to generalize from its experience. Generalization in this context is the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) and the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases.

The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory. Because training sets are finite and the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common.

In addition to performance bounds, computational learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

There are many similarities between machine learning theory and statistical inference, although they use different terms.

**WHY MACHINE LEARNING**

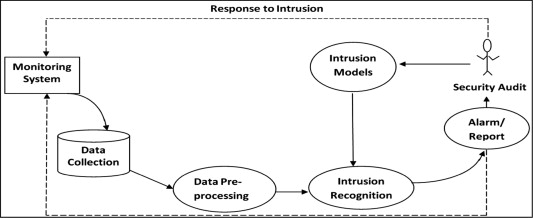
* Normal and Intrusion activities leave evidence in audit data.
* The pattern of the normal activities and malicious activities can be learned and distinguished.
* Machine learning can result in higher detection rates, lower false alarm rates.
* On dynamic data, the model can be updated and maintained.
* Huge amount of data featuring network activities is available and can be generated nowadays.

**MOTIVATION**

This research focuses on solving the issues in intrusion detection communities that can help the administrator to make pre-processing, classification, labeling of data and to mitigate the outcome of Distributed Denial of Service Attacks. Since, the network administrator feels difficult to pre-process the data. Due to the overwhelming growth of attacks which makes the task hard, attacks can be identified only after it happens. To overcome this situation, frequent updating of profiles is needed. Reduced workload of administrator increases the detection of attacks. Data mining includes many different algorithms to accomplish the desired tasks. All of these algorithms aims to fit a model to the prescribed data and even analyzes the data and simulate a model which is closest to the data being analyzed.

**METHODOLOGY**

Some of the open issues have been taken to detect attacks over the network. To achieve this, the framework of the proposed methods has been discussed below. The entire framework of the proposed methodology in Intrusion Detection System is described in Fig.

****

**DATA FOR INTRUSION DETECTION**

Attacks can be described as

•*Dos attack* – It is a kind of attack where the attacker makes processing time of the resources and memory busy so as to avoid legitimate user from accessing those resources.

• *U2R attack* – Here the attacker sniffs the password or makes some kind of attack to access the particular host in a network as a legitimate user. They can even promote some vulnerability to gain the root access of the system.

• *R2L attack* – Here the attacker sends a message to the host in a network over remote system and makes some vulnerability.

• *Probe attack* – Attacker will scan the network to gather information and would make some violation in the future.

KDD Cup 99 data set contains 23 attack types and their names are shown in Table and its features are grouped as,

1.Basic features

It encompasses all the attributes of TCP/IP connection and leads to delay in detection.

2. Traffic features

It is evaluated in accordance with window interval & two features as same host and same service.

(a)Same host feature

It examines the number of connections for the past 2 s that too from the same destination host. In other words, the probability of connections will be done in a specific time interval.

(b) Same service feature

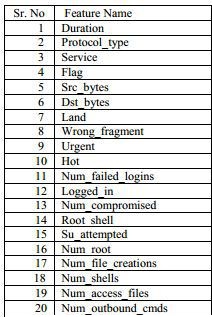
It examines the number of connections in a particular time interval that too posses same service.

3. Content features

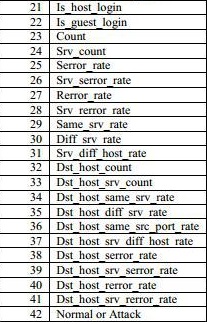
Dos & probe attack have frequent intrusion sequential patterns than the R2L & U2R. Because these two attacks include many connections to several hosts at a particular time period whereas R2L and U2R perform only a single connection. To detect these types of attacks, domain knowledge is important to access the data portion of the TCP packets. Ex. Failed login, etc. these features are called as content features.

* We are using NSL-KDD dataset divided into two parts, one for training and other for testing.
* There are total 42 features in the dataset including Protocol\_type, Service, Flag, Source bytes, Destination bytes, Num\_failed\_logins, Root shells etc.
* The last column contains the output denoting a normal activity or an attack.
* Different types of attacks are included like DoS, U2R, R2L, Probing.

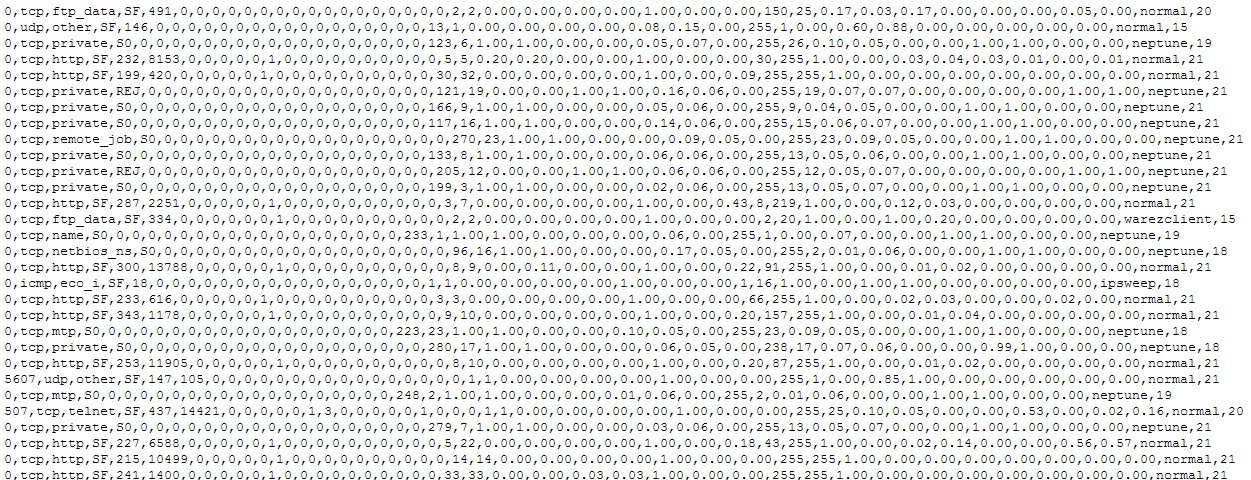
**FEATURES OF DATASET**

****

**MORE FEATURES…**

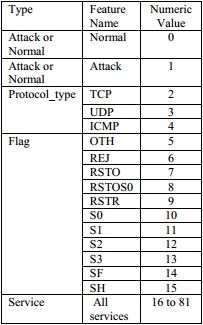
****

**A LOOK AT THE DATASET**

****

**DATA PREPOCESSING**

* As you can see, the dataset contains nominal values also   
  and to train a model we need all numerical values.
* Here is the transformation table that we used.
* Dataset is very large and there is a large variation between   
  values, Data Normalization is also required for better perfor  
  -mance.
* Mean normalization is used that makes the values of each   
  feature in the data have zero-mean and unit-variance

****

**MEAN NORMALISATION**

 Normalization refers to the creation of shifted and scaled versions of statistics, where the intention is that these **normalized values** allow the comparison of corresponding normalized values for different datasets in a way that eliminates the effects of certain gross influences, as in an anomaly time series. Some types of normalization involve only a rescaling, to arrive at values relative to some size variable. In terms of levels of measurement, such ratios only make sense for *ratio* measurements (where ratios of measurements are meaningful), not *interval* measurements (where only distances are meaningful, but not ratios).

# Normalization by Mean

Assume that there are *n* rows with seven variables (columns), A, B, C, D, E, F and G, in the data. We use variable E as an example in the calculations below. The remaining variables in the rows are normalized in the same way.

**Without rescaling (Baseline variable = None)**

The normalized value of *ei* for variable E in the ith row is calculated as:

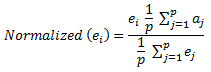
images/n_mean_ekv_without.gif

Where,

*p* = the number of records used to calculate the mean

**Rescaling by a baseline variable**

If we select variable A as baseline variable, the normalized value of *e*i for variable E in the ith row is calculated as:



where

*p* = the number of rows used to calculate the mean

*a*j = the value for variable A in the jth record

**RESULTS AND DISCUSSIONS**

KDD Cup 99 data set has been used in this research of which 60% is treated as training data and 40% is considered as testing data. The proposed framework has been implemented in MatLab10 and Java using data mining techniques. Performance of four proposed methods such as,

• Classification of network data using EDADT algorithm.

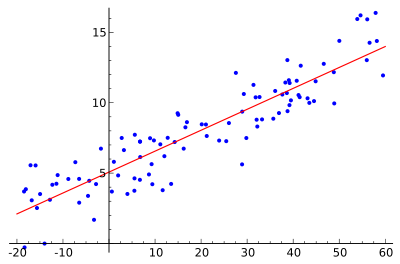
• Proposed Hybrid IDS.

• Performance of Semi-Supervised Approach for IDS and,

• Mitigating DDoS attacks using Varying Clock Drift Mechanism.

1. **LINEAR REGRESSION**

* Study of linear relationship between an output variable and one or more input features.
* For a linear model, our hypothesis of the form,  
   **(x) = + x** or **(x) = x.**



* We have to adjust and so that (x) is close to y. For that we define an error function,   
   J() = (1 / 2m) \*
* To minimize the error, we use Gradient Descent Algorithm,  
    
   Repeat until convergence {
* = – α for j=0 and j=1

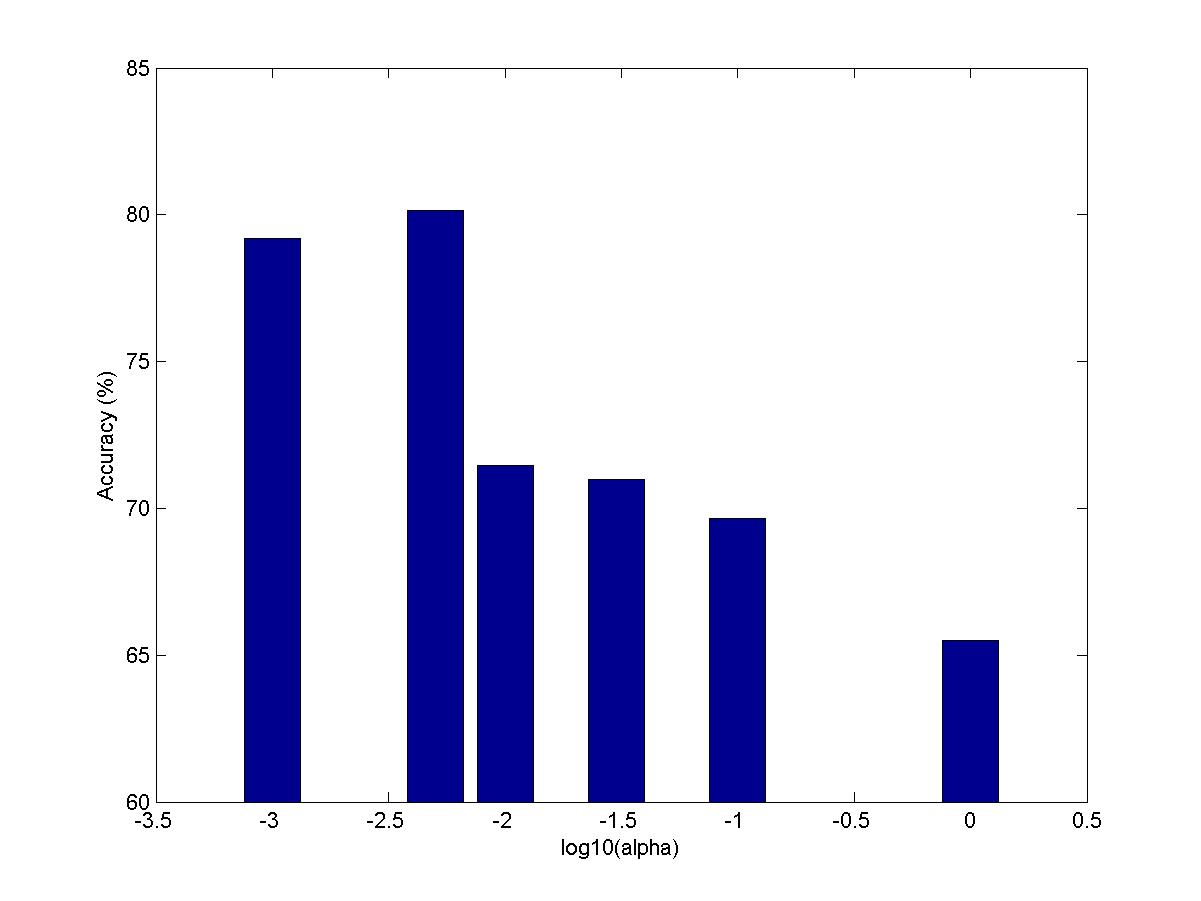
}

RESULTS:

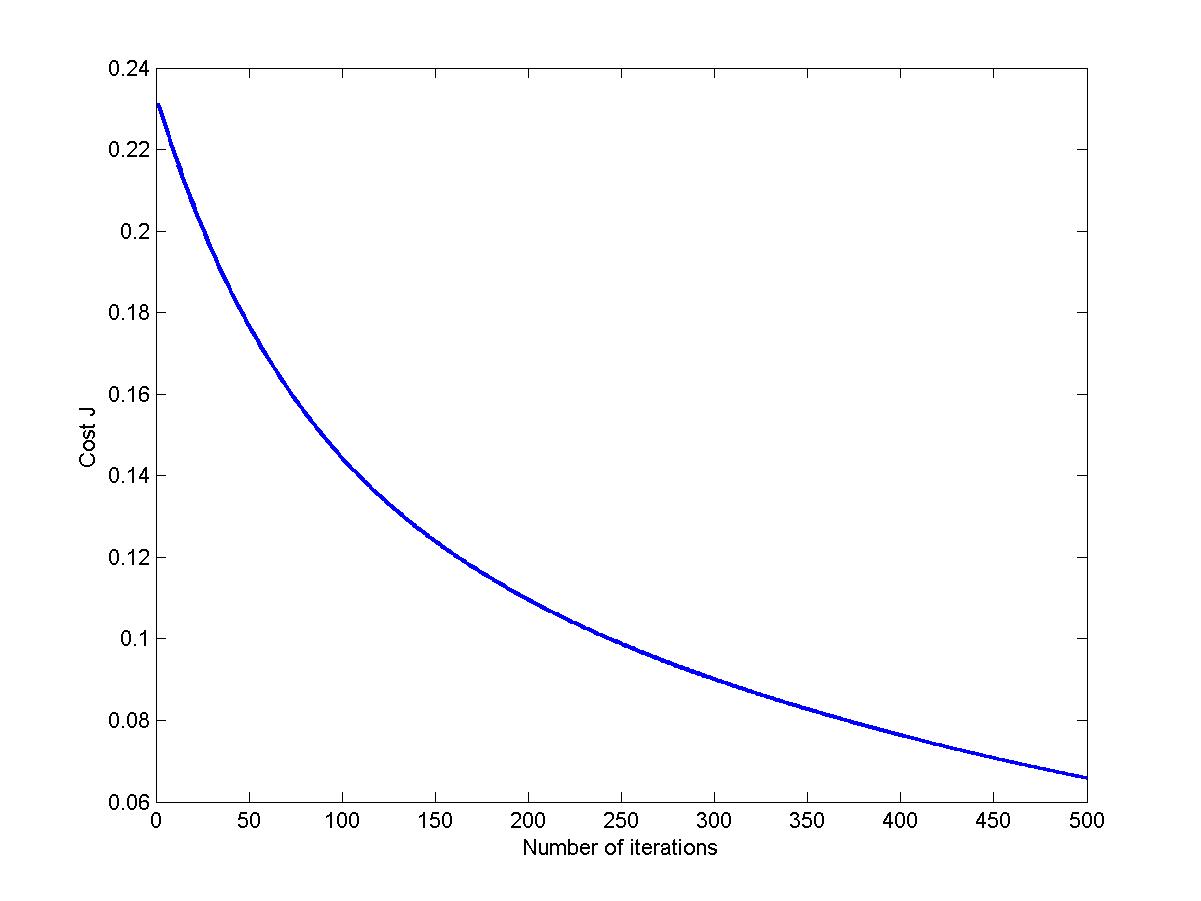
* For different values of alpha, linear regression algorithm was trained to find the best result.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Alpha(α) | 0.001 | 0.005 | 0.01 | 0.03 | 0.1 | 1 |
| Accuracy (%) | 79.20 | 80.14 | 71.47 | 70.98 | 69.65 | 65.5 |

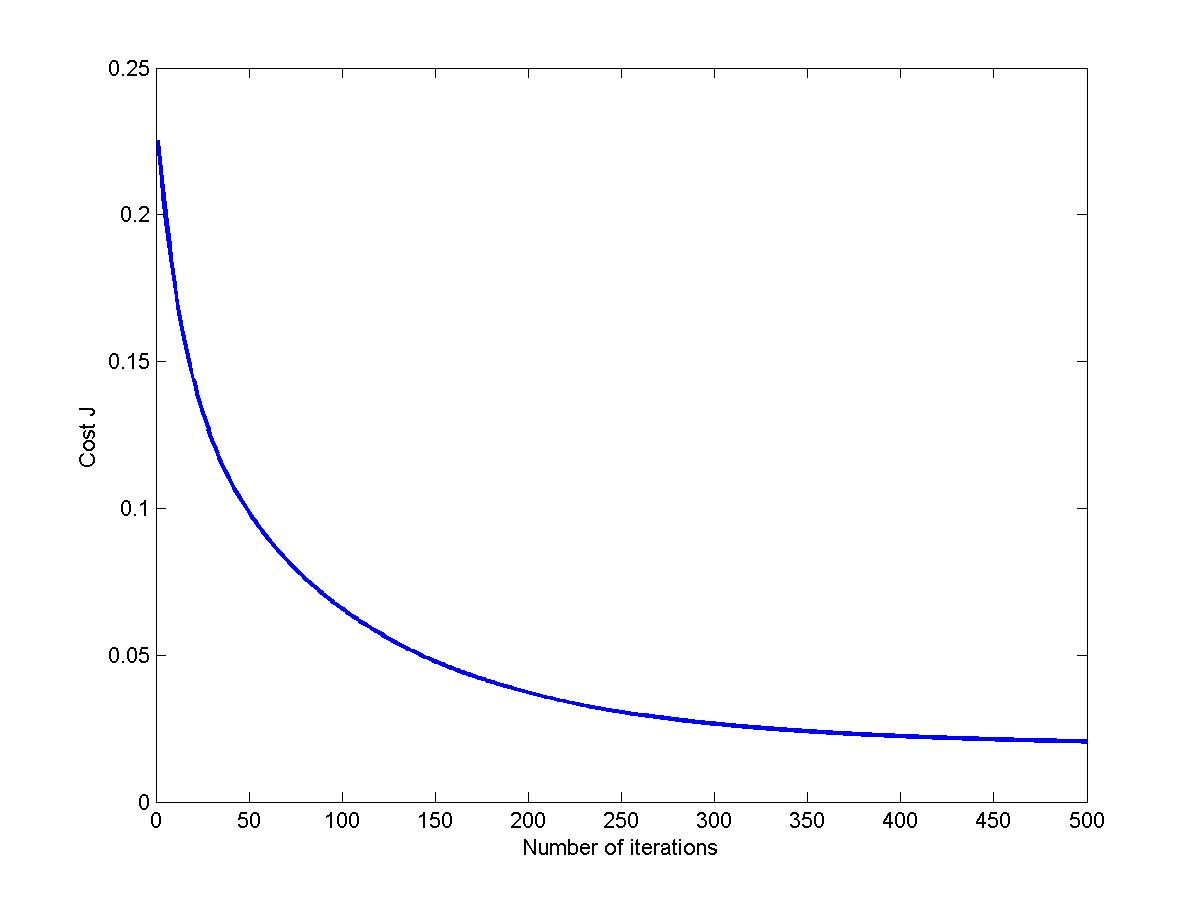
* Bar plot



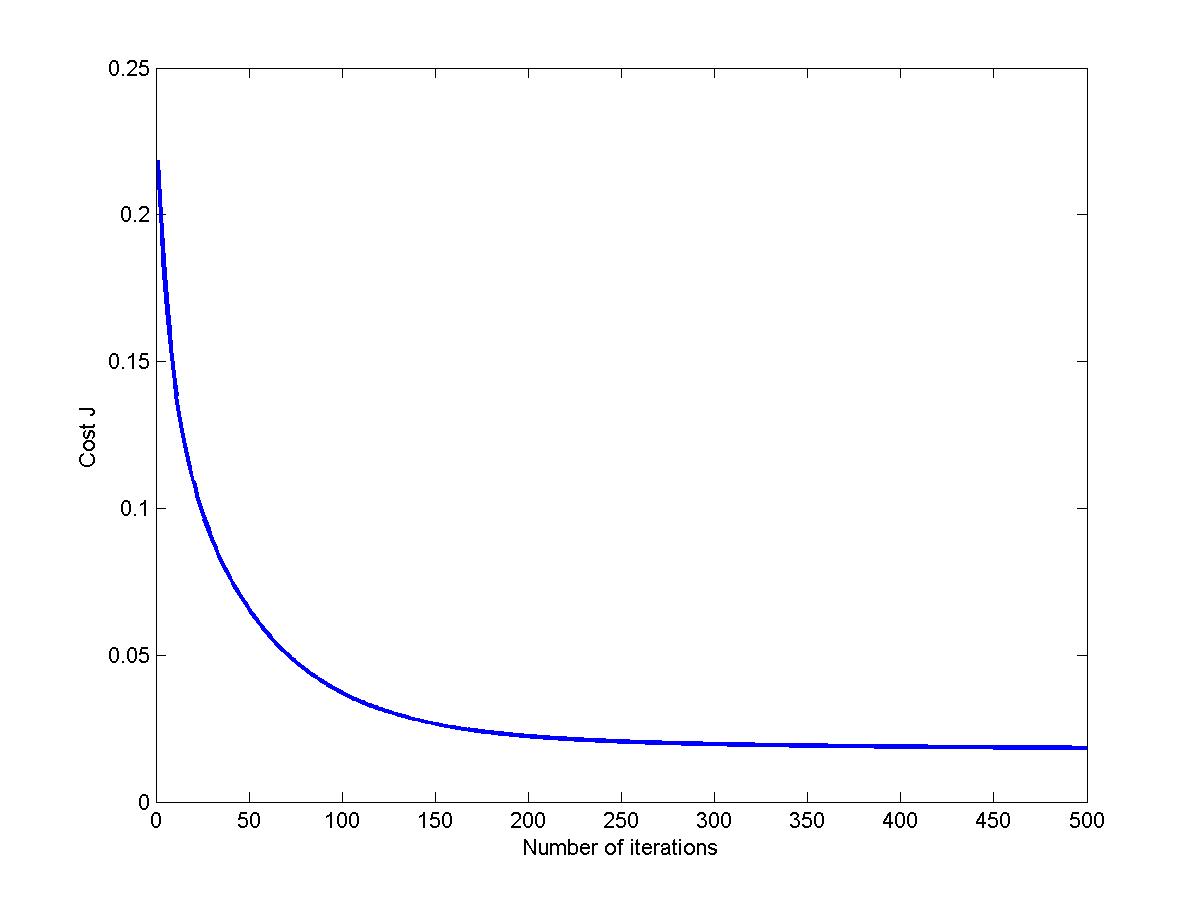
* Cost Variations (alpha = 0.001)



* Cost Variation (alpha = 0.005)



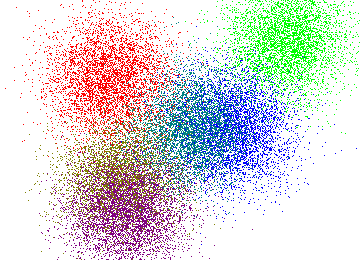
* Cost Variation (alpha = 0.01)



1. **K-Means Clustering**

* It is an algorithm to classify or to group your objects based on attributes/features into K number of groups.
* The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.
* **Algorithm  
   Randomly initialize K cluster centroids  
   Repeat {  
   for i=1 to m  
   c(i) := j that minimizes ,  
   for k=1 to K  
   = average mean of points assigned to   
   cluster k**

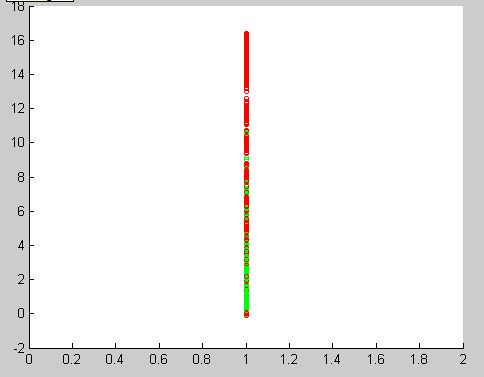
**}**

****

**RESULTS :**

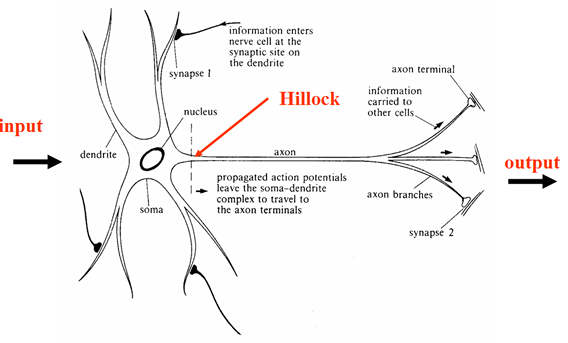
* We used two clusters, one to classify attacks and other for normal patterns.
* Random centroid were selected and there results were compared.
* Best centroid value was found to be   
  X(i = 84252, : ) and X( j = 17428, : )
* Accuracy was found 67.53%.

Clusters plot:



1. **Neural Networks**

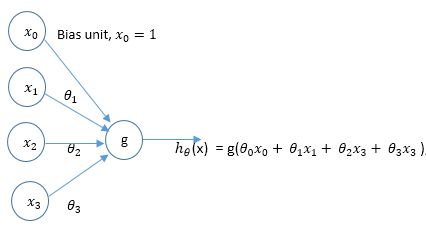
* Neural network is a machine learning approach that models human brain and consists of a number of artificial neurons.
* Basic computational units in a neuron:
  + Dendrites (Input)
  + Cell body
  + Axon (Output)
* A neuron receives input from other neurons. Inputs sum. Once input exceeds a critical value, the neuron discharge a spike – an electrical pulse that travels from the body, through axon, to next neurons.

****

**Neuron Model**

* Our neuron model is similar to brain neuron.
* In neural model, we use a term called activation function which is a function used to transform the activation level of a unit (neuron) into an output signal.
* Here we use sigmoid function as our activation function,

**This is a single neuron.**

****

**(x) =**

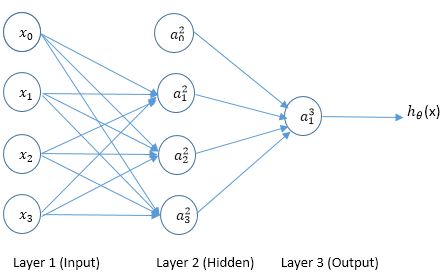
**Neural Network Architecture**

= g( + + + )

= g( + + + )

= g( + + + )

(x) = = g( + + +

****

**Forward Propagation  
  
1. = x (1st layer)**

**2. =**

**3. = g() (add bias term )**

**4. =**

**5. = g()**

**Back-propagation Algorithm**

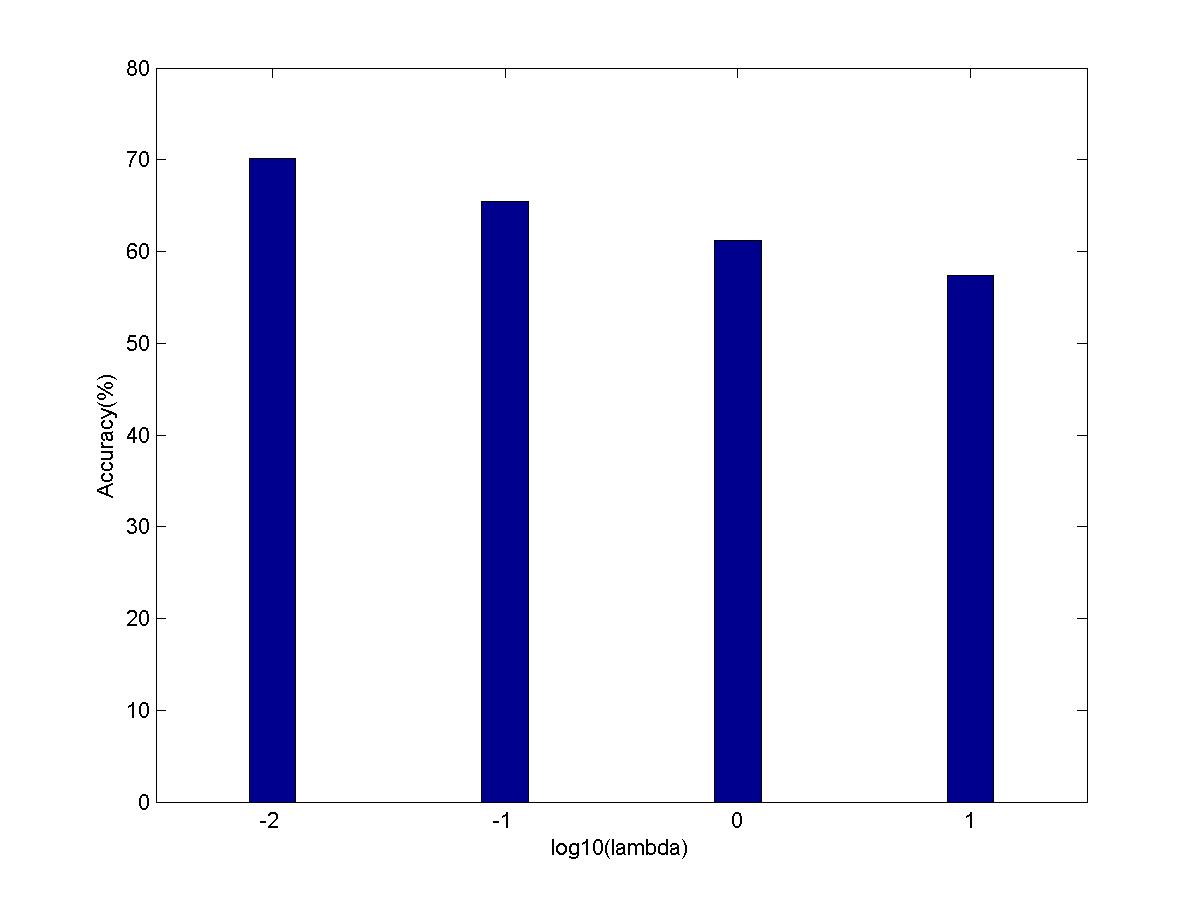
* Perform a forward propagation.
* After calculating all the activation function, we propagate backward calculating the difference between the actual output value and calculated value and then adjust accordingly.
  + For output layer - = -
  + For hidden layers - = .\* g’()
* Use the errors to compute weight adjustments.  
   = +
* Apply the weight adjustments

**RESULTS:**

* First Neural Network with one hidden layer was implemented.
* Different values of lambda was used to check the performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lambda (λ)** | **0.01** | **0.1** | **1** | **10** |
| **Accuracy (%)** | **70.15** | **65.50** | **61.2** | **57.35** |

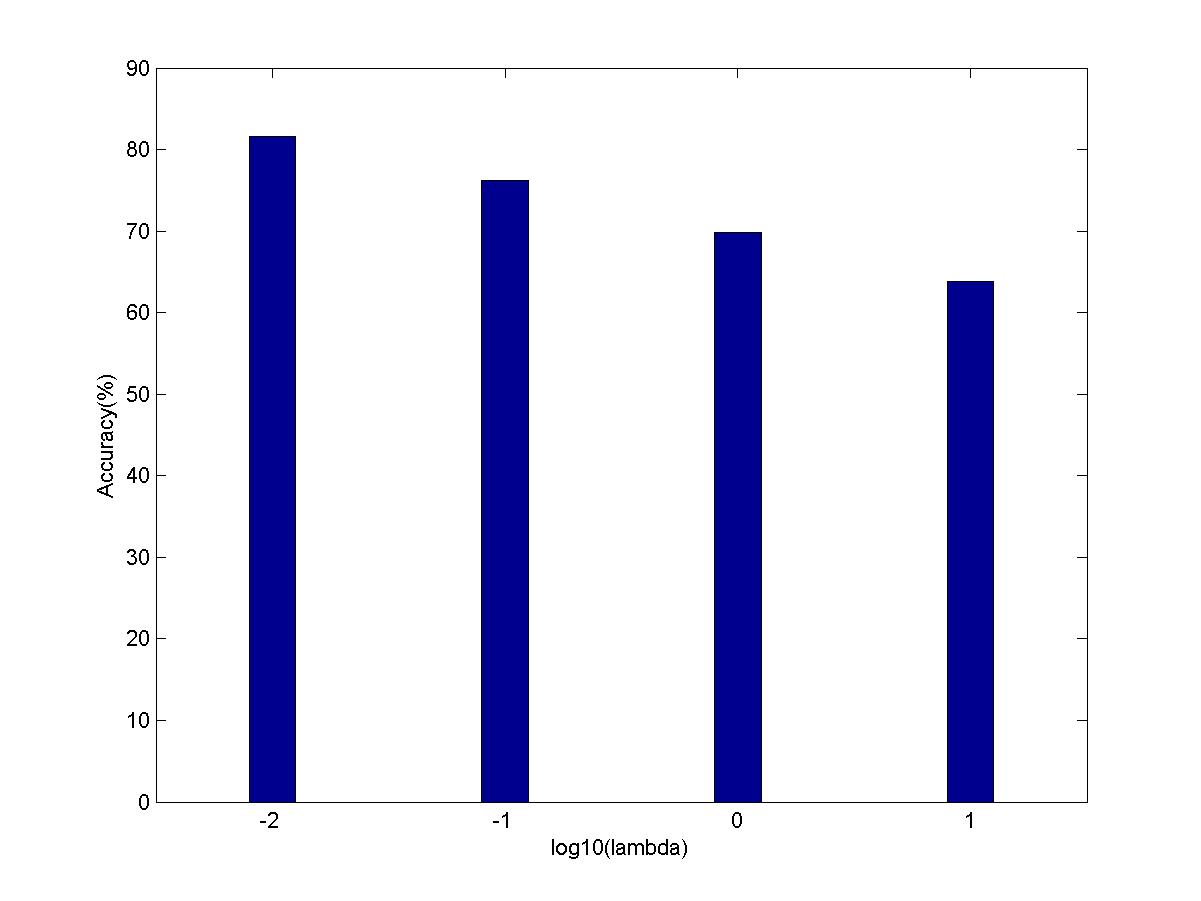
**Bar plot:**

****

* For Neural Network with two hidden layers, results were pretty good.
* Again different values of lambda was used to compare the performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lambda (λ)** | **0.01** | **0.1** | **1** | **10** |
| **Accuracy (%)** | **81.56** | **76.25** | **69.78** | **63.86** |

**Bar Plot:**

****

**Cost Variation with Itearation:**

Iteration 1 | Cost: 6.800585e-001

Iteration 2 | Cost: 3.131947e-001

Iteration 3 | Cost: 2.102862e-001

Iteration 4 | Cost: 1.804524e-001

Iteration 5 | Cost: 1.743623e-001

Iteration 6 | Cost: 1.684271e-001

Iteration 7 | Cost: 1.436359e-001

Iteration 8 | Cost: 1.160207e-001

Iteration 9 | Cost: 9.234027e-002

Iteration 10 | Cost: 7.105888e-002

Iteration 11 | Cost: 6.713526e-002

Iteration 12 | Cost: 6.312395e-002

Iteration 13 | Cost: 5.832920e-002

Iteration 14 | Cost: 5.583898e-002

Iteration 15 | Cost: 5.141869e-002

Iteration 16 | Cost: 4.932675e-002

Iteration 17 | Cost: 4.897259e-002

Iteration 18 | Cost: 4.648696e-002

Iteration 19 | Cost: 4.276392e-002

Iteration 20 | Cost: 4.126713e-002

Iteration 21 | Cost: 3.824391e-002

Iteration 22 | Cost: 3.561850e-002

.

.

.

.

Iteration 63 | Cost: 8.541200e-003

Iteration 64 | Cost: 8.446996e-003

Iteration 65 | Cost: 8.411133e-003

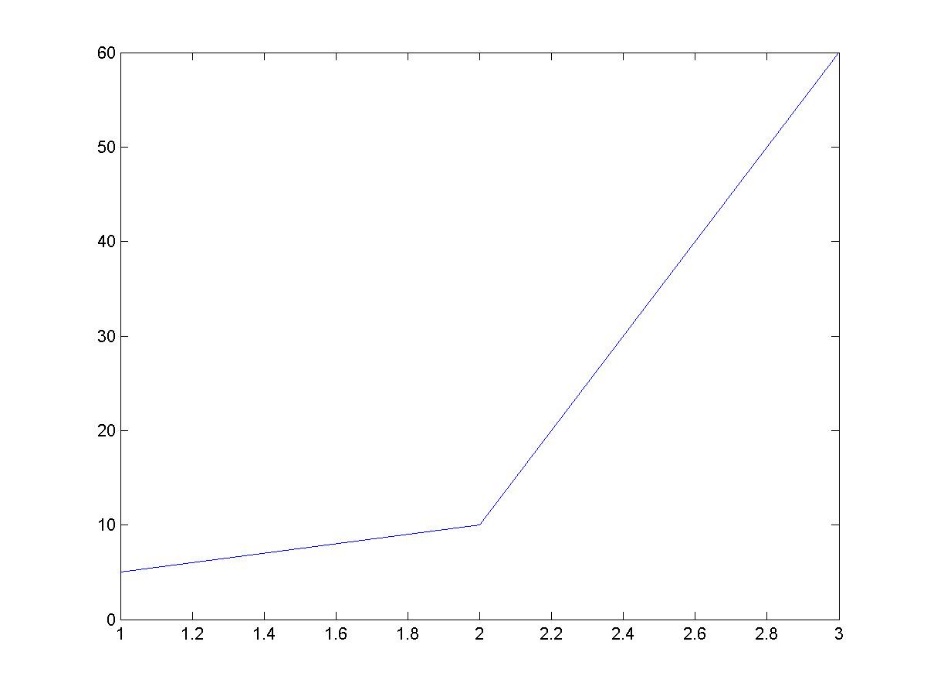
Iteration 66 | Cost: 8.295483e-003

Iteration 67 | Cost: 8.222684e-003

Iteration 68 | Cost: 8.015609e-003

**CONCLUSION**

* Algorithms based on Machine Learning were implemented successfully showing different accuracies.
* NSL-KDD dataset was preprocessed using mean normalization method.
* Linear regression, surprisingly, proved to be very effective in detecting network attacks with a 80% accuracy.
* K-Means Clustering being a semi-supervised approach showed decent results with a 67.5 % accuracy.
* Neural Networks was implemented with one hidden layer one time and two hidden layers other time. With two hidden layers, it proved to be the best among all the approaches above.
* But there is always a trade-off between the accuracy and time an algorithm takes.
* Neural Network took the most time to get trained while K-Means Clustering took lowest amount of time.

****

**FUTURE WORKS:**

* Network Intrusion Detection can be improved using latest machine learning techniques like deep neural network, dbscan etc.
* Here we have not used feature selection to select only relevant features. Using feature selection time performance can be improved as well as accuracy.
* Dimensionality Reduction using principal component analysis can be used to improve the time and visualization.
* The NSL-KDD dataset is very old and there are some bugs which can be tackled using a well derived dataset.

**CRITICIZE**

* The attacks are continuously evolving. So the machine learning model cannot detect these attacks.
* It takes too much time to train the algorithms.
* The models needs to be changed with time.
* Large amount of care is required to manage the codes.

**REFRENCES:**

1. Intrusion Detection System Using Data Mining Technique: Support Vector Machine

Yogita B. Bhavsar , Kalyani C.Waghmare

# Effective approach toward Intrusion Detection System using data mining techniques

G.V. Nadiammai ,  M. Hemalatha

1. **Automatic network intrusion detection: current techniques and open issues**

Carlos A. Catania, Carlos García Garino

1. KDD Cup99 intrusion Detection Dataset. Available from

<[http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html](http://www.sciencedirect.com/science?_ob=RedirectURL&_method=externObjLink&_locator=url&_cdi=280281&_issn=11108665&_origin=article&_zone=art_page&_plusSign=%2B&_targetURL=http%253A%252F%252Fwww.kdd.ics.uci.edu%252Fdatabases%252Fkddcup99%252Fkddcup99.html)>

1. Survey of Network Intrusion Detection Using K-Mean Algorithm

**Poonam Dabas , Rashmi Chaudhary**

1. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection

Robin Sommer , Vern Paxson

1. Andrew Ng, Stanford University Professor Machine Learning Lectures