International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Intrusion Detection in Dynamic Distributed Network Using PSO and SVM Machine Learning Algorithms

Amol S. Jadhav[1], Dhanashree Kulkarni[2]

[1, 2]Department of Computer Engineering, Dr. D. Y. Patil College of Engineering, Ambi, Pune, India

Abstract: ***The number of computers connected to a network and the Internet is increasing with every day. This combined with the increase in networking speed has made intrusion detection a challenging process. System administrators today have to deal with larger number of systems connected to the networks that provide a variety of services. The challenge here is not only to be able to actively monitor all the systems but also to be able to react quickly to different events. Overall intrusion detection involves defense, detection, and importantly, reaction to the intrusion attempts. An intrusion detection system should try to address each of these issues to a high degree. So network security becomes more complex due to the arrival of large no. of new type of attacks and lack of dynamic system to detect new types of attack. In this paper we define the solution to frequently changing network environment. We propose Online Adaboost-based parameterized method. It contains two models, Local model and Global model. In the local model, online Gaussian mixture models (GMMs) and online Adaboost processes are used as weak classifiers. A global detection model is constructed by combining the local parametric model. This combination is achieved by using an algorithm based on support vector machines (SVM) and particle swarm optimization (PSO). This system is able to detect new types of attack when network environment changes. It gives high detection rate and low false alarm rate.***

Keywords: Adaboost, detection rate, false alarm rate, network intrusions, parameterized model.

1. Introduction

An intruder can be defined as somebody attempting to break into an existing computer. This person is popularly termed as a hacker, black hat or cracker. The number of computers connected to a network and the Internet is increasing with every day. This combined with the increase in networking speed has made intrusion detection a challenging process. System administrators today have to deal with larger number of systems connected to the networks that provide a variety of services. The challenge here is not only to be able to actively monitor all the systems but also to be able to react quickly to different events. Overall intrusion detection involves defense, detection, and importantly, reaction to the intrusion attempts. An intrusion detection system should try to address each of these issues to a high degree.

The current practical solutions for NIDS used in industry are misuse-based methods that utilize signatures of attacks to detect intrusions by modelling each type of attack. As typical misuse detection methods, pattern matching methods search packages for the attack features by utilizing protocol rules and string matching. Pattern matching methods can effectively detect the well-known intrusions. But they rely on the timely generation of attack signatures, and fail to detect novel and unknown attacks. In the case of rapid proliferation of novel and unknown attacks, any defence based on signatures of known attacks becomes impossible. Moreover, the increasing diversity of attacks obstructs modelling signatures. Machine learning deals with automatically inferring and generalizing dependencies from data to allow extrapolation of dependencies to unseen data. Machine learning methods for intrusion detection model both attack data and normal network data, and allow for detection of unknown attacks using the network features. This proposed system will focuses on machine learning-based NIDS. The machine learning-based intrusion detection methods can be classified as statistics based, data mining based, and classification based. All the three classes of methods first extract low-level features and then learn rules or models that are used to detect intrusions.

New algorithms will be designed for local intrusion detection. The traditional online Adaboost process and a newly proposed online Adaboost process are applied to construct local intrusion detectors. The weak classifiers used by the traditional Adaboost process are decision stumps. The new Adaboost process uses online Gaussian mixture models (GMM) [1] as weak classifiers. In both cases the local intrusion detectors can be updated online. The parameters in the weak classifiers and the strong classifier construct a parametric local model. The local parametric models for intrusion detection are shared between the nodes of the network. The volume of communications is very small and it is not necessary to share the private raw data from which the local models are learnt. A PSO [8] and SVM [14]-based algorithm is proposed for combining the local models into a global detector in each node. The global detector that obtains information from other nodes obtains more accurate detection results than the local detector.

From the above discussion, we see the meaning of network intrusion. In particular we discussed in good detail which one encounters while using the traditional security. Usage of traditional security measures can have serious repercussions in the modern day intricacy of the web and the ubiquitousness and penetration of internet which make all the networks, and computers associated to it, more prone to such attacks. In this information age, when the value of data and information is critical, theft and misuse of data and

information is the top priority of all organizations. Thus, the need of modern network intrusion detection system has taken prime importance.

The rest of the paper is organized as follows: Section II introduces the overview of framework. Section III describes the local detection model. Section IV presents the method for constructing the global detection models. Section V shows the experimental results. Section VI summarizes the paper

2. Framework of System

In this system each node independently constructs its own local intrusion detection model according to its own data. There have been many survey of the field Dynamic DIDS. In particular Weiming Hu et al. [8] provide a comprehensive review of the online Adaboost-Based parameterized methods for Dynamic distributed network Intrusion detection which contain two models; Local Model and Global Model. Fig.2 gives an overview of framework that consists of the local models, and global models.

1) Local Models: Local model is constructed into each node by using weak classifiers and Adaboost-based training. So that each node contains a parametric model that consists of the parameters of the weak classifiers and the ensemble weights.
2) Global Models: It is constructed by combining all local parametric models by using PSO and SVM based algorithms. Global models are used to update local models and then updated models are shared by other nodes.
3) First we extract the features of incoming packets. There are total 41 features, all this features are grouped together and then passed to the SVM algorithm. After that PSO search the optimal results in KDD Dataset and output is created as a normal data or malicious data.

3. System Architecture

This section explains the overall process system architecture to find out intrusion in the network. Following figure1 shows the process architecture of the methodology used in the designed system.

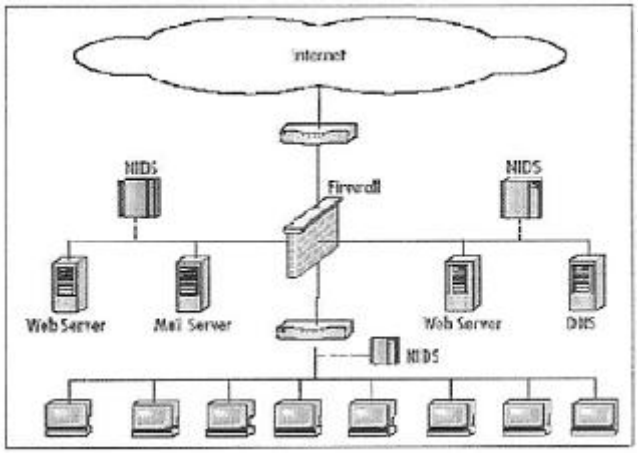

Figure 1: System Architecture

Background: The concept of intrusion detection system is based on machine learning algorithm. The designed system classifies the network data into normal or intrusion data.

3.1 Data Preprocessing

In network connection, three groups of features are commonly used for network intrusion detection: basic features of individual transmission control protocol (TCP) connections, content features within a connection, and traffic features computed using a two-second time window [12]. The extracted feature values form a vector $x = (x1, x2, \ldots xN)$, where N is the number of feature components. There are categorical and continuous features, and the value ranges of the features may differ greatly from each other. There are many types of attacks on the Internet. The attack samples are labeled as −1, −2, …. depending on the attack type, and the normal samples are all labeled as +1.

3.2 Local Model

Local model is constructed into each node by using weak classifiers and Adaboost-based training. So that each node contains a parametric model that consists of the parameters of the weak classifiers and the ensemble weights

A. Weak Classifiers

Weak classifier consist two types.

1) Decision stumps and normal behaviours for classifying attacks. The limitation of weak classifier is that the decision stumps do not consider the different types of attacks. This cause the influence in the performance of the Adaboost method.
2) Online GMMs that model a distribution of values of each factor component for each attack type.

Online GMM: For each type of attack or normal samples, we use a GMM. Let s ϵ {+1, -1,-2,....., -N} be a sample label where +1 represents normal samples and 1,-2,....., -N represent different types of attacks where N is number of different type of attacks, s represent the jth element of sample. The GMM model θcj on the jth feature component for the samples c is

$\theta cj = \{wcj(i), ucj(i), \sigma cj(i)\}ki=1$

Where,

k=number of GMM components indexed by i, w=weight, μ= mean, and σ= standard deviation. Where the computational complexity of the online GMM for one sample is O(k), which is higher than the decision stumps.

Design of the weak classifiers and the strong classifier, as shown in Figure 2

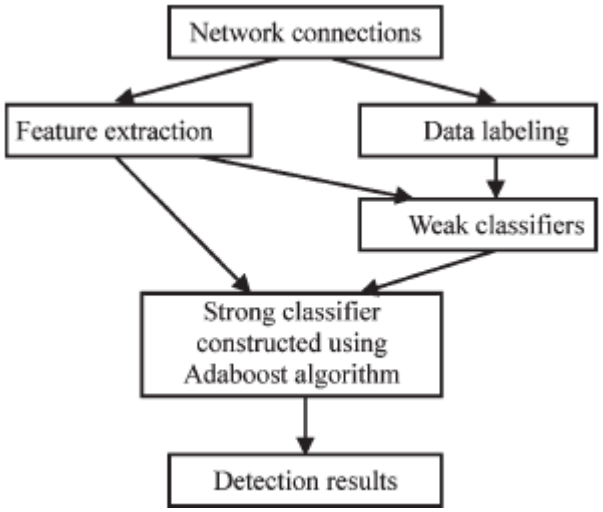

Figure 2: Framework of our algorithm.

New online Adaboost algorithm overcomes the limitation of traditional online Adaboost algorithm

The performance of algorithm is calculated by using detection rate and false alarm rate. And it depends on the initial weight of the training samples.

Let t be initial weight of each training sample

$$t = \begin{cases} \dfrac{(M_{normal} + M_{intrusion}) * r}{M_{normal}} & For\ normal\ connections \\ \dfrac{(M_{normal} + M_{intrusion}) * (1 - r)}{M_{intrusion}} & For\ network\ intrusion \end{cases}$$

Where M_{normal} is a number of normal sample, $M_{intrusion}$ is a number of attack sample and r Ɛ (0, 1). The value of r depends on the proportion of the normal samples, detection rate and the false alarm rate in specific applications.

3.3 Method for Constructing The Global Detection Model

Global detection model is constructed by combining the local parametric detection model from each node. This then used to detect intrusion on each distributed site.

Kittler et al. develop a different framework for combining the local model like, product rule, the sum rule, the max rule, the min rule, the median rule, and the majority vote rule. But by using this rule local detection model has two problem a) performance gap between the new type of attacks and local detection model. b) Dimension of the vector for similar test sample at the local models. The solution to this problem is combine local model by using PSO and SVM algorithms. PSO is a population search algorithm and the SVM is a learning algorithm, so by using the searching and learning ability of PSO and SVM respectively a global intrusion detection model is constructed in each node. The global intrusion detector constructed in the following simple manner:

$$G(n) = \begin{cases} -1 & \text{if there exist } C(n)=-1 \\ 1 & \text{else} \end{cases}$$

C(n) is final strong classifier generated by Adaboost training. Two things for global detection models are:
i) Global models constructed for all local nodes are uniform.
ii) The computational complexity of the PSO is O(QIA2L2) where I is the number of iterations, and L is the number of the training samples.

A) Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) [8] is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called lbest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and lbest locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations.

In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods.

Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

B)Support Vector Machine (SVM)

In machine learning, support vector machines (SVMs, also support vector networks)[14] are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are

divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data is not labeled, a supervised learning is not possible, and an unsupervised learning is required, that would find natural clustering of the data to groups, and map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering (SVC)[14] and is highly used in industrial applications either when data is not labeled or when only some data is labeled as a preprocessing for a classification pass; the clustering method was published.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

4. Mathematical Model

Mathematical Model for Proposed Work
Assumptions:

S: System; A system is defined as a set such that:
S = {I, P, O}.

Where,
U: Set of users
= {UR: Set of Registered Users,
UN: Set of Un-Registered Users}

I: Set of Input.
O: Set of output.
P: Set of Processes.

Input Set Details:

1. PHASE 1: REGISTRATION.
Ir= { username: i1,
Email ID: i2,
Password: i3,
}

2. PHASE 2: Communication
Iv= { USerID: i1,
FileContext: i2,
FileAttributes: i3
}

Process Set Details:

1. PHASE 1: REGISTRATION.
P1= { Userregistration: p11,
NetworkKey: p12}

2. PHASE 2: Communication
P2= { Storage: p21,
Session verification:p22,
commnication: p23,
attributeclassification: p24}

3. PHASE 3: Result
P3= { SR_Statistic : p31,
SR_Result : p32}

Output Set Details

1. PHASE 1: REGISTRATION.
O1= { userid: o11,
Password: o12 ,
SessionKey : o13
}

2. PHASE 2: Communication
O2= { Data: O21,
context Attributes: O22}

3. PHASE 3: Result
O3= { DR_Statistic : o31,
DR_Result : o32[Classified Data]}

5. Experimental Results

At the point when certain conditions are met, hubs might transmit their nearby models to one another. At that point, every hub can develop an altered worldwide model utilizing a little arrangement of preparing tests arbitrarily chose from the recorded preparing tests in the hub as indicated by the extent of different sorts of the system practices. When neighbourhood hubs pick up their own worldwide models, the worldwide models are utilized to distinguish interruptions; for another system association, the vector of the outcomes from the nearby models picked by the worldwide best molecule is utilized as the info to the worldwide model whose outcome figures out if the present system association is an assault.

We utilize the knowledge discovery and data mining (KDD) CUP dataset [15] to test algorithms. It has served as a reliable benchmark data set for many network intrusion detection algorithms. In this data set, each TCP/IP connection was labeled and 41 continues or categorical feature were extracted (41 features including 9 categorical features and 32 continuous features for each network connection). Attacks in the dataset fall into four main categories. i) Denial of service (DOS). ii) User to root (U2R). iii) Remote to local (R2L). iv) PROBE.

The number of sample of various types in the training set and in the test set are listed in Table1.

Table 1: The KDD CUP Dataset

Categorie	Training data	Test data

s		
Normal	97 278	60 593
DOS	391 458	223 298
R2L	1126	5993
U2R	52	39
Probing	4107	2377
Others	0	18 729
Total	494 021	311 029

Input is given as any kind of file (i.e. malicious and normal files) from client machine to server machine where our system exists. But here first we need to register for sharing the file with mail id and password as shown below. After that we can login with mail id and corresponding password, so now we can able to share the data.

After receiving the file in server first its features are extracted and this features grouped together by using SSMA algorithm. This features then passed to PSO and SVM algorithms. Finally this algorithm generates the result with the help of KDD CUP dataset.

The proposed circulated interruption identification calculation is tried with deferent hubs. The KDD CUP 1999 preparing dataset is part into six sections and every piece of information is utilized to develop a nearby identification model in a location hub. Along these lines, the span of preparing information in every hub is little, with the outcome that exact nearby interruption indicators can't be found in a few hubs.

In every hub, a worldwide model is built utilizing the neighborhood models. To reproduce a dispersed interruption recognition environment, the four sorts of assaults: neptune, smurf, portsweep, and satan in the KDD CUP 1999[15] preparing dataset are utilized for developing neighborhood discovery models, as tests of these four sorts take up 98.46% of the considerable number of tests in the KDD preparing dataset. Below Table demonstrates the preparation sets utilized for developing the worldwide models in the six hubs. It is seen that the sizes of the preparation sets are similarly little. Below specified table defines the cluster details with respect to kdd dataset with Attack details as below: Attacks detail with respect to KDD DATASET

Table 2: Attacks and cluster size detail with respect to KDD Dataset

Attack	Protocol	Cluster Size
back.	tcp	2203
buffer_overflow.	tcp	30
guess_passwd.	tcp	53
ipsweep.	tcp	94
ipsweep.	icmp	1153
loadmodule.	tcp	9
multihop.	tcp	7
neptune.	tcp	107201
nmap.	udp	25
nmap.	tcp	103
portsweep.	tcp	1039
portsweep.	icmp	1
rootkit.	udp	3
rootkit.	tcp	7
satan.	icmp	3
satan.	udp	170
satan.	tcp	1416

Following graph shows the attacks verses cluster details with respect to kdd dataset.

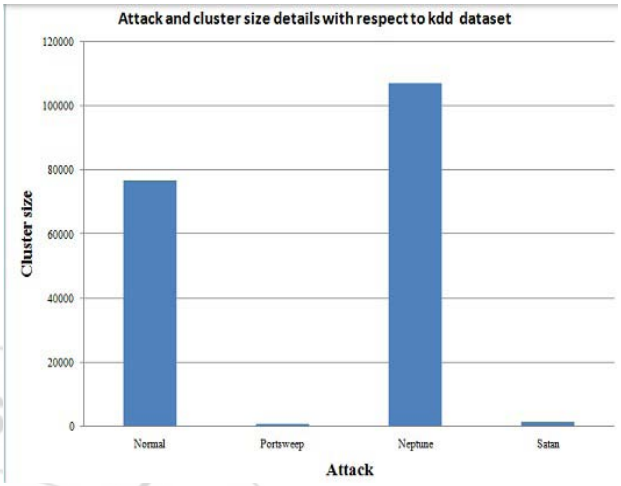

Figure 3: Attack and cluster size details with respect to kdd dataset

Below mentioned table express all activities found on respective machine with their all data input.

Table 3: Result for three local detection node using various algorithms

Algorithm	*Total test file*	*Malicious file*	*Normal file*	*Detection rate in %*	*False alarm Rate in %*
PSO	106	46	60	93.48	5
SVM	106	46	60	95.65	3.33
KNN	73	33	40	87.88	12.5
PSO + SVM	309	49	260	97.96	0.38

Following graph shows the detection and false alarm rate of various algorithms. These values are calculated from the large number of input files which contain both normal and malicious files.

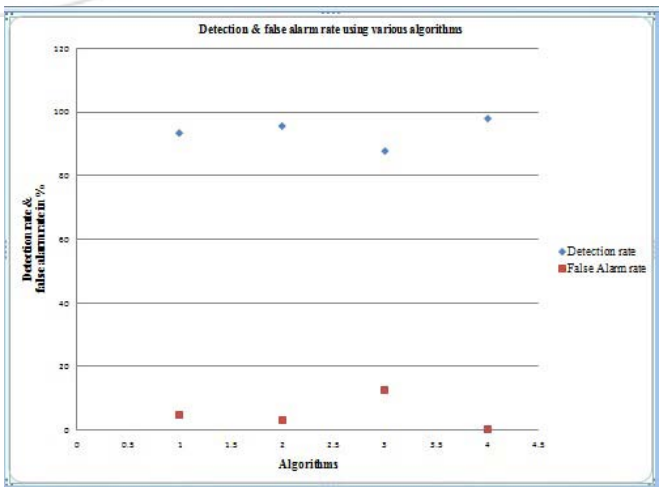

Figure 4: Detection and false alarm rate using various algorithms

From the result it is shown that the detection rate of PSO+SVM is high as compare to other algorithms and false alarm rate is low as compare to other algorithms like PSO, SVM and K-NN.

6. Conclusion

In the distributed intrusion detection framework, proposed the parameters in the online Adaboost algorithm formed the local detection model for each node, and local models are combined into a global detection model in each node using a PSO and SVM-based algorithm.

The advantages of this projects will be as follows: 1) Online Adaboost-based algorithms successfully overcome the difficulties in handling the mixed-attributes of network connection data; 2) the online mode in algorithms will ensures the adaptability of algorithms to the changing environments; the information in new samples will be incorporated online into the classifier, while maintaining high detection accuracy; 3) local parameterized detection models will be suitable for information sharing: only a very small number of data will shares among nodes; 4) no original network data will be shared in the framework so that the data privacy is protected; and 5) each global detection model will improve considerably on the intrusion detection accuracy for each node.

And main aim of this project is to maintain detection rate high and false alarm rate low.

References

[1] Weiming Hu, Jun Gao, Yanguo Wang, Ou Wu, and Stephen Maybank," Online Adaboost-Based Parameterized Methods for Dynamic Distributed Network Intrusion Detection," IEEE TRANSACTIONS ON CYBERNETICS, VOL. 44, NO. 1, JANUARY 2014
[2] Weiming Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., vol. 38, no. 2, pp. 577–583, Apr. 2008.
[3] D. Denning, "An intrusion detection model," IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. 1987.
[4] Yan-guo Wang, Xi Li, and Weiming Hu: "Distributed Detection of Network Intrusions Based on a Parametric Model".
[5] S. Mabu, C. Chen, N. Lu, K. Shimada, and K. Hirasawa, "An intrusiondetection model based on fuzzy class-association-rule mining using genetic network programming," IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev., vol. 41, no. 1, pp. 130–139, Jan. 2011.
[6] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," IEEE Trans. Syst., Man, Cybern., Part C:Appl. Rev., vol. 38, no. 5, pp. 649–659, Sep. 2008
[7] Yamille del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems" IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 12, NO. 2, APRIL 2008
[8] J. Kennedy, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Netw., 1995, pp. 1942–1948.
[9] Z. Zhang and H. Shen, "Online training of SVMs for real-time intrusion detection," in Proc. Adv. Inform. Netw. Appl., vol. 2, 2004, pp. 568–573.
[10] Muhammad Qasim Ali, Ehab Al-Shaer, and Taghrid Samak, "Firewall Policy Reconnaissance: Techniques and Analysis" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 2, FEBRUARY 2014
[11] Purvag Patel, Chet Langin, Feng Yu, and Shahram Rahimi, "Network Intrusion Detection Types and Computation" (IJCSIS) International Journal of Computer Science and Information Security, Vol. 10, No. 1, January 2012
[12] D. Smallwood and A. Vance, "Intrusion analysis with deep packet inspection: Increasing efficiency of packet based investigations," in Proc. Int. Conf. Cloud Service Computing, Dec. 2011, pp. 342–347.
[13] W. Lee, S. J. Stolfo, and K. Mork, "A data mining framework for building intrusion detection models," in Proc. IEEE Symp. Security Privacy, May 1999, pp. 120–132.
[14] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in Proc. Int. Joint Conf. Neural Netw., vol. 2. 2002, pp. 1702–1707.
[15] B. Pfahringer, "Winning the KDD99 classification cup: Bagged boosting," SIGKDD Explorations, vol. 1, no. 2, pp. 65–66, 2000.
[16] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," ACM Trans. Inform. Syst. Security, vol. 3, no. 4, pp. 227–261, Nov. 2000.

Author Profile

Amol Shahaji Jadhav is a M.E. 2nd year Computer Engineering Student from Dr. D.Y. Patil COE, Ambi - Talegaon, Savitribai Phule Pune University, Maharashtra, India. He has completed B.E. in Information Technology from K.K. Wagh COE, Nashik. He's current research in Dynamic Distributed Network Intrusion Detection.