Development of Hybrid-Multi-Stages Intrusion Detection Systems

M.A. Mohamed[1] and M.A. Mohamed[2]
[1]*Faculty of Engineering, Mansoura University*
[2] *SCADA & Telecom. Assist. General Manager, GASCO, Egypt*

Summary
Most intrusion detection systems (IDSs) are based on a single algorithm that is designed to either model normal behavior patterns or attack signatures in network data traffic. Most often, these systems fail to provide adequate alarm capability that reduces false positive and false negative rates. We had proposed multi-stages approaches to enhance the overall performance of IDSs. All models implemented in this paper, must have a perfect 2-classes classifier to differentiate between attacks & normal patterns, so we grant to detect attacks at first stage of IDS and secure the protected system, through other stages we tried to identify the name of intrusion to increase the efficiency of IDS. The first stage is highly capable in detecting normal signature and diverse what-else to attacks category, so it is capable in detecting unseen or unknown attacks. The results of the proposed techniques had shown that a very high increase in the performance of IDS systems. The practical results showed that the multistages system composed of MLP and improved hybrid J48-DT provided the best results among all discussed systems.

Key words:
Intrusion detection systems (IDSs); knowledge discovery and data mining (KDD); Multilayer Perceptron (MLP)

1. Introduction

Recently, the size of internet and volume of traffic have grown steadily. This expansion and increase in computerization generally have also seen a rise in computer misuse and attacks on networks. Prevention of such crime is impossible and so, monitoring and detection are resorted as the best alternative line of defense; the implementation of this process, called IDS. It is performed with the aid of dedicated software/hardware systems operating on security logs, audit data or behavior observations. IDS also needs to process very large amounts of audit data and are mostly based on hand-crafted attack patterns developed by manual encoding of expert knowledge [1].

1.1 What is Intrusion Detection?

With the increase of attacks on computers and networks in recent years, improved and essentially automated surveillance has become a necessary addition to information technology security. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions [1]. Intrusions are attempts to compromise the confidentiality, integrity and availability of a computer or network or to bypass its security mechanisms. They are caused by attackers accessing a system from Internet, by authorized users of the systems who attempt to gain additional privileges for which they are not authorized, and by authorized users who misuse privileges given to them.

1.2 Main Benefits and Characteristics

The main benefits of IDS include: (i) detecting attacks and other security violations, which have not been prevented by other primary protection techniques; (ii) preventing problem-behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system; (iii) presenting traces of intrusions, allowing improved diagnosis, recovery and corrective measures after an attack; (iv) documenting the existing threat from inside and outside a system, permitting security management to realistically assess risk and adapt its security strategy in response, and (v) acting as quality control for security design and implementation (highlighting some deficiencies or errors, before serious incidents occur) [2].

Much work has been done to implement these features, so that now over 150 commercial, freeware and shareware IDSs are available.

To facilitate evaluation of these solutions, Purdue University IDS research project put a list of characteristics for good systems: (i) it must run continually without human supervision. The system must be reliable enough to allow it to run in the background of the system being observed. That is, its internal workings should be examinable from outside; (ii) it must be fault tolerant in the sense that it must survive a system crash and not lose its knowledge-base at restart; (iii) it must resist subversion; (iv) the system can monitor itself to ensure that it has not been subverted; (v) it must impose minimal overhead on the system; a system that slows a computer to a crawl will simply not be used; (vi) it must observe deviations from normal behavior; (vii) it must be easily tailored to the

Manuscript received March 5, 2010
Manuscript revised March 20, 2010

system in question; every system has a different usage pattern, and the defense mechanism should adapt easily to these patterns, and (viii) it must cope with changing system behavior over time as new applications are being added. The system profile will change over time; i.e. it must be adaptable [3].

2. Overview of IDS Techniques

In general IDSs may be analyzed as misuse/anomaly based detection and network-based/host-based systems.

2.1 Misuse-Based Detection

Misuse detection depends on the prior representation of specific patterns for intrusions, allowing any matches to them in current activity to be reported. Patterns corresponding to known attacks are called signatures, also giving rise to the term signature-based detection. These systems are unlike virus-detection systems; they can detect many known attack patterns and even variations; thereof but are likely to miss new attacks. Regular updates with previously unseen attack signatures are necessary [4].

2.2 Anomaly-Based Detection

Anomaly detection identifies abnormal behavior. It requires the prior construction of profiles for normal behavior of users, hosts or networks; therefore, historical data are collected over a period of normal operation. IDSs monitor current event data and use a variety of measures to distinguish between abnormal and normal activities. These systems are prone to false alarms, since user's behavior may be inconsistent and threshold levels will remain difficult to fine tune. Maintenance of profiles is also a significant overhead but these systems are potentially able to detect novel attacks without specific knowledge of details. It is essential that normal data used for characterization are free from attacks [4].

2.3 Network-Based IDS Systems

Network-based IDS monitors traffic by capturing and analyzing network packets. Advantages of network-based IDSs are: (i) the deployment of these systems has little impact on the existing network; (ii) little effect on the normal network operation and are relatively easy to upgrade, and (iii) robust in the face of attacks and can be made invisible to attackers. On the other hand, the disadvantages are: (i) during peak-traffic periods some packets may go unprocessed and attacks undetected; (ii) encrypted information cannot be analyzed; (iii) attack attempts may be detected but hosts must usually then be investigated manually to determine whether or not they were penetrated and damage caused, and (iv) attacks involving fragmentation of packets can cause these IDS to crash [5].

2.4 Host-Based IDS Systems

Host-based IDS monitors network traffic of a particular host and some system events on the host itself. One may be installed on each host or simply on some chosen critical ones within a network. Advantages of host-based IDSs are: (i) some local events on hosts can only be detected; (ii) raw data are available for analysis in non-encrypted form, and (iii) software integrity checks can be used in the detection of certain types of attack (e.g. Trojan horse). In addition, it has the following disadvantages: (i) more complex to manage; (ii) may be disabled if host is attacked and compromised; (iii) not suitable for network attacks involving distributed scans and probes; (iv) can be disabled by overload attacks (e.g. denial of service); (v) for large amounts of information to be processed, local storage may be necessary, and (vi) use host's own computing resources at a cost to performance [5].

3. Performance Indices

Important measures of efficiency of IDSs are false-alarm rates; the percentage of time-consuming false positives registered- normal data detected falsely as an intrusion and the percentage of more dangerous false negatives; intrusions falsely classified as normal data. Such measurements do not indicate the human workload required in analyzing false alarms generated by normal background traffic. Low false-alarm rates combined with high detection rates mean; the detection outputs can be trusted [6].

4. Data Collection

Defense Advanced Research Projects Agency (DARPA) intrusion-detection evaluation datasets were the original source of data most directly relevant to this work. For 1998 DARPA datasets, 7-weeks (about 4 GBytes of compressed binary tcpdump data) of training data were accumulated from multi-system testbed, to represent basically normal operation spiced with a series of automatically or manually launched attacks. Further 2-weeks of test data were collected containing additional new and novel intrusions [7].

The Knowledge Discovery and Data Mining (KDD) Cup 1999 are the datasets, which were issued for use in the KDD '99 Classifier-Learning Competition [8]. This was preprocessed with the feature-construction framework MADAM-ID, to produce about 5×10^6 connection records.

A connection is defined to be a sequence of TCP packets starting and ending at some well-defined times, between which data flow to and from a source IP address to a destination IP address under some well-defined protocol. Each connection is labeled as either normal or with the name of its specific attack. A connection record consists of about 100 bytes [9]. A 10% of the complementary 2-weeks of test data were, likewise, preprocessed to yield a further less than half-a-million connection records. It was stressed that these test data were not from the same probability distribution as the training data and that they included specific attack types not found in the training data. A total of 22 attack types were included in the training data.

5. Attack Categorization

Simulated attacks were classified, according to actions and goals of the attacker. Each attack falls into one of the following: (i) Denial-of-service (DoS) have the goal of limiting/denying services provided to a user, computer or network; a common tactic is to severely overload the targeted system (e.g. SYN flood); (ii) Probing (PRB) have a goal of gaining knowledge of existence or configuration of computer system or network; port scans/sweeping of a given IP-address range are typically used in this category (e.g. IPsweep); (iii) Remote-to-Local (R2L) have a goal of gaining local access to a computer or network to which attacker previously only had remote access; e.g. attempts to gain control of a user account, and (iv) User-to-Root (U2R) have a goal of gaining root/super-user access on a particular system on which attacker previously had user level access; attempts by a non-privileged user to gain administrative privileges (e.g. Eject).

6. KDD Features

In the KDD'99 data [8], the initial features extracted for a connection record include the basic features of an individual TCP connection, such as: its duration, protocol type, number of bytes transferred and the flag indicating normal or error status of a connection. These intrinsic features provide information for general network-traffic analysis purposes. Since most DoS and Probe attacks involve sending a lot of connections to the host(s) at the same time, they can have frequent sequential patterns, which are different to the normal traffic.

For these patterns, same host feature examines all other connections in the previous 2-secs, which had the same destination as the current connection. Similarly, same service feature examines all other connections in the previous 2-secs, which had the same service as the current connection. Temporal and statistical characteristics are referred to as time-based traffic features; there are several Probe attacks which use a much longer interval than 2-secs (e.g., one minute) when scanning hosts or ports; mirror set of host-based traffic features were constructed based on a connection window of 100 connections.

The R2L and U2R attacks are embedded in the data portions of the TCP packets and may involve only a single connection. To detect these, connection features of an individual connection were constructed using domain knowledge [10]. These features suggest whether the data contains suspicious behavior, such as: number of failed logins, successfully logged in or not, whether logged in as root, whether a root shell is obtained, etc. In general, there are 42 features (including the attack name) in each connection record, with most of them taking on continuous values.

7. Elements of multi-stage Classifiers

A brief description of the single stage classifiers used in developing the proposed multistage systems will be presented in the following subsections. In this paper, four different classifiers were used: (i) MLP; (ii) SVM; (iii) Naïve-Bayes, and (iv) decision tree.

7.1 Multilayer Perceptron (MLP)

MLP is a layered feed forward networks typically trained with static back propagation. These networks have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than network weights) [11-13].

7.2 Support vector machine (SVM)

SVMs are a set of related supervised learning methods used for classification and regression. An SVM constructs a hyperplane or set of hyperplanes in a high-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 points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

SVM is implemented using the kernel Adatron algorithm. The kernel Adatron maps inputs to a high-dimensional feature space, and then optimally separates data into their respective classes by isolating those inputs which fall close to the data boundaries. Therefore, the kernel Adatron is

especially effective in separating sets of data which share complex boundaries. SVMs can only be used for classification, not for function approximation [14].

7.3 Naive-Bayes (NB)

Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong naive independence assumptions. A more descriptive term for the underlying probability model would be independent feature model. In simple terms, NB classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Depending on the precise nature of the probability model, NB classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for NB models uses the method of maximum likelihood; in other words, one can work with the NB model without believing in Bayesian probability or using any Bayesian methods. In spite of their Naive design and apparently over-simplified assumptions, these classifiers often work much better in many complex real-world situations than one might expect. An advantage of the NB classifier is that it requires a small amount of training data to estimate the parameters (means and variances). Because independent variables are assumed, only the variances of each class need to be determined and not the entire covariance matrix [15].

7.4 Decision Tree Model (DT)

DT is one of the most used machine learning techniques in intrusion detection field. This technique builds a tree structure of attack signature using anomalous log data as in. Moreover, the normal behavior of a system or a user can be traduced in a tree structure as in. The decision tree technique was applied both for misuse and anomaly detection either for network or single host. The DT classifier consists of decision and leaf nodes. Each decision node corresponds to a test over a single attribute of the given instances. It has different branches on other decision or leaf nodes that represent the possible values of the actual feature. Leaf nodes represent the possible attack and normal class labels that can serve as an output when classifying a new example.

During this paper, J48-DT has been used. J48 is a version of an earlier algorithm developed by J. Ross Quinlan, the very popular C4.5. Decision trees are a classic way to represent information from a machine learning algorithm, and offer a fast and powerful way to express structures in data. The J48 algorithm gives several options related to tree pruning. Many algorithms attempt to "prune", or simplify, their results.

The basic algorithm recursively classifies until each leaf is pure, meaning that the data has been categorized as close to perfectly as possible. This process ensures maximum accuracy on the training data, but it may create excessive rules that only describe particular idiosyncrasies of that data. When tested on new data, the rules may be less effective. Pruning always reduces the accuracy of a model on training data. This is because pruning employs various means to relax the specificity of the decision tree, hopefully improving its performance on test data. The overall concept is to gradually generalize a decision tree until it gains a balance of flexibility and accuracy [16-18].

8. The proposed hybrid IDS techniques

We had proposed multistage approaches capable of enhancing the overall performance of IDSs. All models implemented in this paper, must have a perfect 2-classes classifier to distinguish between ‘attack’ and ‘normal’ patterns. So we grant to detect ‘attack’ at first stage of IDS and secure the protected system (host or network). Through other stages we try to identify intrusion to increase the efficiency of IDS. So the first stage is highly capable in detecting ‘normal’ signature and diverse what-else to ‘attacks’ category, so it is capable in detecting ‘unseen” or ‘unknown” attacks. For example, in the first model, multistage-MLP-IDS, its first stage, ‘attacks’ and ‘normal’ categories are highly clustered.

During the next stages, MLP classifier will be used to cluster identical intrusions in a separate group and leave the rest of intrusions in another group. The proposed algorithm mentioned above is based on empirical observations for confusion matrix results from single MLP at each stage. From confusion matrix, it’s noticed that some intrusions are highly classified and others are poorly classified. Poorly classified intrusions are grouped together in one group and named with any dummy name. That MLP is re-tested only with highly classified intrusions and that dummy group. If the detection rate is still high, the dummy group is tested separately in another MLP. The same procedure is repeated and according to the confusion matrix for this dummy group, highly detected intrusions are separated and group the poorly classified in another dummy group. This procedure is repeated until detecting intrusions as much as possible. In some situations, no more intrusions could be detected and no better recognition rate could be established so in these cases the procedures are stopped.

We proposed empirical hybrid multistage IDS classifiers; based on the following assumptions: (i) we insist to use standard dataset not tailored dataset like most papers. The original 10% KDD dataset (494021 records) is reduced to 145587 records after removing duplications; (ii) all

intrusions must be highly recognized with percent not less 97%; i.e. any intrusion recognized with any percent less than 97% is supposed to be poorly classified; (iii) first stage must be perfect classifier to distinguish between 'attacks' and 'normal'; (iv) ''G1, G2…Gn" are dummy group names, and (v) normal and highly classified intrusions will be indicated with bolded frames in the following flow charts of the results.

8.1 Multi-Stage MLPs

Firstly, we propose a hierarchical multistage scheme for combining multiple MLPs; Fig.1. According to acceptable classification rates stated above, this system succeeded to recognize all 22 attacks in addition to normal data stream.

8.2 Multi-Stage SVMs

Secondly, we propose a hierarchical multistage scheme for combining multiple SVMs; Fig.2. We tried to implement SVM in first stage to classify "normal/attacks"; unfortunately, it was found that SVM consumes huge amount of time so we implemented MLP in first stage. This system showed that it is able to discriminate between 4 intrusion classes but totally, it recognized 11 of 22 attacks, in addition to normal data stream.

8.3 Multi-Stage Naïve Bayes

Thirdly, we propose a hierarchical multistage scheme for combining multiple naïve-Bayes; Fig.3. At first we tried this algorithm to classify ''normal/attacks'' but it showed 97.9%/94.7% respectively so using this classifier in the first stage is excluded. So MLP is used as first stage, also using this algorithm as 5-classes classifier showed the results (normal=86.5%, u2r=90.4%, DoS=96.1%, r2l=38.3%, prb=89.7%). So we used MLP as first stage and continue with this algorithm. The final results of this algorithm recognized 6 of 22 attacks, in addition to normal data stream.

8.4 Multi-Stages J48-DT

Finally, we propose a hierarchical multistage scheme for combining multiple J48-DTs; Fig.4. This algorithm perfectly discriminate between "normal" and "attacks" as first stage, finally it recognized 12 of 22 attacks, in addition to normal data stream. Fig.5. is an improved multistage J48 implemented and in its final stage, SVM is implemented to recognize the rest of attacks.

Conclusion

Three categories of single stage IDS based classifiers: (i) ANN-based; (i) Naïve-Bayes, and (iii) DT had been combined to provide what we had called hybrid-multistage IDSs. The combination algorithms had been developed empirically to form four hybrid classifiers: (i) MLP-based; (ii) SVM-based; (iii) Naïve-Bayes-based, and (iv) J48-DT-based. Among all the classifiers tested multistage-MLP and hybrid-multistage-J48 provided 100% recognition rate for both normal and all types of attacks. This high rate comes on the cost of processing time and hardware complexities.

References

[1] R. Power, "CSI/FBI computer crime & security survey," Computer Security Journal, Vol.18, No.2, pp:7-30, 2002.

[2] A. Delamer, "intrusion detection with data mining," Master's thesis, Donau University, Krems, Austria, 2002.

[3] R. Bace and P. Mellm "NIST special publication on intrusion detection systems", Computer security resources, national institute of standards and technology, pp: 1-51, 2002.

[4] T. Verwoed and R. Hunt, "intrusion detection techniques and approaches," Elsevier: computer communications, Vol.25, No.10, pp: 1356-1365, 2002.

[5] S. Chobrolu, A. Abraham, P. Johnson, "feature deduction and ensemble design of intrusion detection systems," Elsevier computers & security, Vol.24, pp: 195-307, 2005.

[6] A. Tartakovsky, et al., "detection intrusion in information system by sequential change-point methods," Elsevier, statistical methodology, Vol.3, pp: 252-293, 2006.

[7] R. Lippmann, et al., "1999-DARPA offline intrusion detection evaluation," Computer networks, Vol.34, pp: 579-595, 2000.

[8] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html/

[9] G. Mum, Y. Kim, et al., "network intrusion detection using statistical probability distribution," information systems & information technology, Vol. 3984, pp: 340-348, 2006.

[10] H. Kayacik, A. Zincir-Haywood, and M. Haywood, "selecting features for intrusion detection: a feature relevance analysis on KDD99 intrusion detection datasets," *Dalhousie University, 2005.*

[11] R. Beghdad, "Critical study of neural networks in detecting intrusions" computers & security, Vol. 27, No. 5, pp: 168-175, 2008.

[12] Y. Yu, Y. Wei, et al., "anomaly intrusion detection approach using hybrid MLP/CNN neural network," intelligent systems design & applications. ISDA 6th int. conference, Vol.2, issue.16-18, pp: 1095-1102, 2006.

[13] J. Skaruz, "recurrent neural networks on duty of anomaly detection in databases," proceedings of 4th international symposium on neural networks: advances in neural networks part III, pp: 85-94, 2007.

[14] L.P. Sinclair and S. Matzner, "An Application of Machine Learning to Network Intrusion Detection," proceedings of the 15th annual computer security applications conference, ACSAC'99, p.371-378, 1999.

[15] K. Huang, I. King, M.R. Lyu, "Finite Mixture Model of Bounded Semi-naive Bayesian Networks Classifier," 10th International Conference on Neural Information Processing (ICONIP-2003), PP: 115-122, 2003.

[16] C. Kruegel and T. Toth, "Using Decision Tree to Improve Signature Based Intrusion Detection," 6th Symposium on

Recent Advances in Intrusion Detection (REID), Vol. 2820, PP: 173-191, 2004.

[17] M.D. Twa, S. Parthasarathy, M.A. Bullimore, et. al., "Automated decision tree classification of keratoconus from Videokeratography. Investigative Ophthalmology and Vision Science, E-Abstract, Vol.1082, No.46, ARVO, 2005.

[18] G. Stein, B. Chen, A.S. Wu, and K. A. Hua, "decision tree classifier for network intrusion detection with GA-based feature selection," Proceedings of 43rd annual Southeast regional conference, Vol.2, pp:136-141, 2005.

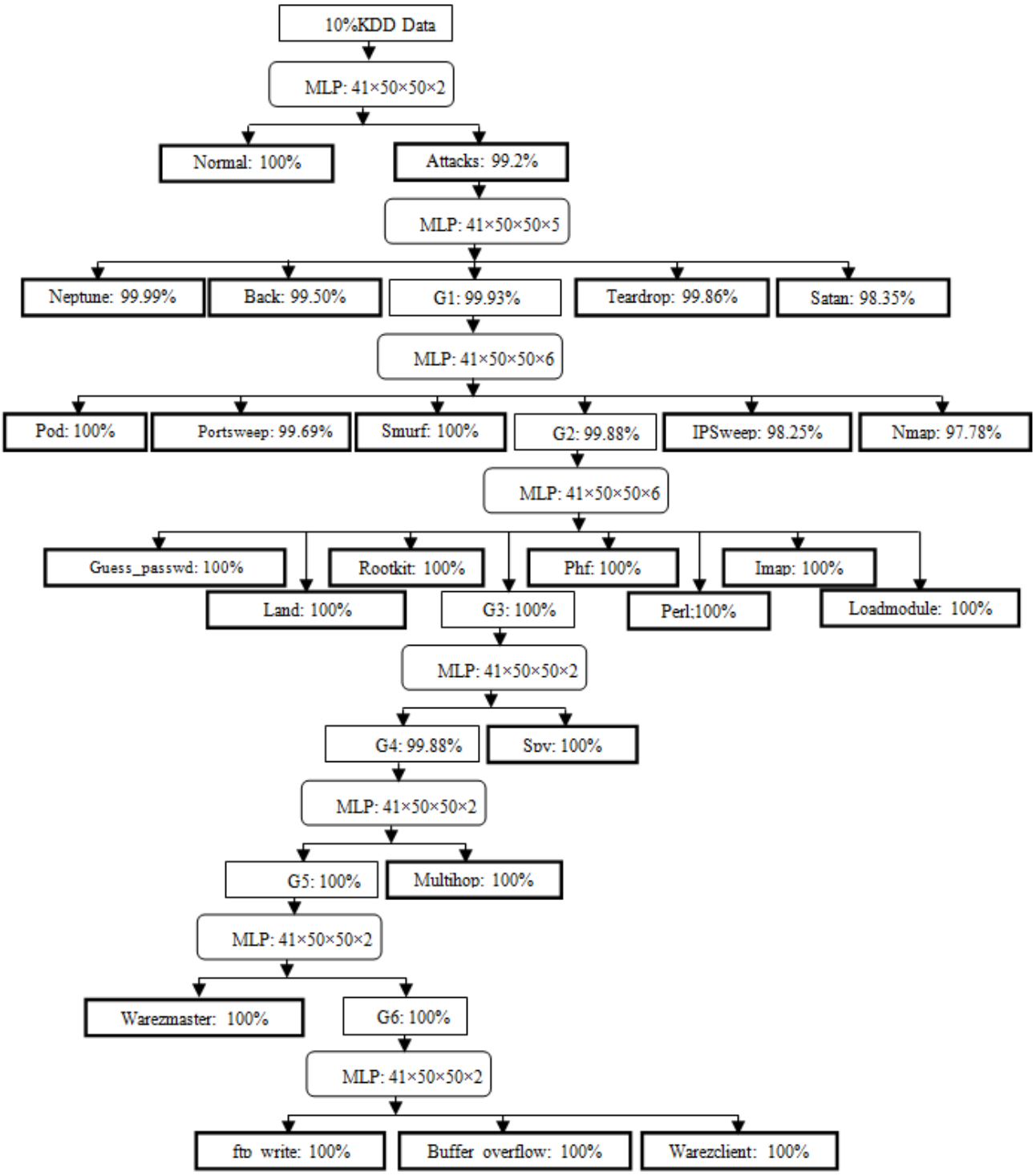

Fig.1 Multistage MLPs

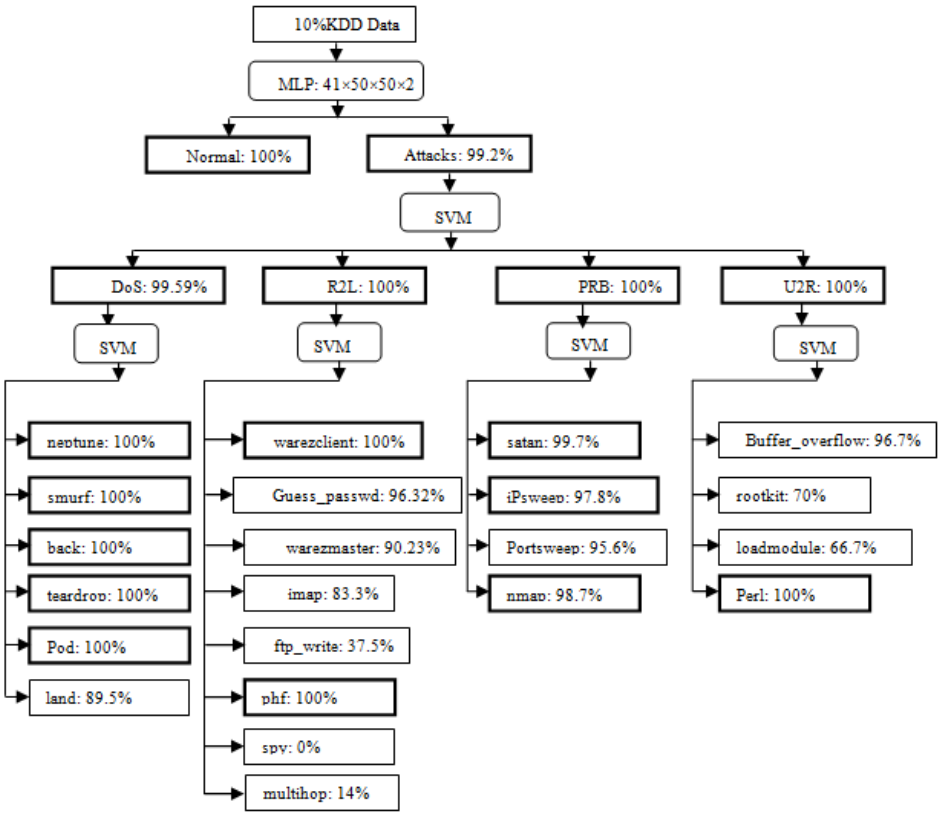

Fig.2 multistage SVMs

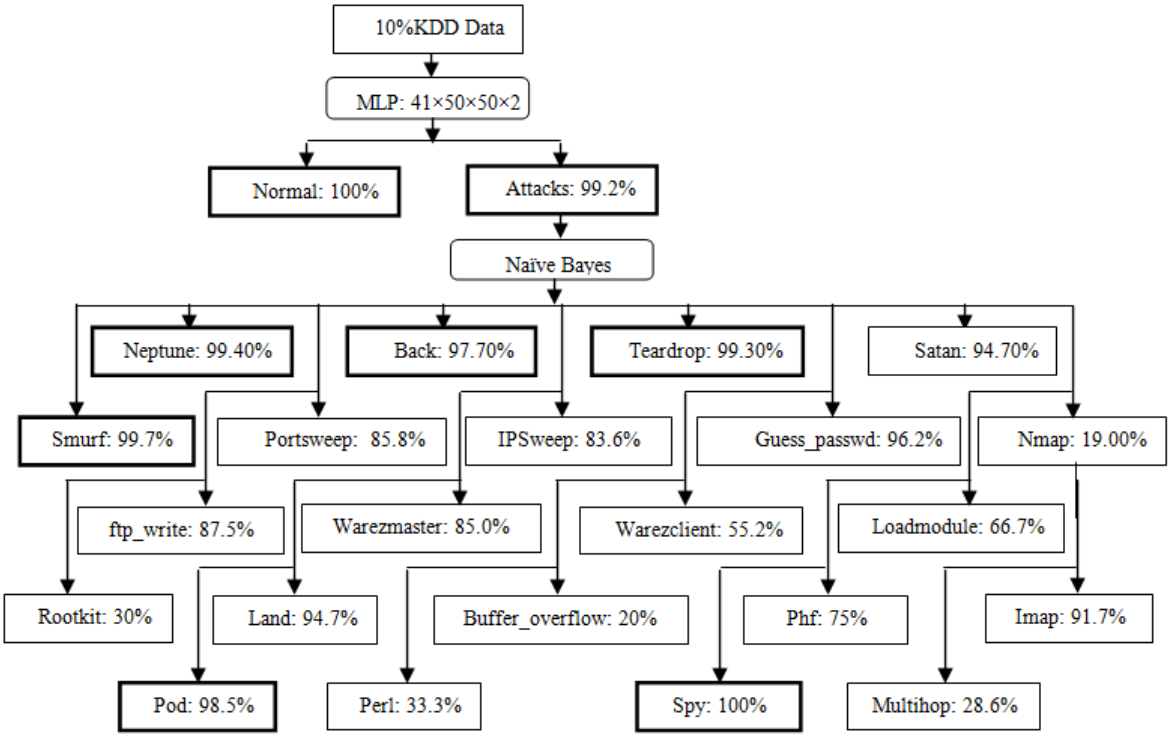

Fig.3 multistage naïve-Bayes

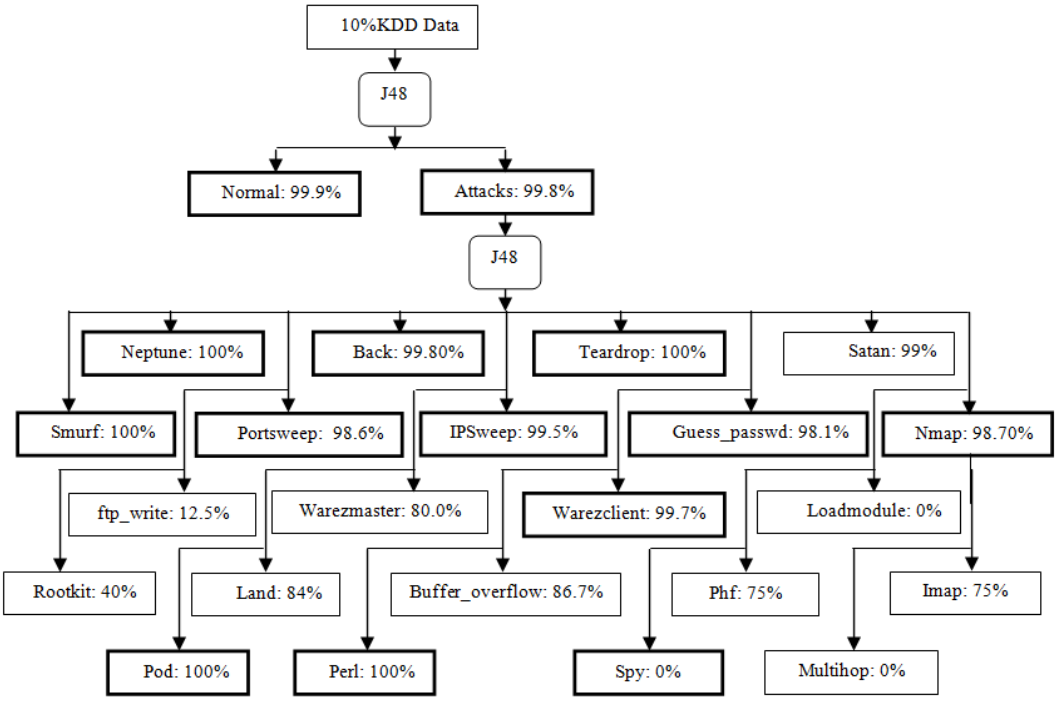

Fig.4 multistage J48 DT

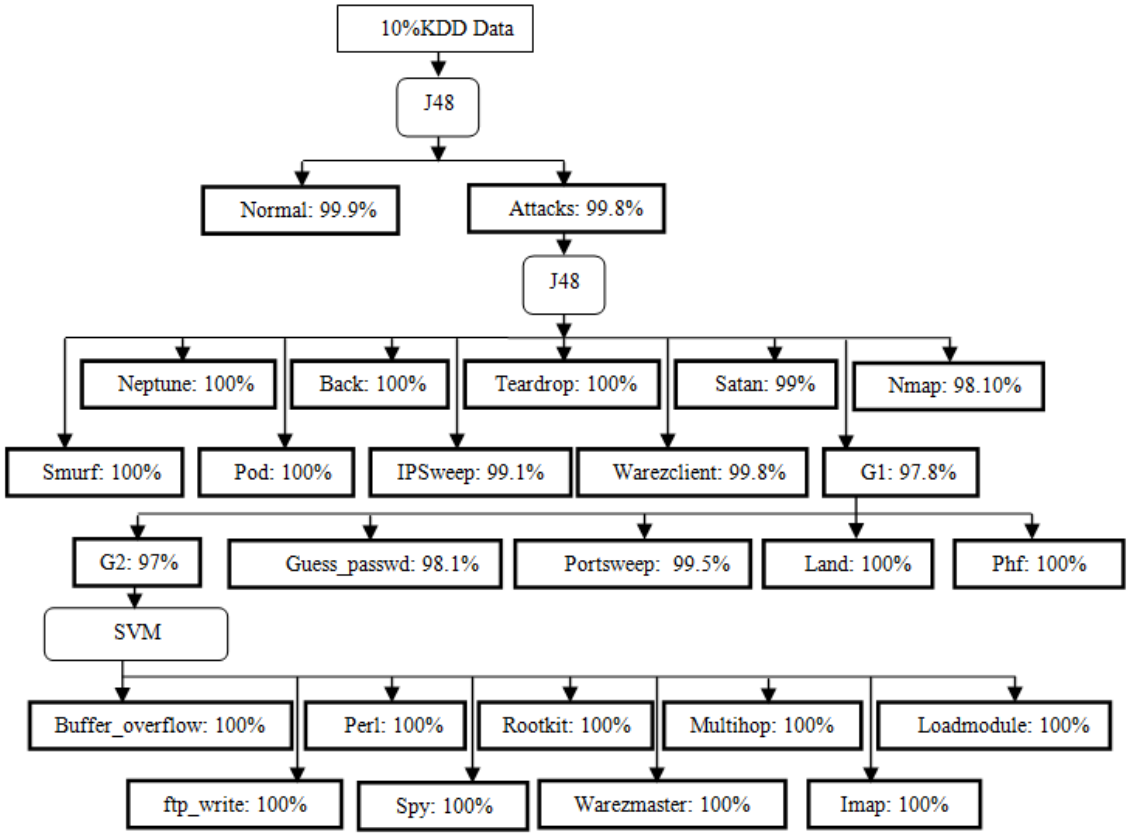

Fig.5 improved multistage J48 DT