Chapter 8
Securing the Smart Grid: A Machine Learning Approach

A. B. M. Shawkat Ali, Salahuddin Azad and Tanzim Khorshed

Abstract The demand of electricity is increasing in parallel with the growth of the world population. The existing power grid, which is over 100 years old, is facing many challenges to facilitate the continuous flow of electricity from large power plants to the consumers. To overcome these challenges, the power industry has warmly accepted the new concept *smart grid* which has been initiated by the engineers. This movement will be more beneficial and sustainable to the extent if we can offer a secure smart grid. *Machine learning*, representing a comparatively new era of Information Technology, can make smart grid really secure. This chapter provides an overview of the smart grid and a practical demonstration of maintaining the security of smart grid by incorporating machine learning.

8.1 Introduction

Due to growing concern over massive carbon emission and consequently, rapid climate change, there has been a global movement towards the discovery of clean and renewable energy sources. Traditional energy sources, like fossil fuel, are non-renewable in the sense that they take millions of years to form. Due to huge energy demand worldwide, these traditional energy sources are being exhausted more quickly than the ones being created. The most significant advantage of renewable energy sources are that they are never going to run out as they are refilled naturally. Another advantage is that almost all of the renewable energy sources are relatively clean and cause minor or no pollution to the environment. The most prominent renewable energy source is the *solar* power. The other significant sources are *wind power*, *ocean waves*, *hydropower*, *biomass power* and *geothermal energy*.

A. B. M. S. Ali (✉) · S. Azad · T. Khorshed
Central Queensland University, Rockhampton, QLD 4702, Australia
e-mail: s.ali@cqu.edu.au

A. B. M. S. Ali (ed.), *Smart Grids*, Green Energy and Technology,
DOI: 10.1007/978-1-4471-5210-1_8, © Springer-Verlag London 2013

Due to the technological advancements, increased automation and ever increasing number of consumers, the hunger for energy is likely to aggravate. As the supply from traditional energy sources is running low, there is a tremendous urge globally to become energy efficient. The practice of being energy efficient not only saves energy but also helps to reduce the amount of carbon discharged in the environment during the generation of power. Energy efficiency can be ensured through minimizing of the loss during generation, transmission and distribution phase on the power production side. On the consumer side, energy efficiency can be guaranteed through the design and use of devices that would consume minimum energy during their operations. The consumers also have the responsibility to use power in a sensible way i.e., they should not keep the devices turned on unnecessarily. Alternatively, the device itself should detect when it is not being used and should switch to sleep mode as soon as possible.

The function of an electric power grid is to carry out mass transfer of electric energy at high voltage from power plants, where power is generated, to the substations located near the customer base. In the substations, the electricity is stepped down in voltage and passed through distribution wiring to the consumer site where the power is further stepped down to service voltage. The major problem with tradition electricity distribution systems are that the energy produced cannot be stored and therefore, should be generated as required. When the supply and demand is not in equilibrium, the generation units and transmission network can be shut down causing blackouts. The introduction of alternative power sources, which are intermittent in nature, has made the stable power supply more difficult. Traditional distribution systems are vulnerable to security threats either from energy suppliers or cyber attacks.

The concept of *smart grid* emerged as result of the desperate attempts to make the power grid stable, reliable, efficient and secure. Smart grid attempts predict the usage pattern of the customers and respond intelligently in order to provide reliable, sustainable, and cost-effective services. One significant aspect of the smart grid is that it can schedule and control the load to effectively shift the usage to off-peak hours and reduce the peak demand, which is known as *demand management*. Dynamic pricing is another mechanism of the smart grid to facilitate demand management, which motivates the consumers by increasing the electricity price during the high demand period and reducing the electricity price during the low demand periods. According to the Energy Future Coalition's Smart Grid Working Group, a smart gird should incorporate the following functionalities [1]:

- Would give consumers control over their usage.
- Increase the efficiency and be more economical.
- Should be self-healing and more secure both from physical and cyber attacks.
- Should be able to integrate alternate energy sources like solar cell or wind power.

Smart grid is seen as a way to allow consumers to take part in optimizing the operations by providing them greater information and options. Figure 8.1 describes the operation of self-healing smart grid.

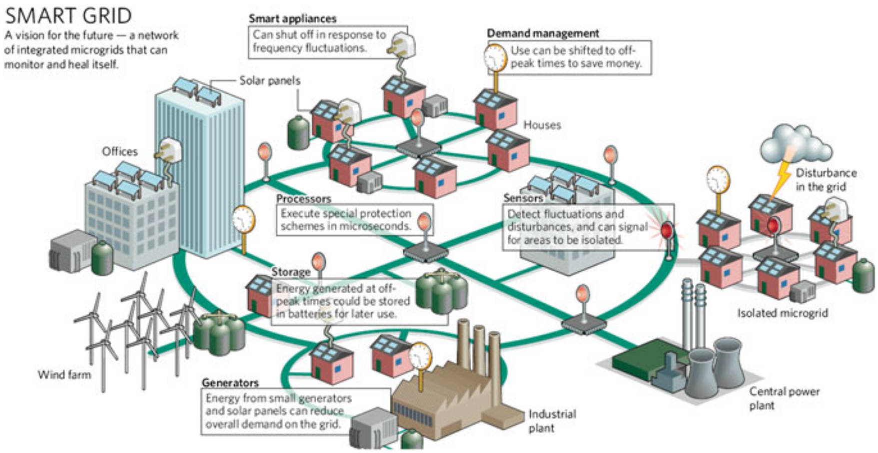

SMART GRID
A vision for the future — a network of integrated microgrids that can monitor and heal itself.
Smart appliances
Can shut off in response to frequency fluctuations.
Demand management
Use can be shifted to off-peak times to save money.
Solar panels
Houses
Offices
Disturbance in the grid
Processors
Execute special protection schemes in microseconds.
Sensors
Detect fluctuations and disturbances, and can signal for areas to be isolated.
Storage
Energy generated at off-peak times could be stored in batteries for later use.
Isolated microgrid
Wind farm
Generators
Energy from small generators and solar panels can reduce overall demand on the grid.
Industrial plant
Central power plant

Fig. 8.1 A smart self-healing grid system [40]

8.2 Smart Power Generation

The main challenge in the current grid is that electricity demand fluctuates throughout the day and also varies from one season to another [2]. Therefore, one of the key factors to maintain reliability and stability in power supply is load balancing. The load balancing task is performed by the transmission operators by matching the power output with the load. In the traditional power grid, combination of three types of power plants—*baseload*, *load follower* and *peaking* are installed to mitigate the balancing task, while maintaining the economy.

Baseload power plants [3] are the power generation facility that produces the minimum amount of energy that is continuously required by a region. Baseload plants operate at the maximum capacity and they typically generate power round the clock and are only shut down for repair and maintenance. Baseload plants are, in general, based on nuclear energy, coal or geothermal energy. *Load follower* power plants [3] generate power only during period of high demand, for example, during daytime or early evening. Gas turbine combined cycle and hydroelectric power plants are typical load followers. *Peaking* power plants [3] generate power when the power demand rises for a short period. Single cycle gas turbines or gas engines are usually deployed as peaking power plants. The baseload plants have the lowest fuel cost followed by the load follower plants and the peaking plants have the highest fuel cost. However, the peaking plants have the shortest start-up time, which means they can start within a very short time during the period of peak demand.

8.2.1 Incorporating Renewable Sources

The supply of power from the renewable sources like solar cell and wind power is not steady and precisely predictable as the time of the day and weather events dictate the output from these sources. It was initially envisaged that statistical smoothing of high peaks of wind or solar power might be possible by interconnecting adjacent areas [4, 5]. But recent research shows that wind and solar power patterns match across a large area and hence, the desired smoothing affect is not achievable [6]. Due to the incorporation of intermittent and variable power generators, the task of load balancing is becoming increasingly complex. For example, the wind speed in South Australia usually drops when the ambient temperature rises during the day [6]. The demand rises, while the electricity production drops during that period, which results in a wind output pattern contrary to the electricity demand. The planned increase in wind power by 2020 in South Australia will severely impact the balance dynamics.

Intermittent power output from renewable sources often causes temporary high-peaks and extended period of zero output. The instantaneous output from the renewable sources is sometimes so huge that it can even drive baseload plants out of production. As the baseload plants require high investment, if the baseload plants remain shut down and operate at reduced load, the capital cost per kWh will rise drastically due to high investment behind the plants. It would be smarter if energy from renewable sources can be stored and used later to flatten the peaks and valleys.

8.2.2 Energy Storage Technologies

Flywheels can be accelerated by electricity and this energy will be preserved as rotation energy. The flywheels can drive generators to cover up sudden shortage of power. The rotational energy will eventually be lost due to friction in the bearing. Putting the rotor inside a vacuum chamber and using magnetic bearing can minimize the energy loss. Electric vehicle batteries may also be used as virtual energy storage which can be charged during off-peak hours and drained off during peak hours. The above mentioned peak-shaving technologies can only provide backup for a short period of time. Storage of energy as thermal energy can be an effective way to balance the demand and supply. Molten salt is heated using solar energy and sent to hot storage tank. This hot salt is later used to produce superheated steam for electricity generation. Surplus renewable energy can also be used to compress air to a high pressure and stored for a long time. The pressurized air can later be used as combustion air in the peaking power plants. Stored natural gas or biogas provides higher energy density and hence, better alternatives to compressed air. Pumped hydro-storage system can pump water from low-altitude reservoir to a high-altitude reservoir. During the period of high demand, this water is passed through the turbines to produce electricity. However, most of the solutions are still more expensive than conventional peaking power plants. Also they are unable to provide backup for extended period of time with zero output from wind or solar power plants. So far, pumped hydro-storage has been proved to be the most economical alternative to the conventional peaking plants. However, pumped hydro-storage system requires huge unoccupied land for the construction of reservoirs.

Recently, hydrogen is emerging as a storage medium. Hydrogen can be produced using electricity or heat and stored in compressed or liquid form. The hydrogen is later converted back to electricity by a combustion engine. An alternative way is to use the hydrogen as a fuel for fuel cell that can convert chemical energy into electricity.

8.2.3 Mitigation of Peak Demand

Since air conditioning is one of the key factors for peak demands, excess energy can be used to produce ice during night time and can be used for air conditioning during daytime. Optimum insulation of buildings is generally the cheapest and most effective solution for reducing the seasonal peaks. Export and import of electricity from neighboring countries can ease the gap between production and demand to a large extent. Cogeneration units use waste heat produced in the power plants for heating purposes either utilizing the heat near the power plants or through district heating. The fuel utilization efficiency of cogeneration units is higher than producing electricity and heat separately [2]. District heating is one of the cheapest methods for reducing carbon footprint.

Demand management rewards the customers for using less energy during the peak time and moves the energy usage to off-peak time so as to shave the peaks and to fill the valleys. Price-based demand management charges the customers at time varying rates that correspond to the value and cost of electricity at different times. Incentive based demand response pay for reducing their loads as requested by the utility provider. Demand management also increases the load during the time of high supply or low demand. The Queensland government is planning to install devices into some household appliances such as air conditioners, pool pumps and hot water systems. The devices would enable the utility companies to cycle the use of these devices during the peak hours [7].

8.2.4 Forecasting Renewable Energy Supply

Wind and solar forecasting technology makes renewable energy supply more reliable and cost-effective as it provides more time to the grid operators to plan in advance for a backup energy supply when the renewable energy source is expected to produce less energy than required. For example, grid operators will not carry much spinning reserve if they know ahead that a sudden drop in renewable is less likely. The objective of wind power prediction is to predict the wind speed and direction, while the objective of solar power prediction is to predict the solar irradiation. Forecasting may be done with a numerical weather prediction model or statistical analysis of local measurements [8]. The first method uses models on meteorological measurements and observations coming from satellites or weather stations. As this is a pure analytical method, no historical data is necessary but the computational complexity is huge. The second method based is on the relationship between the historical data and forecast variables. Since it learns from previous experience, the statistical model can account for the local terrain and other details that can't realistically be represented in the numerical weather prediction model [9]. For the same reason, the statistical model tends to predict typical weather events rather than exceptional events [10].

8.3 Smart Grid Security Issues

Since the smart grid technology is complex which includes generation, transmission, and distribution and hence, there are manifold opportunities for the attackers to disrupt the system. It is a common conception that threats solely comes from hackers or other individuals or groups with malicious intent. In fact, staff and other insiders can also pose a risk as they have authorized access to one or more components of the system. Insiders know about sensitive information of the system such as passwords stored in the database, cryptographic keys and others security mechanisms that could be utilized to organize an attack. However, not all security infringements are malicious; many of them originate from accidental misconfigurations, failure to follow procedures and other oversights.

Smart grid threats can be classified into three broad groups: (1) system level threats that attempt to take down the grid; (2) attempts to steal electrical service; and (3) attempts to compromise the confidentiality of data on the system.

8.3.1 System Level Threats

System level threats attempts to take down part or the entire smart grid. For example, entities or individuals with malicious intent could attempt to change programmed instructions in the meter, change alarm thresholds or issue unauthorized commands to meters or other control device on the grid. This type of actions could result in damage to equipment, premature shutdown of power or processes or even disabling of control equipment. The following are among the system level threats commonly encountered.

8.3.2 Radio Subversion or Takeover

This threat aims to capture one or more radios or the RF communication modules in the meters so they act on behalf of the attackers. The common form of this attack is firmware replacement. Attackers attempt to insert modified firmware into a device or attempt to spread compromised firmware to numerous devices.

8.3.3 Network Barge-in by Strangers

This threat is characterized by attempt of stranger radios to join the RF radio network and/or preventing the communication modules to communicating properly. For example, an attacker may try to use the communication module to piggyback unauthorized traffic through the network or use a stranger radio to intercept or relay traffic. Moreover, an attacker may attempt to modify a radio or communications modules' credentials to assume a different role. Since the interface between the radio and microcontroller is frequently not encrypted, an attacker can participate in the communication as a legitimate device by manipulating the trust relationship.

8.3.4 Denial of Service

This threat results in part or the entire network becoming unresponsive to service request. This attack can take place in form of resource exhaustion or jamming. In routing black holes attack, a node is hacked so that it's advertised as the shortest path to everywhere, resulting in all traffic being diverted to it. RF spectrum jamming attack prevents signal from being received. In jabbering attack, a legitimate node is co-opted to send so much traffic that other nodes can't communicate; Kill packets are protocol packets that cause radios to crash or to become unreachable via the RF field. Stack smashing, a method of subverting or crashing a device's operating system or applications by overloading memory buffers so that data is exposed, lost or corrupted.

The core elements of the network—routers and switches, if not safeguarded properly, can be compromised and make smart grid vulnerable. For example, routers can be shipped with factory default password and remote access such as Telnet or HTTP services turned on. Network administrators knowingly or unknowingly leaving these default settings unchanged can create an entry point into the system for the intruders. In case the devices are compromised, they can be utilized to disrupt grid operations through denial-of-service (DoS) attack. In the worst case, they can be used to take control of more critical control systems.

8.3.5 Malicious Code

Software update feature allows the devices to check for download and install the software packages and patches. If the update is not from trusted vendors, malicious codes such as Trojan horses, or malicious worms can make a way into the system. It is a common practice to write the downloaded software into flash. The code is only executed if it passes the authentication, while the unauthenticated code remains in the flash. This could be executed through code exploit or glitch hardware. The local interfaces are often not secured and create an easy entry point [11].

8.3.6 Glitching

No two transistors in the system are the same due to their locations, tolerances and I/O factors. A glitch forces transistors to operate when they shouldn't [12]. The glitch can be injected through power supply, clock signal, and electromagnetic radiation.

8.4 System Level Theft of Service

Theft of service attack consumes service from the utility provider without paying the revenue to the provider. For example, individual meters or a group of meters can be subverted to misreport to the customer, the amount of service provided or the cost of service provided (changing from a higher-priced one to lower-priced one). The following are theft of service threats commonly encountered [13]:

8.4.1 Cloning

A perpetrator commits this attack by replacing a meter or radio ID with a duplicate one designed to report zero usage. As a result, the utility providers receive no revenue for the service it provides.

8.4.2 Migration

This attack aims at reducing the reported usage and associated bills by swapping a meter (or communication module) from a location reporting high usage to with a meter (or module) from a location reporting low usage.

8.4.3 Meter/Communication Module Interface Intrusion

The communication module connects to meter through a serial port. A perpetrator can disconnect the communication module from the meter or can break into communication module so that it doesn't report any usage information or incorrect usage information.

8.5 Breach of Privacy or Confidentiality

This sort of attack may render personally identifiable information being exposed. Common privacy and confidentiality attacks include:

8.5.1 Meter Compromise

A meter can be broken into to retrieve personal information.

8.5.2 RF Interception

A perpetrator can intercept the packets by passive eavesdropping on the radio network.

8.5.3 Forwarding Point Compromise

If a node on the network is compromised, it can be used to forward traffic to some unauthorized individual or group.

8.5.4 Backbone Network Interception

Packets can be captured as it passes through the backbone IP network.

8.5.5 Bus Sniffing

The interface between the microcontroller and the radio is not often encrypted which can attract bus sniffing. It may be possible for an eavesdropper to capture radio configuration information, cryptographic keys, and network authentication credentials.

8.5.6 Key Compromise

The secret key can be revealed by monitoring the power consumption of the device [11]. Transistors draw more current when switching occurs. Since processing is deterministic and repeatable, each operation in the device leaves an EM signature which can be detected by sampling the current consumed by that device. This is known as *power analysis attack*. *Timing attack* can decode the entire secret key by examining the variation in time a cryptographic operation takes. The secret key can also be inferred from the analysis of electromagnetic radiation emitted from a device. This is known as *electromagnetic attack*.

Using the same symmetric key for encryption across the system can make the system seriously vulnerable. If one of the devices in the system is compromised, the whole system is system is eventually compromised.

8.6 Threat Mitigation

Threat mitigation strategies in smart grid can be broadly classified into four main categories—*physical security*, *privacy and security of data*, *authorization and access control*, and *securing network devices and systems*. Moreover, a number of general strategies are needed to tighten the overall security of the system as discussed in Sect. 8.6.5.

8.6.1 Physical Security

When it comes to physical security, the SCADA network always found to have poorly protected. The primary security measure to secure a smart grid would be to keep the intruders off the premises. This could be achieved through video surveillance, cameras, electronic access control, and emergency response [14].

8.6.2 Privacy and Security of Data

Since different entities are involved in the smart grid, it is crucial to protect the data as it is stored and transmitted. The following measures should be implemented to protect the data in the smart grid [14]:

- Implement firewall functionality to impose access policies between different segments of the smart grid.
- Deployment of VPN architecture that encrypts data to ensure secure and confidential data transmission.
- Leverage encryption and data storage security capabilities to protect critical assets on servers and endpoints.
- Provide granular access to sensitive data at the application level.
- Provide ubiquitous and consistent security measures across wired and wireless security connections.

Moreover, care must be taken in the following cases [14]:

- Avoid using insecure remote management and communication protocols.
- Avoid failed authentication account lockout and logging weaknesses.
- Implement certificate revocation list checking practices.

8.6.3 Authentication and Access Control

The smart grid is accessed by various user groups such as employees, contractors or even customers. Access to these user groups, be it local or remote, must be granular and authorization should be granted to ‘need to know’ assets.

Identity must be verified through strong authentication such as multifactor authentication. Passwords should strong, attempts must be logged and unauthorized attempts also should be logged. Moreover, all access points should be hardened to avoid any loophole in the system. Only ports and services required for normal operation should be enabled.

Communication modules in the meters should use cryptographic keys and digital signatures to confirm that the firmware is from a genuine source and not been tempered. The secret keys must be stored securely and properly protected. One way to achieve this is requiring a password to control the use of the secret key. Blinding the cryptographic operations so that the timing of the operation does not depend on the key can protect against timing attack.

8.6.4 Securing Networked Devices and Systems

Meters should be equipped with temper detection mechanisms. Local temper detection systems are fitted with physical indicators that the meter has been tempered. In remote temper detection mechanism, the meter notifies the head-end about the tempering. System integrity protection system allows the meter to protect its integrity by self-erasure of keys and firm-ware.

To prevent bus sniffing or bus injection, the microcontroller and the radio are placed on the same chip in the meter. However, some devices still have bus sniffing turned on for debugging purposes.

A unique key should be assigned for per device and per use rather than using a single key for all devices across the system. Caution must be taken when cryptographic errors take as it may be an indication of an attack (such as glitching). Care must be taken so that local interfaces are disabled by software.

8.6.5 Maintaining Overall Security

In spite of having discrete functional zones and clear segmentation, it is hard to predict what form of attack would take place. A comprehensive defense strategy is required which would largely cover all sorts of threats and vulnerabilities a smart grid can encounter. An effective, layered defense mechanism should be put in place, which would implement the security principles across the whole infrastructure. The security of the system should be reviewed by the third parties.

System and software development lifecycle implement security all layers (design, coding, testing, deployment and maintenance). Security should be enforced by default instead of making it the responsibility of the end user.

An intrusion detection system (IDS) and/or an intrusion prevention system (IPS) should be implemented to identify external threats trying to penetrate the system and stop any attempts at internal propagation. Host protection mechanism should be deployed to protect critical client systems such as clients, servers and endpoints. Antivirus software and host-based IPSs should be kept up-to-date with latest threat intelligence and signature update.

8.7 An Intelligent DoS Attack Prevention Mechanism

As mentioned earlier in this chapter, the DoS attack is basically an attempt to make a smart grid resource unavailable to its existing users. Although the methods to carry out, motives for, and targets of a DoS attack may differ, it commonly consists of the efforts of an individual or a group of people to temporarily or indefinitely interrupt or suspend the services of smart grid. Distributed denial-of-service

(DDoS) attack is the distributed version of DoS attack, where the attacker uses multiple computers to launch the attacks. Modern DDoS attacks use new techniques to exploit areas where the traditional security solutions are lacking. These attacks can cause severe network downtime to the systems that heavily depend on networks and servers. The number of DDoS attacks remained steady in the recent times but complex multi-vector attacks are becoming more common.

We should not assume that all disruptions to service in the smart grid are the result of a DoS attack. Sometime, the grid may have technical problems with a particular network, or the system administrators may be performing maintenance works. However, the following symptoms *could* indicate a DoS or DDoS attack in the smart grid [15]:

- unusually slow grid network performance
- unavailability for the administrator/customer to get access to a particular or any Website
- dramatic increase in the amount of spam e-mails in the user account

Until today there are no effective ways to prevent being the victim of a DoS or DDoS attack, but there are steps we can take to reduce the likelihood that an attacker will use smart grid network computer to attack user computers. The steps are as follows:

- Install and keep updated anti-virus and operating system software.
- Install a firewall, and configure it to restrict traffic coming into and leaving your computer.
- Following proper security practices.
- Applying email filters may help you manage unwanted traffic.

This chapter demonstrates an intelligent DoS prevention mechanism using machine learning techniques to reduce the likelihood of this sort of attack. Machine learning techniques have been proved to be a very useful tool to prevent a DOS attack [16]. If there is a known type of attack, machine learning can take proactive action to address the issue, and at the same time, notify systems/security administrators as well as the data owner. If an unknown type of attack happens, machine learning will still be able to detect it as an attack from the performance variations from the standard usage, and can notify the designated person with the closest type attack known to its database. That would make the security administrator's job easier and help the administrator fight against unknown types of attacks. In the previous research experiments, the authors successfully identified Cloud insiders activities [17] and DoS/DDoS attacks [18] using machine learning techniques. The previous research found that rule based technique C4.5 is an efficient technique to identify this sort of attacks. The authors validated the performance of different machine learning techniques with the rigorous testing of tenfold cross validation. The experimental outcome demonstrated that C4.5 algorithm not only performs better than other techniques, but also the level of performance is of acceptable standard. The other algorithms tested were Naive

Bayes, Multilayer Perceptron, SVM and PART [17, 18]. This chapter investigates the two groups of machine learning algorithms: *Statistical based* and *Rule based* algorithms to stop DoS attack in the Smart Grid Environment.

8.8 Data Collection

As discussed in the previous section, DoS and DDoS attacks cause significant changes in system performances, and machine learning can easily identify when noteworthy change in system performance occurs. Also [17, 19] suggest that some activities carried out in modern day systems may cause major changes in the performance graph and look very similar to some of the DoS/DDoS attacks in naked eyes. In this situation, machine learning can play a significant role by distinguishing an attack from an activity as it can work with multiple system performance parameters simultaneously that is not possible by human being.

In this section, the pictures of the performance charts that are taken from the modern day server during the cyber attack are presented. The performance plot generated by the data collection spreadsheet during the attack and the similarity between these two indicates are also presented. The primary intention was to train machine with some of the well known DoS/DDoS attacks types and also to train with some normal activities that are carried out in modern network environments every day. So that the machine can distinguish between these two types of activities and can detect unknown type of attacks which are not recorded in the database as an attack or an activity. All these cyber attacks were generated in an experimental environment with some real attack tools. It is to be noted that the performance data of 20 different parameters of System, CPU, Memory and network were collected. In this chapter, only performance charts and plots of those that show significant changes during an attack are included, however, to refine the data using machine learning, all 20 parameters at the same time, irrespective of whether they made any noteworthy dissimilarity or not, are included.

8.8.1 Similarity Between Attacks and Other Activities

Accurate data collection is very important to achieve correct results from machine learning, and also, to make a distinction between an attack and a non-attack activity. To give example of similarity between an attack and a non-attack activity, at first three charts of three different activities are presented. Figure 8.2 presents the disk performance chart for the period of cloning a Virtual Machine (VM) that looks exceptionally similar to disk performance chart at some point in DoS attack using RDoS. Figure 8.3 represents network performance chart while taking the snapshot of a VM over the network and the chart looks incredibly alike to the network performance chart during SYN flood attack. Figure 8.4 shows network

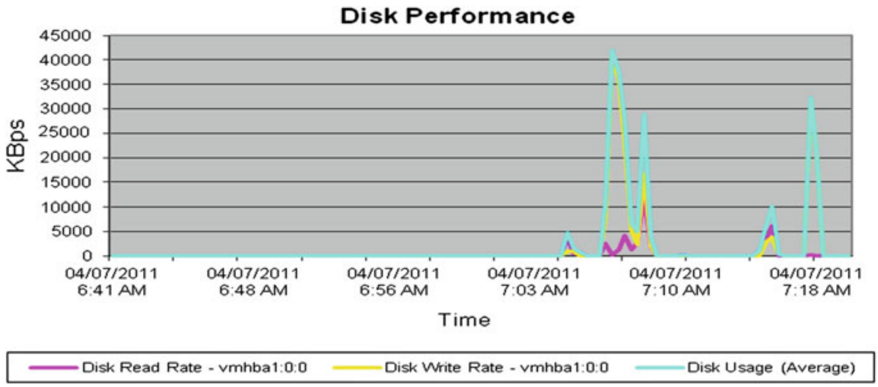


Fig. 8.2 Disk performance chart during the cloning of a VM; it looks incredibly similar to the disk performance chart during DoS attack using RDoS

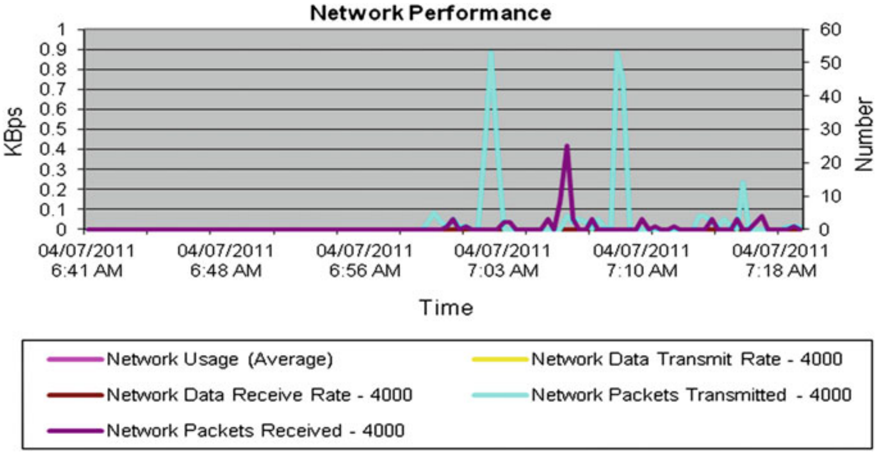


Fig. 8.3 Network performance chart while taking a snapshot of a VM over the network; it looks exceptionally identical to the network performance chart during SYN flood attack

performance chart during installation of new VM and it looks very comparable to the network performance chart during HTTP-DoS attacks.

8.8.2 DoS Using Real-time Disk Operating System

Real-time disk operating system (RDoS) by Rixer [20] is one of the most easily available DDoS attack tool for Web attack. This tool together with a port scanner can be very useful DDoS attacking tools for Web resources. In our experiment the authors only used RDoS and did not use any port scanner as the authors created

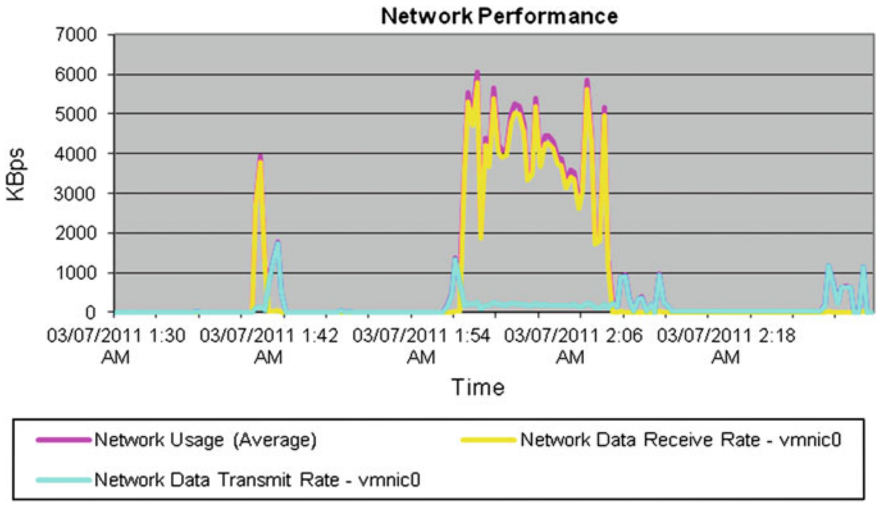


Fig. 8.4 Network performance chart during the installation of a new VM; it looks especially alike to the network performance chart during HTTP-DoS attack

their own Website on a virtual network environment and knew the port number already, which in this case was default HTTP port 80. The HTTP server's Internet Protocol (IP) address is 10.1.1.1 and RDoS tool was executed from other VMs selecting victim's IP address 10.1.1.1 and port 80 from 4:30 to 4:40 a.m.

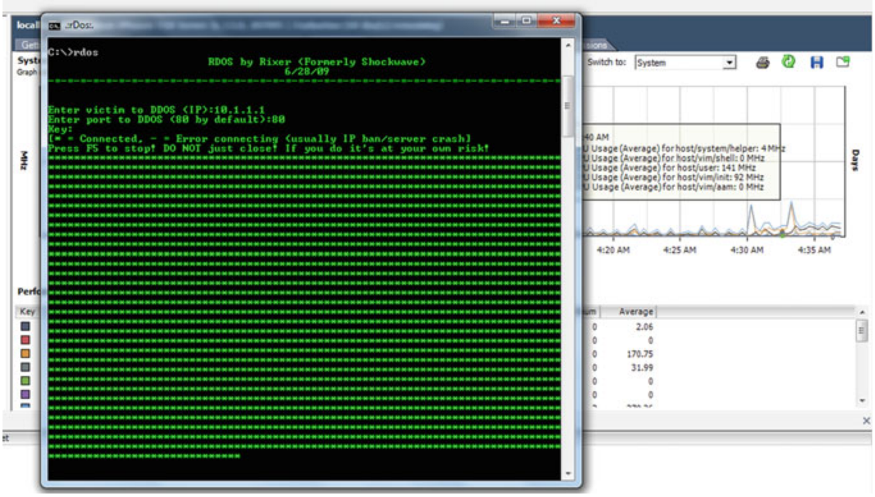
Figure 8.5 shows RDoS by Rixer tool operation in the virtual network environment and also victim's system performance chart screening important changes during the attack. There is a notable change in the performance chart from 4:30 a.m. onwards since the authors ran this tool.

Figures 8.6 and 8.7 represent System and CPU performance charts respectively, during the attack which happened between 4:30 to 4:40 a.m. Important changes in both System and CPU performances are obvious during attack moment.

The following four diagrams presented here are the plots taken from the data collection spreadsheet. These are exactly the same as the hypervisor performance charts during the time of attack, which indicates how precisely the performance data were collected. Figures 8.8 and 8.9 show the performance plots of CPU and Disk respectively, generated by the data collection spreadsheet.

8.8.3 HTTP-DoS Attack Using Low Orbit Ion Cannon

Low Orbit Ion Cannon (LOIC) is an open source network for stress testing and DoS/DDoS attack application [21, 22]. An attacker can flood TCP/UDP packets with the intention of disrupting the service of a particular host. On December 2010, BBC report entitled "Anonymous Wikileaks supporters explain Web attacks"



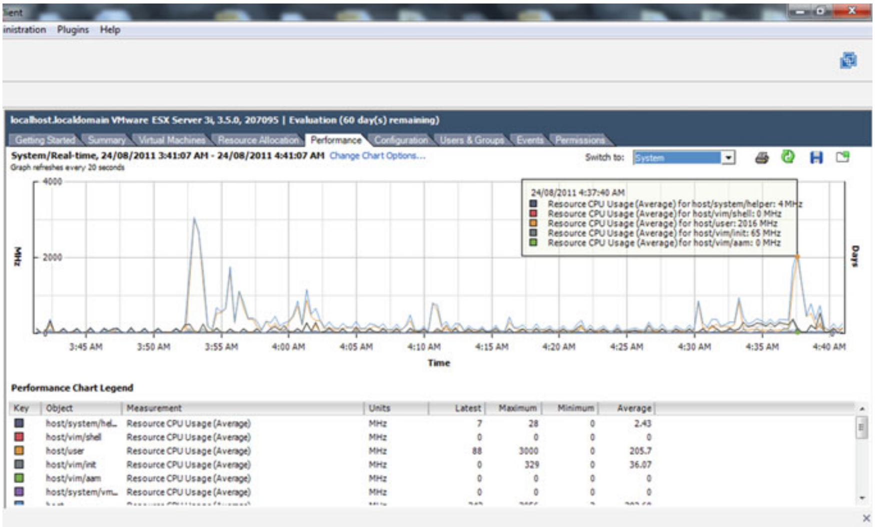
```
C:\>rdos
                    RDOS by Rixer (Formerly Shockwave)
                                 6/28/09
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=


Enter victim to DDOS (IP):10.1.1.1
Enter port to DDOS (80 by default):80
Key:
[* = Connected, - = Error connecting (usually IP ban/server crash)]
Press F5 to stop! DO NOT just close! If you do it's at your own risk!
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
********************************************************************************
******************************
```

40 AM
U Usage (Average) for host/system/helper: 4 MHz
U Usage (Average) for host/vim/shell: 0 MHz
U Usage (Average) for host/user: 141 MHz
U Usage (Average) for host/vim/init: 92 MHz
U Usage (Average) for host/vim/aam: 0 MHz

um	Average
0	2.06
0	0
0	170.75
0	31.99
0	0
0	0

Fig. 8.5 DoS attack using RDoS; the system performance chart screening significant changes during the attack [18]



localhost.localdomain VMware ESX Server 3i, 3.5.0, 207095 | Evaluation (60 day(s) remaining)

System/Real-time, 24/08/2011 3:41:07 AM - 24/08/2011 4:41:07 AM

Graph refreshes every 20 seconds

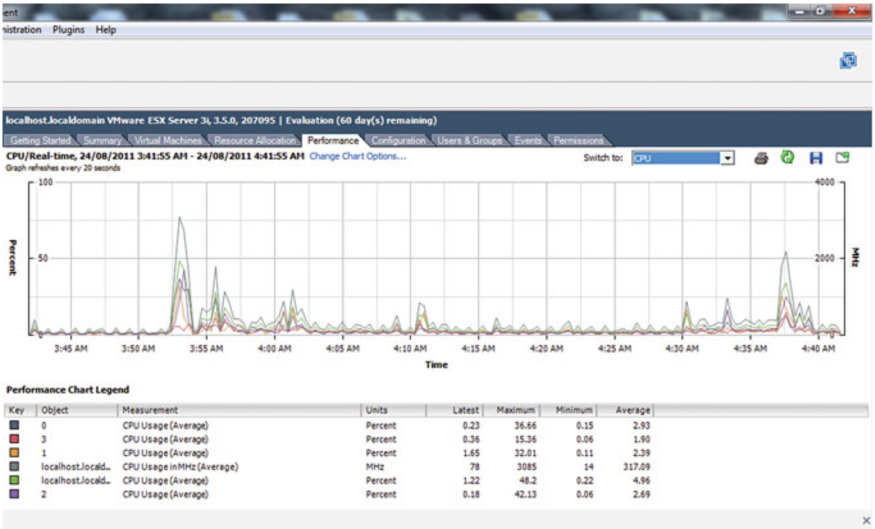
24/08/2011 4:37:40 AM
- Resource CPU Usage (Average) for host/system/helper: 4 MHz
- Resource CPU Usage (Average) for host/vim/shell: 0 MHz
- Resource CPU Usage (Average) for host/user: 2016 MHz
- Resource CPU Usage (Average) for host/vim/init: 65 MHz
- Resource CPU Usage (Average) for host/vim/aam: 0 MHz

Performance Chart Legend

Key	Object	Measurement	Units	Latest	Maximum	Minimum	Average
■	host/system/hel...	Resource CPU Usage (Average)	MHz	7	28	0	2.43
■	host/vim/shell	Resource CPU Usage (Average)	MHz	0	0	0	0
■	host/user	Resource CPU Usage (Average)	MHz	88	3000	0	205.7
■	host/vim/init	Resource CPU Usage (Average)	MHz	0	329	0	36.07
■	host/vim/aam	Resource CPU Usage (Average)	MHz	0	0	0	0
■	host/system/vm...	Resource CPU Usage (Average)	MHz	0	0	0	0

Fig. 8.6 System performance chart during RDoS attack that happened between 4:30 to 4:40 a.m

quoted security experts that well-written firewall rules can filter out most traffic from harmful DDoS attacks by LOIC [23]. However, in the previous research by the authors discovered that these corporate firewalls are not very effective if the attacker resides or shares the same physical hardware from same service provider [24]. For that reason, here, the authors attacked a specific VM from other VMs that is sharing the same physical resources. The HTTP-DoS attack started on victim



localhost.localdomain VMware ESX Server 3i, 3.5.0, 207095 | Evaluation (60 day(s) remaining)

CPU/Real-time, 24/08/2011 3:41:55 AM - 24/08/2011 4:41:55 AM
Graph refreshes every 20 seconds

Performance Chart Legend

Key	Object	Measurement	Units	Latest	Maximum	Minimum	Average
🟦	0	CPU Usage (Average)	Percent	0.23	36.66	0.15	2.93
🟥	3	CPU Usage (Average)	Percent	0.36	15.36	0.06	1.90
🟧	1	CPU Usage (Average)	Percent	1.65	32.01	0.11	2.39
⬛	localhost.locald...	CPU Usage in MHz (Average)	MHz	78	3085	14	317.09
🟩	localhost.locald...	CPU Usage (Average)	Percent	1.22	48.2	0.22	4.96
🟪	2	CPU Usage (Average)	Percent	0.18	42.13	0.06	2.69

Fig. 8.7 CPU performance chart generated in hypervisor during the attack

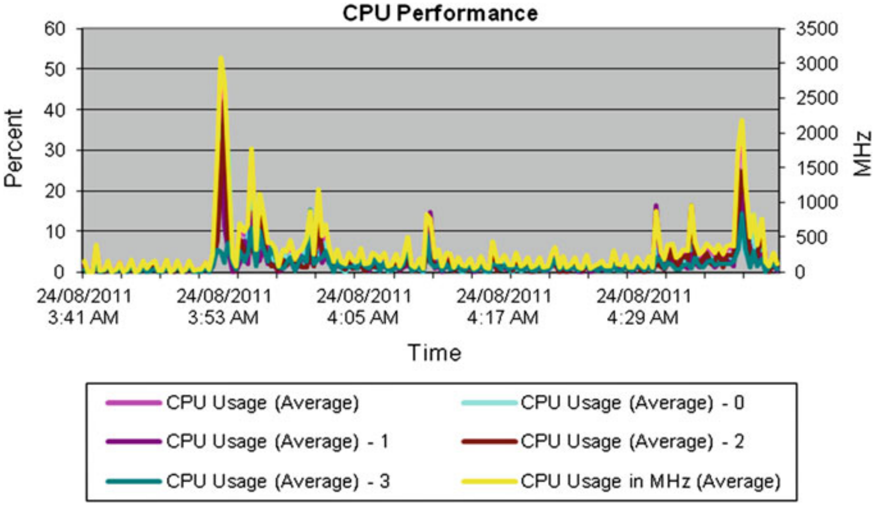


Fig. 8.8 CPU performance plot generated by data collected for the experiment; It has similarity with the one generated automatically by the hypervisor

(IP 10.1.1.1) using LOIC at 6:08 a.m. and ended at 6:15 a.m. Figure 8.10 shows LOIC running from attacker VM with target IP 10.1.1.1.

Figures 8.11, 8.12 and 8.13 present the performance plots of CPU, network and system generated by the data collection spreadsheet that were collected from the

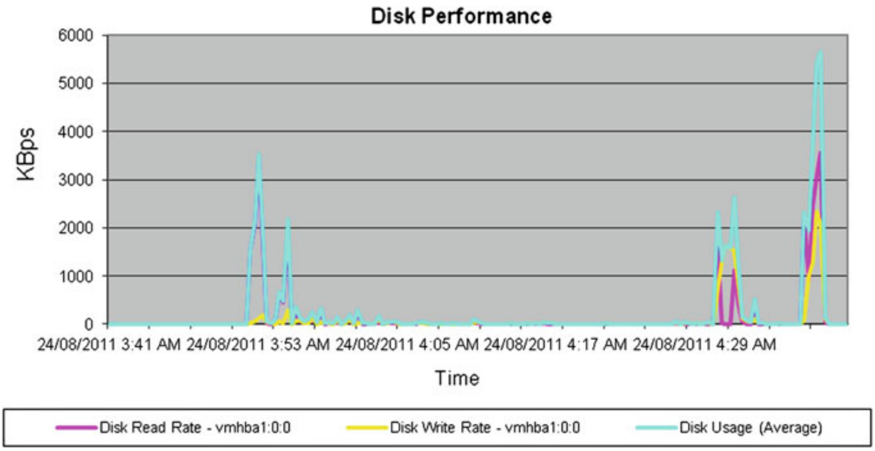
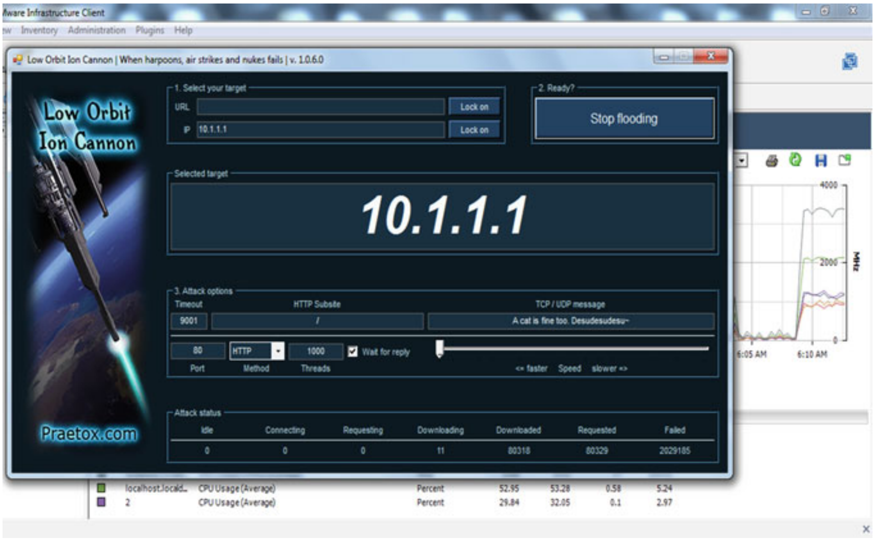


Fig. 8.9 Performance plot disk performance from the data collection spreadsheet


Mware Infrastructure Client
ew Inventory Administration Plugins Help
Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.6.0
Low Orbit Ion Cannon
Praetox.com
Lock on
Lock on
2. Ready?
Stop flooding
4000
2000
0
MHz
6:05 AM
6:10 AM

1. Select your target

URL

IP 10.1.1.1

Selected target

10.1.1.1

3. Attack options

Timeout 9001

HTTP Subsite /

TCP / UDP message A cat is fine too. Desudesudesu~

Port 80

Method HTTP

Threads 1000

☑ Wait for reply

<= faster Speed slower =>

Attack status

Idle	Connecting	Requesting	Downloading	Downloaded	Requested	Failed
0	0	0	11	80318	80329	2029185

🟩	localhost.locald...	CPU Usage (Average)	Percent	52.95	53.28	0.58	5.24
🟪	2	CPU Usage (Average)	Percent	29.84	32.05	0.1	2.97

Fig. 8.10 HTTP-DoS attack using LOIC running, also showing CPU performance chart generated in hypervisor [18]

Virtual machine manager (VMM) during the attack. Sudden increase in the performance was noticed during the time of attack (start 6:08 a.m. and ended at 6:15 a.m.).

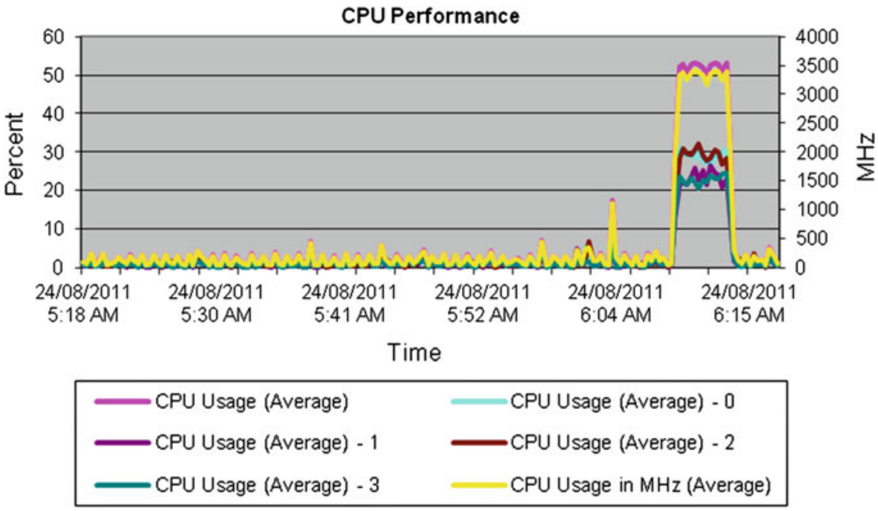


Fig. 8.11 CPU performance plot generated from the data collection during HTTP-DoS attack [24]

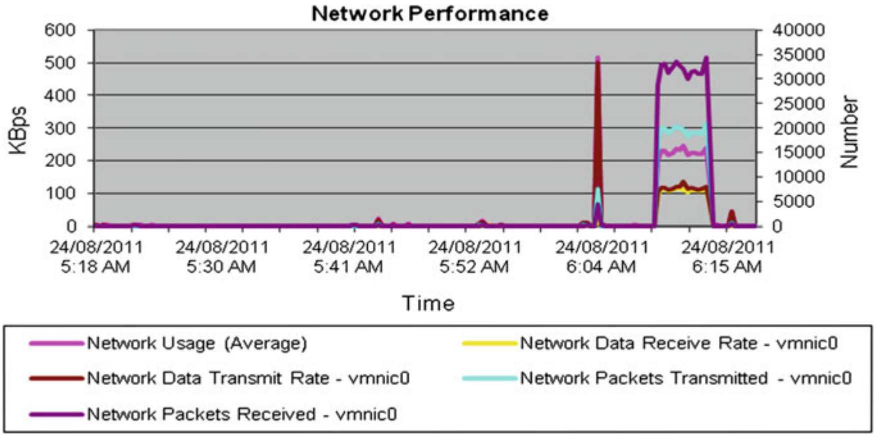


Fig. 8.12 Network performance plot generated from the collected data [24]

8.8.4 Ping Flood Attack

Ping flood is another kind of DoS attack where the attacker crushes the victim with Internet Control message protocol (ICMP) Echo Request (ping) packets. This method could be very successful when sending packets quickly without waiting for a response from the victim. If ICMP service is not disabled by the target host, it will flood the target host with large data segments [25, 26]. However, in our study and from work experience, we found organizations usually disable ICMP requests

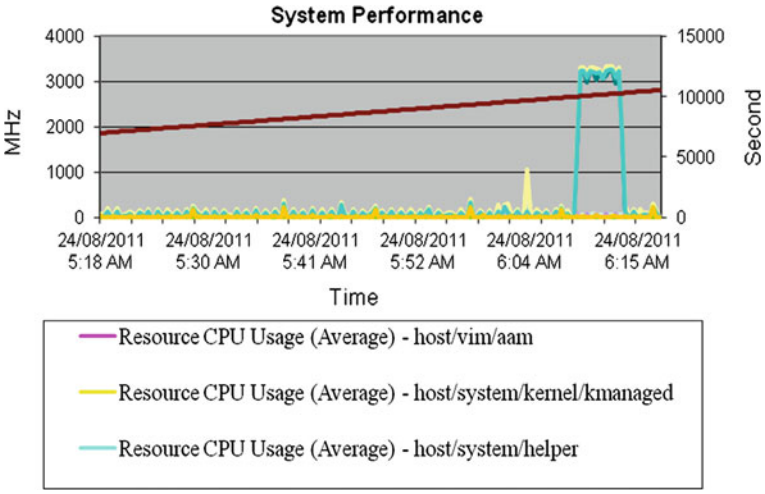


Fig. 8.13 System performance plot generated from the data collected

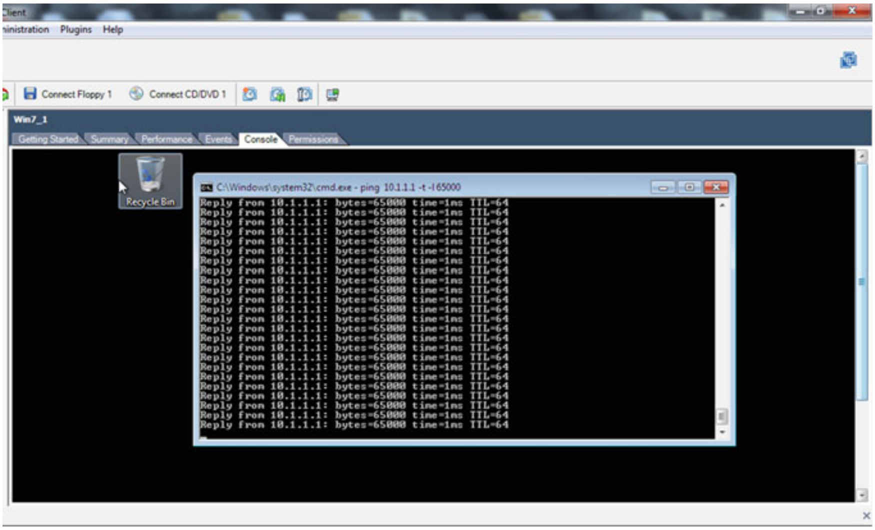
at firewall or in the router so that it can stop ICMP requests from external networks, traditionally they keep ICMP open on hosts in their own internal networks so that they can do network diagnostics. Our concern for modern VMs is that an attacker could be residing on the same physical hardware or somehow can manage to hack into another low secured VM that is residing on same internal virtual network and, carry this kind of attack to a target VM.

A certain kind of ping flood attack in the past was named "ping of death" where an attacker deliberately used to send packets larger than the 65,536 bytes, many computer systems were not able to handle a ping packet larger than this maximum IPv4 packet size [27]. So, in our experiment, we send ICMP packets from each attacker VM slightly lower than that so that the attacker VM itself does not get overwhelmed. We ran "ping 10.1.1.1 –t –l 65000" command from each attacker VM. Here –t was used for repeated sending of echo messages and –l indicates the size of packet to be sent, in this case it was 65,000 bytes from attacker VM1 (we named it win7_1 as shown in Fig. 8.14).

Figure 8.14 shows hypervisor console running ping flood attack from attacker VM to Victim VM. While Fig. 8.15 represents network performance chart from VMM in the instance of attack and Fig. 8.16 is the performance plot of network generated by our data collection spreadsheet.

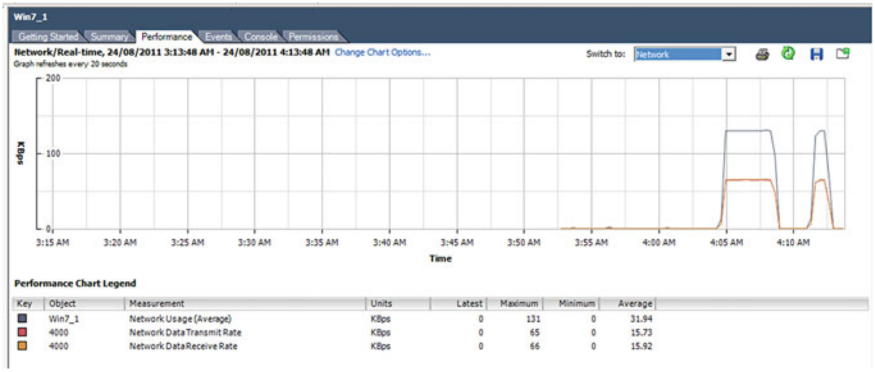
8.8.5 SYN Flood Attack Using Engage Packet Builder

A SYN flood attack is also another kind of DoS attack where a network becomes overwhelmed by a series of SYN requests to a target's system [28]. Engage Packet



```
C:\Windows\system32\cmd.exe - ping 10.1.1.1 -t -l 65000
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
Reply from 10.1.1.1: bytes=65000 time=1ms TTL=64
```

Fig. 8.14 Snapshot of the hypervisor console running ping flood attack from attacker VM to Victim VM



Network/Real-time, 24/08/2011 3:13:48 AM - 24/08/2011 4:13:48 AM Change Chart Options...
Graph refreshes every 20 seconds

Performance Chart Legend

Key	Object	Measurement	Units	Latest	Maximum	Minimum	Average
🟦	Win7_1	Network Usage (Average)	KBps	0	131	0	31.94
🟥	4000	Network Data Transmit Rate	KBps	0	65	0	15.73
🟧	4000	Network Data Receive Rate	KBps	0	66	0	15.92

Fig. 8.15 Network performance chart generated from the hypervisor

Builder [29] is a powerful and scriptable packet builder with capability of packet injection starting from link layer (MAC address spoofing), it can also generate SYN-Floods by building “strange” packets [29]. We used Engage Packet builder to execute SYN flood attack twice, at 5:13 and 5.17 a.m. Figure 8.17 shows Engage Packet Builder running from attacker VM (IP 10.1.1.10) with target IP 10.1.1.1 and Fig. 8.18 shows a performance plot of the network generated from the data which was collected from the VMM during the attack.

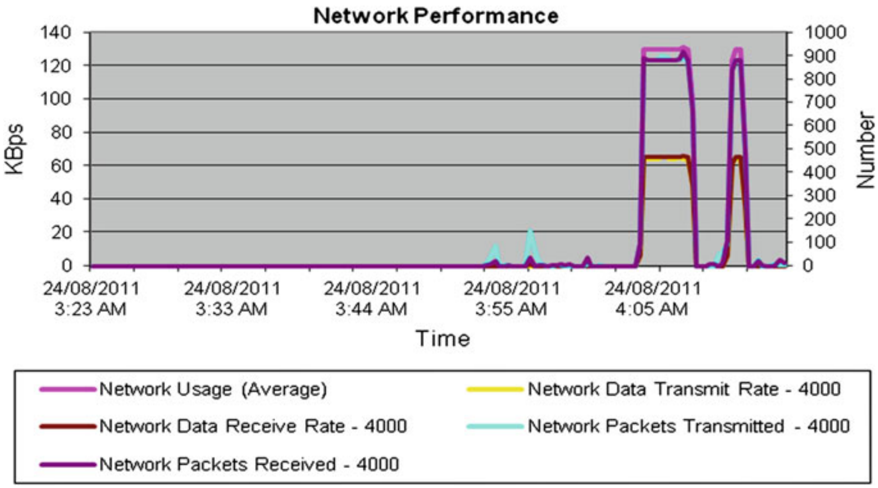
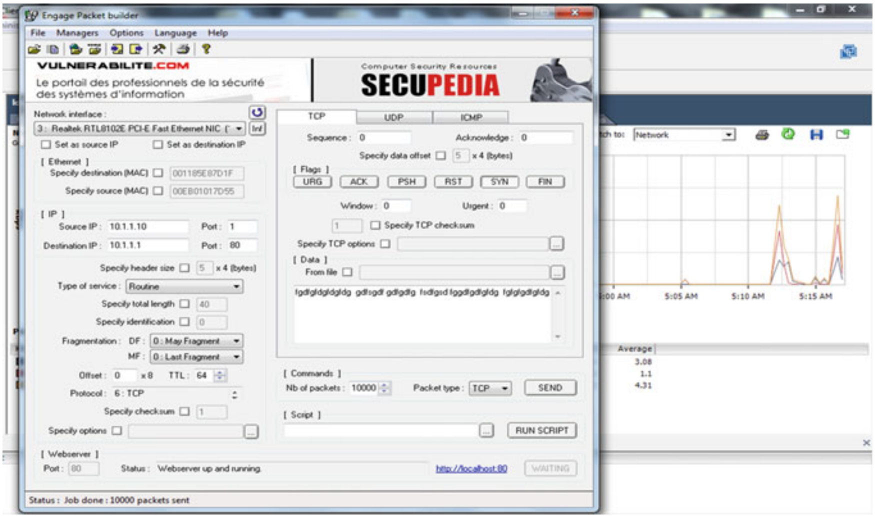


Fig. 8.16 Network performance plot generated from collected data



Network interface : 3 : Realtek RTL8102E PCI-E Fast Ethernet NIC
□ Set as source IP □ Set as destination IP

[Ethernet]
Specify destination (MAC) □ 001185E87D1F
Specify source (MAC) □ 00EB01017D55

[IP]
Source IP : 10.1.1.10 Port : 1
Destination IP : 10.1.1.1 Port : 80
Specify header size □ 5 x 4 (bytes)
Type of service : Routine
Specify total length □ 40
Specify identification □ 0
Fragmentation : DF : 0 : May Fragment
MF : 0 : Last Fragment
Offset : 0 x 8 TTL : 64
Protocol : 6 : TCP
Specify checksum □ 1
Specify options □

Sequence : 0 Acknowledge : 0
Specify data offset □ 5 x 4 (bytes)
[Flags]
Window : 0 Urgent : 0
1 □ Specify TCP checksum
Specify TCP options □
[Data]
From file □
fgdfgldgldgldg gdfsgdf gdfgdfg fsdfgsd fggdfgdfgldg fglgfgdfgldg

[Commands]
Nb of packets : 10000 Packet type : TCP
[Script]

[Webserver]
Port : 80 Status : Webserver up and running. http://localhost:80

Status : Job done : 10000 packets sent

Fig. 8.17 Snapshot of SYN flood attack using engage packet builder and network performance monitoring from the hypervisor during the attack

8.9 Experimental Outcome

The data were collected from a real life environment to measure the strength of the machine learning modeling performance to prevent any DoS attack in the Smart Grid environment. The total numbers of instances in the dataset are 536 and the number of attributes is 21. All the attributes are numeric in the dataset. The authors

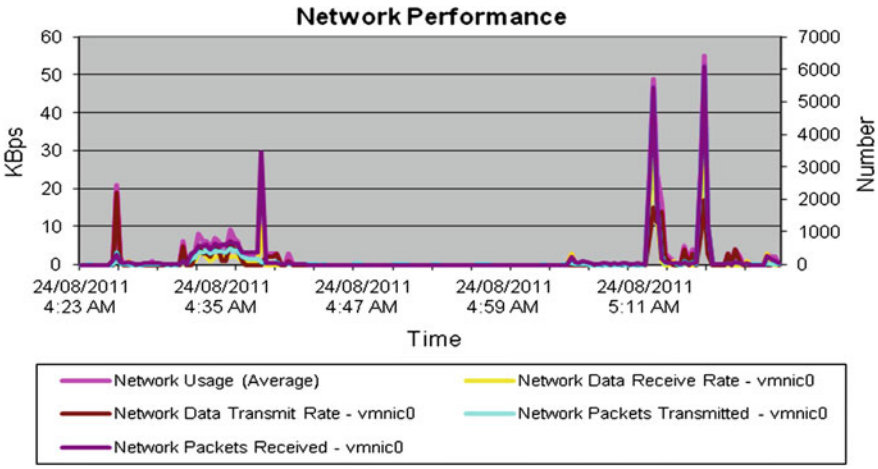


Fig. 8.18 Network performance plot generated from the data collected during the SYN flood attacks

have chosen two sets of machine learning algorithms to prevent DoS attack in the Smart Grid environment: *statistical based learning algorithms* and *rule based learning algorithms*. Naive Bayes [30], Multilayer Perceptron [31], Support Vector Machine [32, 33] are from the statistical based learning algorithms group and PART [34], J48 [35], NBTree [36], and REPTree [37] are from the rule based learning algorithm group selected for the experimental demonstration. All these algorithms are implemented in Java with default parameter settings, which are available in WEKA [38]. WEKA is a machine learning tool developed at the University of Waikato and has become very popular among the academic community working on learning theory.

In the data modeling environment, modeling is a comparative easy task rather than predict an unseen data instance which is called test data. In general, the percentage of model prediction accuracy always carries the strength of the model. Computational intelligence researchers are considering a range of methods to verify the model strength. Among these, cross validation is one of the most widely used methods for the final selection of a model. With cross validation measure, the kappa statistics and model building time for the final model selection to stop attacker in the smart grid network was also chosen. In the following, the tenfold Cross Validation method [38] is explained.

The steps of tenfold cross validation procedure are as follows:

- First, use a random sampling procedure to split the entire training set into 10 sub-samples. Let's call these samples S1, S2, S3 and so on, until we get to S10.
- As a first step, remove one sample set, say 10 (S10), from the training set.
- Train the machine learning algorithm using data from S1 to S9.

- Once the machine has built a model based on data from S1 to S9, it sees how accurately the model predicts the unseen data of S10. Error rates are stored by the system.
- Once the accuracy of predicting the values in S10 is tested, S10 is put back into the training set.
- For the next step, we remove sample set S9 from the training set.
- Re-train the machine learning algorithm, this time using data from S1 to S8 and S10 (i.e., leave out S9).
- Once the machine has built a model based in the training set as described in Step 7, it evaluates how accurately it can predict values in the new test set (i.e., S9). Error rates are stored by the system.
- Put S9 back into the training set.
- Now, remove S8 from the training set, and repeat the testing procedure.

At the end of the sequence, the 10 results from the folds can be averaged to produce a single estimation of the model's predictive performance. The main advantage of the tenfold cross validation method is that all observations are used for both training and validation, and each observation is used for validation exactly once. This leads to a more accurate measure of how efficiently the algorithm has "learned" a concept, based on the training set data. Thus, a final model is setup to predict the upcoming new sample.

Experimentally, the accuracy as the overall number of correct classifications averaged across all tenfolds is estimated. Let D_i be the test set that includes sample $v_i = \langle \mathbf{x}_i, y_i \rangle$ and the cross validation accuracy estimation is defined as:

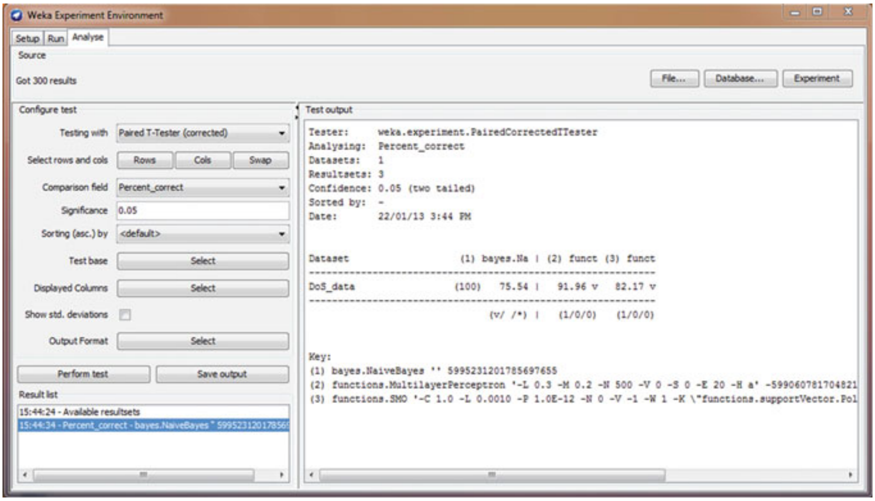
$$\mathrm{Acc}_{\mathrm{cv}} = \frac{1}{nf}\sum_{\mathrm{i}=1}^{f} \delta\left(\Im\left(D_{(i)}, \mathbf{x}_i\right), y_i\right),$$

where f is the number of folds and n is the number of labeled instances in the fold. Kappa is a chance-corrected measure of agreement between two raters, each of whom independently classifies each of a sample of subjects into one of a set of mutually exclusive and exhaustive categories [39]. It is computed as

$$K = \frac{p_o - p_e}{1 - p_e},$$

where $p_o = \sum_{i=1}^{k} p_{ii}$, $p_e = \sum_{i=1}^{k} p_{i.}p_{.i}$, and $p =$ the proportion of ratings by two raters on a scale having k categories.

Both statistical and rule based learning algorithms' performances have been summarized in Figs. 8.19, 8.20, 8.21, 8.22, 8.23 and 8.24. Figure 8.19 shows Multilayer Perceptron algorithm as is the best choice for preventing the DoS attach in the Smart Grid. In the Kappa statistics measure of Fig. 8.20, Multilayer Perceptron is again shown as the most superior. However, in terms of computational complexity measure, Multilayer Perceptron was the last choice as shown in Fig. 8.21. Naïve Bayes is a comparatively faster algorithm among the statistical learning algorithms. Figure 8.22 shows that PART algorithm is the best choice



Got 300 results

Testing with: Paired T-Tester (corrected)

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Show std. deviations: □

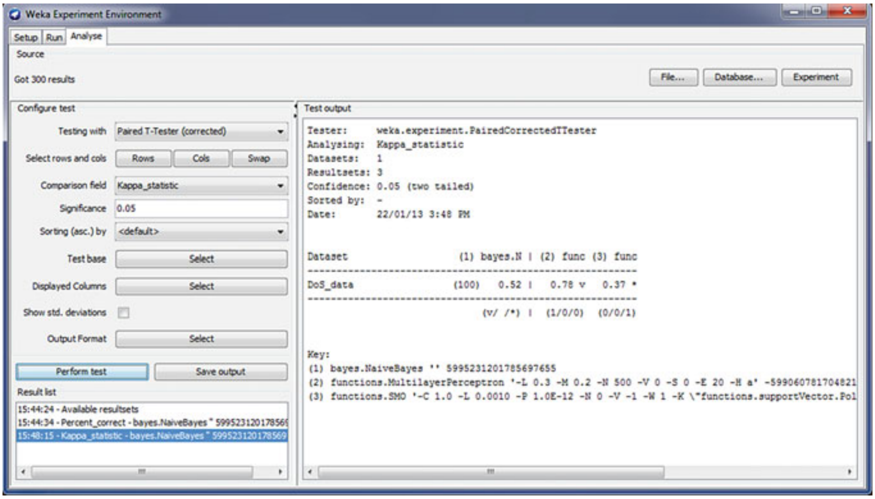
Result list:
15:44:24 - Available resultsets
15:44:34 - Percent_correct - bayes.NaiveBayes '' 599523120178569

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       22/01/13 3:44 PM


Dataset                   (1) bayes.Na | (2) funct (3) funct
------------------------------------------------------------
DoS_data                (100)   75.54 |   91.96 v   82.17 v
------------------------------------------------------------
                              (v/ /*) |   (1/0/0)   (1/0/0)


Key:
(1) bayes.NaiveBayes '' 5995231201785697655
(2) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -599060781704821
(3) functions.SMO '-C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.Pol
```

Fig. 8.19 DoS attack classification accuracies of statistical based learning algorithms



Got 300 results

Testing with: Paired T-Tester (corrected)

Comparison field: Kappa_statistic

Significance: 0.05

Sorting (asc.) by: <default>

Show std. deviations: □

Result list:
15:44:24 - Available resultsets
15:44:34 - Percent_correct - bayes.NaiveBayes '' 599523120178569
15:48:15 - Kappa_statistic - bayes.NaiveBayes '' 599523120178569

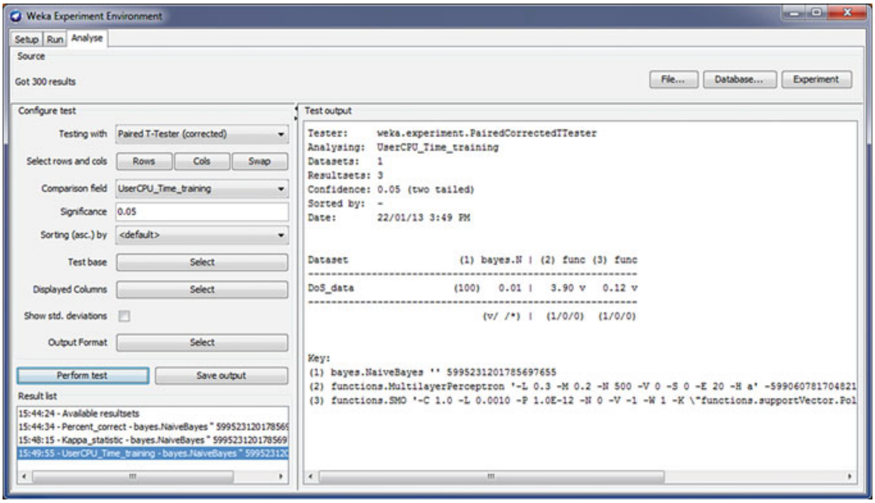
```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Kappa_statistic
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       22/01/13 3:48 PM


Dataset                   (1) bayes.N | (2) func (3) func
---------------------------------------------------------
DoS_data                (100)   0.52 |   0.78 v   0.37 *
---------------------------------------------------------
                             (v/ /*) |  (1/0/0)  (0/0/1)


Key:
(1) bayes.NaiveBayes '' 5995231201785697655
(2) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -599060781704821
(3) functions.SMO '-C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.Pol
```

Fig. 8.20 DoS attack classification Kappa statistic of statistical based learning algorithms

among the rule based learning algorithms. J48 was the second choice within the rule based learning algorithms. In the Kappa statistics measure of Fig. 8.23, PART was again shown as the most superior. However, in terms of computational complexity measure, REPTree is the best choice as shown in Fig. 8.21. J48 was the second best choice comparing among these four rule based learning algorithms.



Got 300 results

Testing with: Paired T-Tester (corrected)
Comparison field: UserCPU_Time_training
Significance: 0.05
Sorting (asc.) by: <default>
Show std. deviations: □

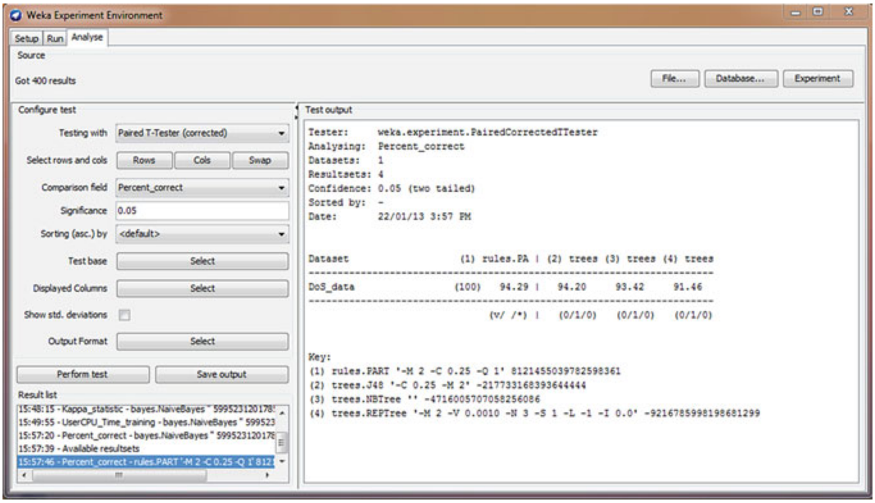
```
15:44:24 - Available resultsets
15:44:34 - Percent_correct - bayes.NaiveBayes '' 599523120178569
15:48:15 - Kappa_statistic - bayes.NaiveBayes '' 599523120178569
15:49:55 - UserCPU_Time_training - bayes.NaiveBayes '' 59952312C
```

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  UserCPU_Time_training
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       22/01/13 3:49 PM


Dataset                   (1) bayes.N | (2) func (3) func
---------------------------------------------------------
DoS_data                 (100)    0.01 |   3.90 v   0.12 v
---------------------------------------------------------
                               (v/ /*) |  (1/0/0)  (1/0/0)


Key:
(1) bayes.NaiveBayes '' 5995231201785697655
(2) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -599060781704821
(3) functions.SMO '-C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.Pol
```

Fig. 8.21 DoS attack classification model building time of statistical based learning algorithms



Got 400 results

Testing with: Paired T-Tester (corrected)
Comparison field: Percent_correct
Significance: 0.05
Sorting (asc.) by: <default>
Show std. deviations: □

```
15:48:15 - Kappa_statistic - bayes.NaiveBayes '' 599523120178!
15:49:55 - UserCPU_Time_training - bayes.NaiveBayes '' 599523
15:57:20 - Percent_correct - bayes.NaiveBayes '' 59952312017E
15:57:39 - Available resultsets
15:57:46 - Percent_correct - rules.PART '-M 2 -C 0.25 -Q 1' 812!
```

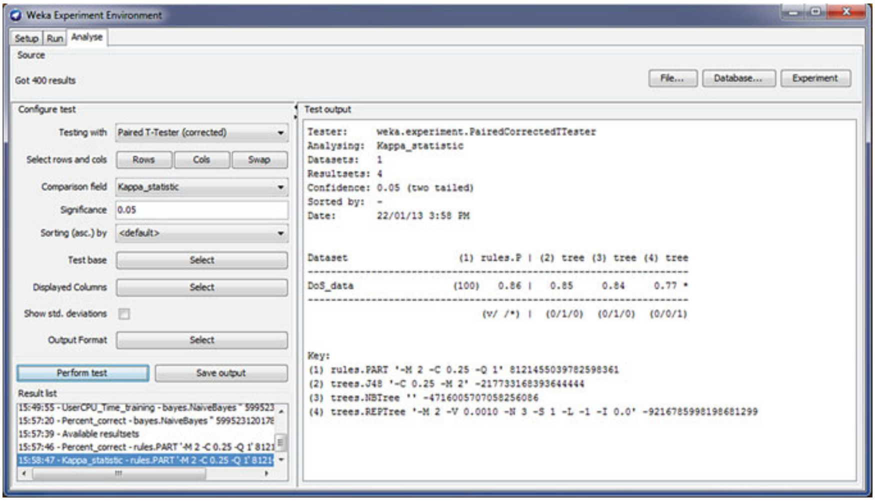
```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 4
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       22/01/13 3:57 PM


Dataset                   (1) rules.PA | (2) trees (3) trees (4) trees
----------------------------------------------------------------------
DoS_data                 (100)   94.29 |   94.20     93.42     91.46
----------------------------------------------------------------------
                               (v/ /*) |   (0/1/0)   (0/1/0)   (0/1/0)


Key:
(1) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) trees.NBTree '' -4716005707058256086
(4) trees.REPTree '-M 2 -V 0.0010 -N 3 -S 1 -L -1 -I 0.0' -9216785998198681299
```

Fig. 8.22 DoS attack classification accuracies of rule based learning algorithms

From the above discussions, it can be concluded that PART algorithm is the best choice to prevent any type of DoS attack in the Smart Grid network. This was not only the fastest algorithm but also in terms of attack classification measures, it appears to be the most accurate. Basically, it uses separate-and-conquer method to build a model. It generates a partial C4.5 decision tree (which is implemented as J48 in Weka) in each iteration and makes the "best" leaf into a rule.



Got 400 results

Testing with: Paired T-Tester (corrected)
Comparison field: Kappa_statistic
Significance: 0.05
Sorting (asc.) by: <default>
Show std. deviations: □

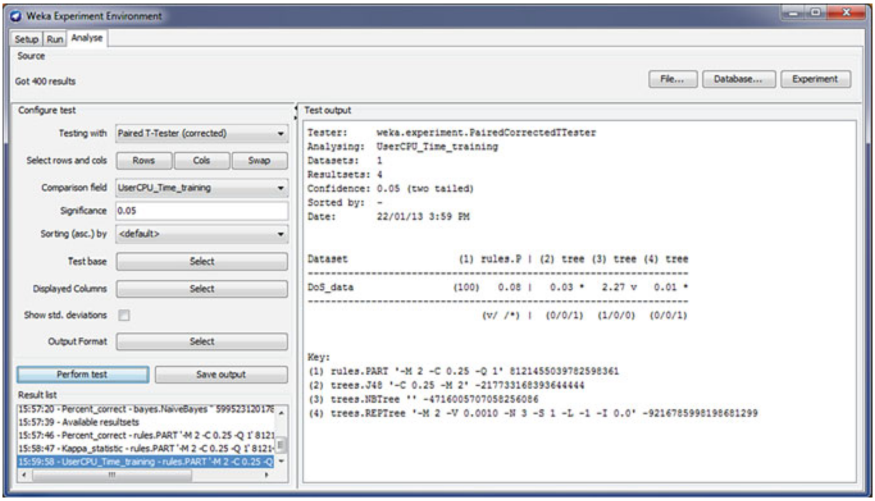
```
15:49:55 - UserCPU_Time_training - bayes.NaiveBayes '' 599523
15:57:20 - Percent_correct - bayes.NaiveBayes '' 59952312017
15:57:39 - Available resultsets
15:57:46 - Percent_correct - rules.PART '-M 2 -C 0.25 -Q 1' 8121
15:58:47 - Kappa_statistic - rules.PART '-M 2 -C 0.25 -Q 1' 8121
```

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Kappa_statistic
Datasets:   1
Resultsets: 4
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       22/01/13 3:58 PM


Dataset                  (1) rules.P | (2) tree (3) tree (4) tree
-----------------------------------------------------------------
DoS_data                (100)   0.86 |   0.85     0.84     0.77 *
-----------------------------------------------------------------
                             (v/ /*) |  (0/1/0)  (0/1/0)  (0/0/1)


Key:
(1) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) trees.NBTree '' -4716005707058256086
(4) trees.REPTree '-M 2 -V 0.0010 -N 3 -S 1 -L -1 -I 0.0' -9216785998198681299
```

Fig. 8.23 DoS attack classification Kappa statistic of rule based learning algorithms



Got 400 results

Testing with: Paired T-Tester (corrected)
Comparison field: UserCPU_Time_training
Significance: 0.05
Sorting (asc.) by <default>
Show std. deviations: □

```
15:57:20 - Percent_correct - bayes.NaiveBayes '' 59952312017
15:57:39 - Available resultsets
15:57:46 - Percent_correct - rules.PART '-M 2 -C 0.25 -Q 1' 8121
15:58:47 - Kappa_statistic - rules.PART '-M 2 -C 0.25 -Q 1' 8121
15:59:58 - UserCPU_Time_training - rules.PART '-M 2 -C 0.25 -Q
```

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  UserCPU_Time_training
Datasets:   1
Resultsets: 4
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       22/01/13 3:59 PM


Dataset                  (1) rules.P | (2) tree (3) tree (4) tree
-----------------------------------------------------------------
DoS_data                (100)   0.08 |   0.03 *   2.27 v   0.01 *
-----------------------------------------------------------------
                             (v/ /*) |  (0/0/1)  (1/0/0)  (0/0/1)


Key:
(1) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) trees.NBTree '' -4716005707058256086
(4) trees.REPTree '-M 2 -V 0.0010 -N 3 -S 1 -L -1 -I 0.0' -9216785998198681299
```

Fig. 8.24 DoS attack classification model building time of rule based learning algorithms

8.10 Discussions

Use of machine learning algorithms to foil DoS attack by means of a simple data-based approach for the Smart Grid network is comparatively a novel technique. In this chapter, the authors put forward a rule based learning approach, PART, for DoS attack classification problem. Unlike other rule based learners, it is easier to visualize the rules during the PART algorithm learning process. PART performance on the real life data were tested for preventing the Dos attack in the Smart Grid network and its performance was compared with that of other rule based learning algorithms. In addition, performance of PART was also tested against a group of statistical based learning algorithms. It is evident that PART outperforms other approaches on both prediction accuracy and the Kappa statistics measure. In terms of computational complexity measure, PART is not the best choice for preventing DoS attack, although its computational complexity is comparable to that of other rule-based algorithms. Further testing on other types of attack in the Smart Grid network is in progress in order to study the robustness of PART algorithm. The classification performance could be improved by adopting the best suited feature selection algorithm inside the PART implementation.

References

1. S. G. W. Group (2003) Challenge and opportunity: charting a new energy future: appendix A working group reports. Energy Future Coalition, Washington DC
2. Paro A, Fadigas E (2011) A methodology for biomass cogeneration plants overall energy efficiency calculation and measurement—a basis for generators real time efficiency data disclosure. In: Proceedings of power systems conference and exposition (PSCE), pp 1–7
3. Denholm P et al (2010) The role of energy storage with renewable electricity generation. National Renewable Energy Laboratory, Colorado
4. DeCarolis JF, Keith DW (2006) The economics of large-scale wind power in a carbon constrained world. Energy Policy 34:395–410
5. Archer CL, Jacobson MZ (2007) Supplying base load power and reducing transmission requirements by interconnecting wind farms. J Appl Meteorol Climatol 46:1701–1717
6. Freris L, Infield D (2008) Renewable energy in power systems. Wiley, New York
7. EDAI Department of Employment (2011) Queensland energy management plan, department of employment, economic development and innovation, Queensland government. http://rti.cabinet.qld.gov.au/documents/2011/may/qld%20energy%20management%20plan/Attachments/Qld%20Energy%20Mgt%20Plan.pdf. Accessed 13 Oct 2011
8. Delucchi M. A, Jacobson M. Z (2011) Providing all global energy with wind, water, and solar power, Part II: Reliability, system and transmission costs, and policies. Energy Policy 39:1170–1190
9. Grant W et al (2009) Change in the Air. Power Energ Mag IEEE 7:47–58
10. Zhong J et al (2010) Wind power forecasting and integration to power grids. In: Proceedings of 2010 international conference on green circuits and systems (ICGCS), pp 555–560
11. Sense of Security Pty Ltd (2011) Securing the Smart Grid. In: Proceedings of smart electricity world conference
12. Jamieson A (2011) Close the door! securing embedded systems. In: Proceedings of AusCERT information security conference
13. Smart Grid Security Myths vs. Reality (2012) White paper, SilverSpring Networks
14. Smart grid security critical success factors. http://www.cio.com.au/article/363005/smart_grid_security_critical_success_factors/R,Cited. 11 Feb 2013
15. McDowell M (2009) Understanding denial-of-service attacks. http://www.us-cert.gov/cas/tips/ST04-015.html. Accessed 10 Jan 2013
16. Ali ABMS (2012) What's at risk as we get smarter?. IEEE Smart Grid Newsletter, USA
17. Khorshed M T et al (2011) Monitoring insiders activities in cloud computing using rule based learning. In: Proceedings of IEEE trustcom-11, Changsha, China

18. Khorshed MT et al (2012) Classifying different DoS attacks in cloud computing using rule based learning, security and communication networks. Wiley, New York
19. Khorshed M T et al (2011) Trust issues that create threats for cyber attacks in cloud computing. In: Proceedings of IEEE ICPADS, Tainan, Taiwan
20. ecuritytube.net. (2012) Ddos attack with Rdos and T3c3i3. http://www.securitytube.net/video/471922. Accessed 12 Aug 2012
21. Batishchev AM (2012) LOIC. http://sourceforge.net/projects/loic/. Accessed 22 Aug 2012
22. G. Inc. (2012) NewEraCracker LOIC. https://github.com/NewEraCracker/LOIC/22. Accessed Aug 2012
23. BBC (2010) Anonymous wikileaks supporters explain web attacks. http://www.bbc.co.uk/news/technology-11971259. Accessed 23 Aug 2012
24. Khorshed MT et al (2012) A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing, Future Generation Comput Syst Elsevier 28(6):833-851
25. Nanda R (2008) DDoS attack/PING flooding: explanation and solution. http://ramannanda.blogspot.com.au/2009/05/ddos-attackping-flooding-explanation.html. Accessed 23 Aug 2012
26. Grid G (2010) Tutorial: how to DoS attack (ping flooding). http://ghostgrid.blog.com/2010/12/16/ping-flooding/. Accessed 23 Aug 2012
27. Rouse M (2006) Ping of death. http://searchsecurity.techtarget.com/definition/ping-of-death. Accessed 23 Aug 2012
28. Kumar A et al (2012) Performance evaluation of centralized multicasting network over ICMP ping flood for DDoS, Performance Evaluation. Int J Comput Appl 37(10):1-6
29. Wilmes G, Kistler U (2007) Engage packet builder—scriptable libnet-based packet builder. http://www.engagesecurity.com/products/engagepacketbuilder/. Accessed 24 Aug 2012
30. John GH, Langley P (1995) Estimating continuous distributions in bayesian classifiers. In: Proceedings of 11th conference on uncertainty in artificial intelligence, San Mateo, pp 338–345
31. Michie D et al (1994) Machine learning, neural and statistical classification. Ellis Horwood series in artificial intelligence, Chichester, New York
32. Platt JC (1999) Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods—Support Vector Learning, pp 185–208
33. Keerthi SS et al (2001) Improvements to platt's SMO algorithm for SVM classifier design. Neural Comput 13:637–649
34. Frank E, Witten IH (1998) Generating accurate rule sets without global optimization. In: Proceedings of 15th international conference on machine learning, pp 144–151
35. Quinlan JR (1993) C4. 5: programs for machine learning. Morgan Kaufmann, San Mateo
36. Kohavi R (1996) Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Proceedings of the 2nd international conference on knowledge discovery and data mining
37. Witten IH et al (2011) Data mining: practical machine learning tools and techniques: practical machine learning tools and techniques. Morgan Kaufmann, USA
38. Contextuall (2012) What is 10-Fold cross validation? https://contextuall.com/what-is-10-fold-cross-validation/. Accessed 12 Jan 2013
39. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Measur 20:37–46
40. Marris E (2008) Upgrading the grid. Nature 454:570–573