

# Applying Knowledge Discovery in Database Techniques in Modeling Packet Header Anomaly Intrusion Detection Systems

Solahuddin B Shamsuddin

School of Informatics, University of Bradford, Bradford BD7 1DP, United Kingdom  
Email: S.B.Shamsuddin@Bradford.ac.uk

Mike E Woodward

School of Informatics, University of Bradford, Bradford BD7 1DP, United Kingdom  
Email: M.E.Woodward@Bradford.ac.uk

**Abstract**—This paper describes packet header anomaly intrusion detection system modeling. The essence of the discussion in this paper is on applying knowledge discovery in database technique to produce expert production rules which is one of the main components of our model which we call as Protocol based Packet Header Anomaly Detector (PbPHAD) Intrusion Detection System. PbPHAD is designed to detect the anomalous behavior of network traffic packets based on three specific network and transport layer protocols namely UDP, TCP and ICMP to identify the degree of maliciousness from a set of detected anomalous packets identified from the sum of statistically modeled individually rated anomalous field values.

**Index Terms**—Anomaly, Intrusion Detection Systems, Knowledge Discovery in Database, Expert Production Rules.

## I. INTRODUCTION

Intrusion Detection Systems (IDS) has been part and parcel of essential key components of an overall security architecture in any computer network [1]. A significant number of research efforts have been geared in this area especially in the design and development of anomaly based IDS as this model has emerged to be a more promising model in detecting unknown attacks or more popularly known as zero day attacks which could come from any malicious hosts in any corners of the globe which appear hastily in today's interconnected computer architectures.

One of the main focus in designing anomaly based IDS is to come up with a model that could give a high detection rate with an acceptable number of false alarm rates as high false alarm rates would significantly reduce the effectiveness of the IDS. Reducing false alarm rates have been the main concern in anomaly based IDS design and it has been the most challenging task to achieve it. A variety of ensemble techniques [2] have been applied by a lot of researchers in their quest to come up with the best

algorithm to produce the expert production rules to deduce the classification of anomalous packets which deem to be malicious from a plethora of incoming packets traversing into any monitored network segment of a particular interest. New trends in IDS research modelling are focused more towards into performing sophisticated protocol analysis and embedding expert production rules in the detection algorithms such that the use of attack signatures has become less dependent [3].

Even though the use of anomaly based IDS is the current trend, the use of signature based IDS is still very much in need as the former model still has not reached its maturity stage yet and as such a lot of research efforts are very much going on in gearing to perfecting the model. We believe, for the time being, a hybrid approach shall be the best approach in making full use of the best advantages of both models [4]. i.e. the combination of high level of detection accuracy of signature based IDS with low false positive rates and the ability to detect unknown attacks or *zero day* attacks of anomaly based IDS.

In this paper, we will discuss our work in modelling our IDS by applying knowledge discovery in database (KDD) techniques in extracting expert production rules which can be embedded in the detection algorithm to reduce the level of false positive to a fairly acceptable rate. We took this approach as rule-based expert systems is the most popular choice for building knowledge-based systems which can be found in a lot of artificial intelligence literatures [5].

The rest of the paper is organized as follows. In section II, we discuss other related works in intrusion detection systems. In section III, we describe our anomaly based IDS model which include its design concept and statistical modelling. In section IV we discuss the life cycle of our IDS modelling process and data engineering process in applying knowledge discovery in database

technique to our IDS model. We discuss our model's experimental results using 1999 DARPA evaluation data set in section V. In section VI we discuss the comparison of our results with the 1999 DARPA IDS evaluation system results on poorly detected attacks. We present our conclusion in section VII.

II. RELATED WORK

Peddabachigari et al. studied two hybrid approaches for modelling IDS where Decision Trees and Support Vector Machines are combined as hierarchical hybrid intelligent system model. They also came up with an ensemble model combining the base classifiers. Their results shows that the ensemble approach produced better results compared to the individual classifiers and the hybrid models. [6].

IDES (Intrusion Detection Expert System) [7] exploited the statistical approach for the detection of intruders. It uses the intrusion detection model proposed by Denning [8] and audit trails data as suggested in Anderson [9]. IDES maintains profiles, which are a description of a subject's normal behavior with respect to a set of intrusion detection measures. Profiles are updated periodically, thus allowing the system to learn new behavior as users alter their behavior. These profiles are used to compare the user behavior and inform significant deviation from them as the intrusion. IDES also uses the expert system concept to detect misuse intrusions.

The advantage of this approach is that it adaptively learns the behavior of users, which is thus potentially more sensitive than human experts. This system has several disadvantages. The system can be trained for certain behavior gradually making the abnormal behavior as normal, which may make the intruders undetected. Determining the threshold above which an intrusion should be detected is a difficult task. Setting the threshold too low results in false positives (normal behavior detected as an intrusion) and setting it too high results in false negatives (an intrusion undetected). Attacks, which occur by sequential dependencies, cannot be detected, as statistical analysis is insensitive to order of events.

ADAM - (A Testbed for Exploring the Use of Data Mining in Intrusion Detection) observe IP addresses and subnets, port numbers and TCP state to build normal traffic models. This model will be used to detect suspicious connection which deviates from the developed normal traffic model [10]. Statistical Packet Anomaly Detection Engine (SPADE) observes ports and addresses to monitor detection [11].

C. Yin et al. developed new methodology in applying genetic programming to evolve learned rules for network anomaly detection [12]. Their work was focusing on rule learning for network anomaly detection which involve evolving rules learned from the training traffic by using Genetic Programming (GP) [13], and with the evolved rules, differentiation of the attack traffics from the normal traffic will be carried out by the system.

M.V. Mahoney and P.K. Chan built their IDS model that learns the normal range of values for 33 fields of the Ethernet, IP, TCP, UDP and ICMP protocols using a generic statistical model for all values in the packet headers for all protocols by estimating probabilities based on the time since the last event [14]. Our experiment in essence is to expand the idea of using just the packet header field values to learn the anomalous behavior of the packets during transmission in any TCP/IP network traffic. We extend the statistical analysis by modeling the detection algorithm based on three specific network and transport layer protocols namely UDP, TCP and ICMP.

III. PROTOCOL BASED PACKET HEADER ANOMALY DETECTION (PbPHAD) STATISTICAL MODEL

A. Data Source

The 1999 DARPA Intrusion Detection Evaluation Data Set [15] has been chosen for this research for its data source. This data set was prepared by MIT Lincoln Lab and is publicly available to all researchers. It has been accepted by IDS research community as the *de facto* standard for benchmarking their IDS models.

Fig. 1 [16] shows of an isolated test bed network for the offline evaluation. Scripting techniques were used to generate live background traffic which is similar to traffic that flows between the inside of one fictional Eyrie Air force base created for the evaluation to the outside internet. Rich background traffic was generated in the test bed which looks as if it were initiated by hundreds of users on thousands of hosts. Automated attacks were launched against the UNIX victim machines and the router from outside hosts. Machines labeled 'sniffer' in Fig. 1 run a program named tcpdump [17] to capture all packets transmitted over the attached network segment.

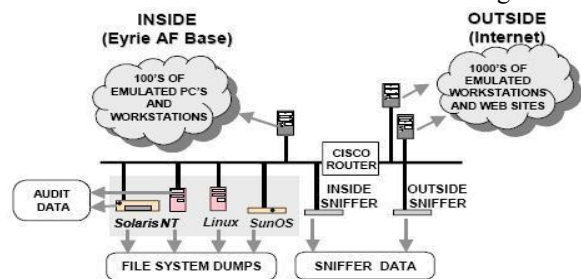


Fig. 1 Block diagram of 1999 test bed

Lincoln Lab provided 5 week of data which consists of 3 weeks of training data and 2 weeks of testing data in several formats such as tcpdump, BSM solaris host audit data and NT audit data. In this research, the tcpdump format will be used as it provides details of the TCP/IP packet that traverse through the network which contains most the information of our interest for detail analysis of the intrusion. In the training data, the first and third weeks of the data do not contain any attacks which are provided to facilitate the training of anomaly based IDS. Only the second week of the training data contains

labeled attacks.

The testing data consist of two weeks of network based attacks in the midst of normal background data. The fourth and fifth weeks of data are the "Test Data" used in the 1999 Evaluation from 29 March 1999 to 9 April 1999. There are 201 instances of about 56 types of attacks distributed throughout these two weeks. Out of 201 attack instances only 176 are found in the inside testing data used for this experiment. Our performance evaluation will be based on the 176 attack instances as we only use the inside testing data. These attacks fall into four main categories:

- Denial of Service (DoS): In this type of attack an attacker makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Examples are Apache2, Back, Land, Mailbomb, SYN Flood, Ping of death, Process table, Smurf, Teardrop.
- Remote to User (R2L): In this type of attack an attacker who does not have an account on a remote machine sends packets to that machine over a network and exploits some vulnerability to gain local access as a user of that machine. Examples are Dictionary, Ftp\_write, Guest, Imap, Named, Phf, Sendmail, and Xlock.
- User to Root (U2R): In this type of attacks an attacker starts out with access to a normal user account on the system and is able to exploit system vulnerabilities to gain root access to the system. Examples are Eject, Loadmodule, Ps, Xterm, Perl, and Fdformat.
- Probing: In this type of attacks an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits. Examples are Ipsweep, Mscan, Saint, Satan, and Nmap.

### B. Protocol-based Packet Header Anomaly Detector (PbPHAD) Model

The fundamental design concept behind our PbPHAD IDS is to learn the normal packet header attribute values during the attack-free week 3 of inside training data which consist of 12,814,738 traffic packets in order to come up with the normal traffic profile based on distinct packet header field values for each of the host in the network. Two separate normal profiles are created for each host for incoming and outgoing traffic. See process 1.0 in Fig. 2.

The packet header field values are taken from layer 2, 3 and 4 protocols which are the IP, Ethernet, TCP, UDP and ICMP which summed up to 30 fields as depicted in the Field Name column in Table 1. We designed our PbPHAD anomaly statistical model based on 3 specific

protocols which are TCP, UDP and ICMP because of their unique behaviour when communicating among hosts, client and servers depending on the purpose and application used for a particular session. With this in mind, a more accurate statistical model with finer granularity which represents the 3 chosen protocols can be built for detecting the anomalous behaviour of the testing data.

For each protocol, if we index each field as  $i$ ,  $i=1,2,\dots,n$ , the model is built based on the ratio of the normal number of distinct field values in the training data,  $R_i$ , against the total number of packets associated with each protocol,  $N_i$ . The ratio,  $p_i = R_i/N_i$  represents the probability of the network seeing normal field values in a packet. Thus, the probability of anomalies will be  $1 - p_i$  for each corresponding field. Each packet header field containing values not found in the normal profile will be assigned a score of  $1 - p_i$  and will be summed up to give the total value for that particular packet.

$$\text{Score packet} = \sum_{i=1}^n (1 - p_i), \quad i = 1, 2, \dots, n \quad (1)$$

As the value of  $R_i$  varies greatly, we use log ratio in our model. The value of column TCP, UDP and ICMP in Table 1 is calculated based on:

*Relative percentage ratio of  $1 - \log(R_i/N_i)$*   
to give the total probability of 1 for each protocol.

Table 1 shows PbPHAD statistical model for one host with IP address 112.016.112.050 for incoming packets. It is obvious from the PbPHAD model that the bigger the number of anomalous fields ( $R$ ), the smaller the anomaly score will be. The anomaly score of 0.000 shows that particular field is not related to that particular protocol.

TABLE 1  
PbPHAD STATISTICAL MODEL FOR HOST 112.016.112.050 INCOMING PACKETS

Ser	Field Name	R	N	Anomaly Score		
				TCP	UDP	ICMP
1	etherdest	1	1545610	0.053342	0.067305	0.073532
2	etherprotocol	1	1545610	0.053342	0.067305	0.073532
3	ethersize	818	1545610	0.031711	0.040035	0.043739
4	ethersrc	6	1545610	0.047563	0.060019	0.065573
5	icmpchecksum	2	84096	0	0	0.057521
6	icmptypencode	2	84096	0	0	0.057521
7	ipchecksum	1	1545610	0.053342	0.067305	0.073532
8	ipdest	1	1545610	0.053342	0.067305	0.073532
9	ipfragid	65536	1545610	0.017574	0.022213	0.024268
10	ipfragptr	2	1545610	0.051106	0.064486	0.070453
11	ipheaderlength	1	1545610	0.053342	0.067305	0.073532
12	iplength	825	1545610	0.031684	0.040001	0.043702
13	ipprotocol	3	1545610	0.049799	0.062838	0.068652
14	ipsrc	28	1545610	0.042595	0.053756	0.058730
15	iptos	3	1545610	0.049799	0.062838	0.068652
16	ipttl	1	1545610	0.053342	0.067305	0.073532
17	tcpack	384656	1076131	0.010744	0	0
18	tcpchecksum	2	1076131	0.049984	0	0
19	tcpdestport	620	1076131	0.031483	0	0
20	tcpflag	8	1076131	0.045513	0	0
21	tcpheaderlen	3	1076131	0.048676	0	0
22	tcpoption	2	1076131	0.049984	0	0
23	tcpseq	383431	1076131	0.010754	0	0
24	tcpsrport	1553	1076131	0.028522	0	0

25	tcpurgptr	1	1076131	0.052220	0	0
26	tcpwindowsize	912	1076131	0.030238	0	0
27	udpchecksum	2	385383	0	0.058839	0
28	udpdestport	4067	385383	0	0.027867	0
29	udplen	46	385383	0	0.046091	0
30	udpsrcport	3	385383	0	0.057190	0
N	Total	842537			1	1

IV. APPLYING KDD TECHNIQUE IN EXTRACTING EXPERT PRODUCTION RULES

Fig. 2 shows the whole process of modelling our packet header anomaly-based IDS. Process 1.0 is the normal profile building phase as described in the previous section. Process 2.0 is where we simulate the testing data and compare it against its normal profile to get its anomaly score for packets which deviates from its normal profile. For anomalous packets which have surpassed their threshold values, expert production rules will be applied to give classification to the packets whether it falls into normal or attack categories. Applying the expert production rules is done in process 3.0. If the anomalous packets are incorrectly classified i.e. big number of false positives or false negatives, a thorough analysis has to be done to identify the packets into its right classification whether it is normal packets or attack packets with proper categories, which is the process 4.0. Process 5.0 is the gist of our discussion in this paper which is applying KDD technique which utilizes machine learning tools to extract the expert production rules.

After extracting the expert production rules, the rules will be updated in the database which is used in Process 3.0 to classify the anomalous packets. The whole process starting from process 1.0 to 5.0 is the normal life-cycle process of IDS modelling for any anomaly based IDS as the data is always dynamic. i.e. after some period of time, when users changed their behaviours in using the network or some new services are introduced into the network, the normal profiles have to be updated and also it is an eminent fact that any network that is connected to the internet is bound to encounter new attacks as new attacks are being developed on a daily basis, therefore process 1.0, 4.0 and 5.0 shall always be an ongoing process as and when it is deemed necessary.

A. Data engineering process

One of the most time consuming process in applying KDD technique to a set of data to learn the association rules of the attributes and coming up with the classification algorithm is the data preparation stage. This is the stage where a set of attributes need to be intelligently chosen and the data is cleansed before the machine learning technique is applied to discover useful knowledge from the data that is being mined. Most of the time, a new set of transformed attributes or secondary attributes need to be introduced into the data structure to increase a chance of getting better results. Fundamentally, choosing the right attributes require a good understanding of the underlying data to be analyzed by the domain expert in that particular field. In the case

of IDS modelling, it requires at least a profound understanding of the ISO-OSI layers, TCP/IP protocol suite, anatomy of attacks and the IDS architectural design principles as domain knowledge can cut down the search space drastically. I. H. Witten and E. Frank put it as “Knowledge is power: a little goes a long way, and even a small hint can reduce the search space dramatically” [18] This stage is known as “data engineering” process which constitutes “engineering the input data into a form suitable for the learning scheme chosen and engineering the output model to make it more effective”. [18]

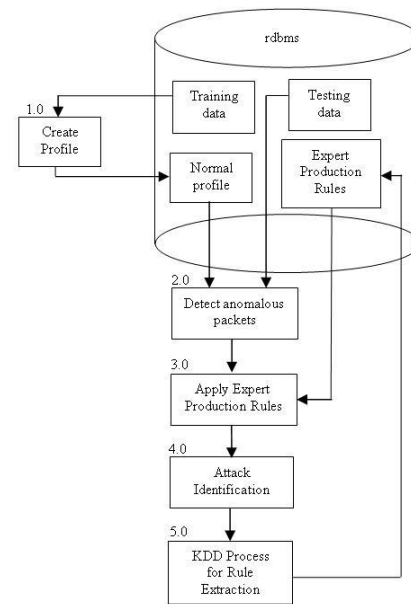


Fig. 2 PbPHAD System Modelling Process

We started modelling the data structure by first selecting the primary fields which is all fields for packet header attributes which comprise of the headers of layer 2, 3 and 4 protocols which are the Ethernet, IP, TCP, UDP and ICMP packet header fields. For each of the packet header field, an anomaly flag field is created for it to indicate the state of that particular field. i.e. whether or not that particular header field value is anomalous which is represented by either ‘1’ or ‘0’ respectively. Not all actual packet header attribute values are included in the data structure. Packet header fields which we thought that would not contribute much to the creation of the rules will be discarded. i.e. the value of IP fragmentation ID is discarded as the value of this 2-byte field is very big and is selected based on how this protocol is implemented by the operating system of the host and does not really tied to any particular protocols. The actual field value of both source and destination IPs are also discarded as our intention is to come up with generic rules which does not get tied to any particular host.

Using 1-second time window, we created 2 secondary attributes which are ‘volume’ – number of bytes destined for a host, measured in byte/s and ‘scan speed’ -

measured in number of packets/s and their corresponding anomaly flag fields as we would foresee that these 2 fields could contribute in the identification of either DoS or Probing attack category. A ‘direction’ field is created to indicate the direction of the packet. i.e. from inside to inside, outside to inside or inside to outside. We would foresee that this field could assist in the rule creation to come up with the right category of attack as we know that R2L and U2R attacks can be identified by this direction.

For transport layer protocol which comprise of TCP and UDP protocols, we introduced two more secondary fields to track the anomaly use of the protocol. As we know that both UDP and TCP use socket-pair to communicate which uniquely identify a connection. i.e. the 4-tuple consisting of the server IP address, server port number, client IP address and client port number. Client port numbers which are known as *ephemeral* port number usually have a value of greater than 1023 and server port numbers which are known as *well-known server* port numbers have a value of less than 1024. [19] If both port numbers in any packet has either value greater than 1023 or less than 1024 this will indicate some anomaly in the protocol being used which might give an indication of a malicious intent. These new secondary fields are named as ‘isbothportsgt1023’ and ‘isbothportslt1024’. For ICMP protocol, we combine the ICMP type and ICMP code fields as for the purpose of identifying an ICMP packet, a unique combination of both fields have to be joined together in order for it to be meaningful.

We also created one field to track if a packet has the same source and destination IP address which obviously shows a grave anomaly for a normal packet. Finally a ‘class’ field and ‘anomaly score’ fields are created to assist the classification of the packets by its anomaly score.

*B. Rule extraction*

Once the data engineering process has finished, we then wrote a program to fill up the values for the secondary fields for all 21,954,377 cleansed packets discovered in the 2 weeks of the testing data to suit the new data structure which has been created. 3 different tables were built for each of the TCP, UDP and ICMP protocols as each one of them has different distinct set of fields to be analyzed by the machine learning tools.

In this exercise we used WEKA [20] for the machine learning workbench. We chose WEKA as it is a very robust open source machine learning workbench which has more than 80 classifier algorithms to choose from. It is quite a challenging task to choose the right algorithm for this purpose as each algorithm has its own strengths and weaknesses which are suitable to particular data structures and furthermore it is very hard to find one algorithm that can out perform all other algorithms for all type of data structures.

We used a small set of data to evaluate the performance of all classifier algorithms that is available in WEKA and after doing a thorough analysis of the results we decided to use J48 Tree classifier algorithm as this algorithm has shown a very good performance for our data set. Furthermore it is very easy to convert the tree to expert production rules which is one of the main components in our IDS model. The ‘Run Information’ of the result will show the structure of the J48 pruned tree and alternatively this tree can be viewed visually using ‘WEKA Classifier Tree Visualizer’ feature. By analyzing the structure of the tree we then convert it to expert production rules. The number of leaves will give the number of rules that can be extracted from the tree. i.e. See Fig. 4.

V. EXPERIMENTAL RESULTS ON THE 1999 DAPRA IDS EVALUATION DATA SET

We tested our model on the 2 weeks of the inside testing data which comprises of 21,954,377 cleansed packets. In this paper, we will discuss the result of one host with IP address 112.016.112.050 which has the most number of attacks among inside hosts in the DARPA 1999 test bed for the duration of the two weeks testing period. Furthermore our IDS model is a host-based model such that the KDD process shall be done by host in order to acquire a meaningful result. We managed to detect 55 out of 61 attack instances which gave us 90.16% success rate as depicted in Table 2 below. Our PbPHAD IDS model shows a very good detection rate for ICMP packets at 100%, a high percentage rate for UDP packets at 90.91% and a slightly lower detection rate for TCP at 89.13%.

TABLE 2  
DETECTION RESULTS FOR HOST 112.016.112.050

Date	Total Attacks			Attacks Detected			False Positive		
	TCP	UDP	ICMP	TCP	UDP	ICMP	TCP	UDP	ICMP
29-Mar-99	4	NA	1	4	NA	1	6	NA	2
31-Mar-99	6	1	NA	5	1	NA	6	1	NA
01-Apr-99	3	NA	NA	3	NA	NA	0	NA	NA
02-Apr-99	NA	NA	1	NA	NA	1	NA	NA	2
03-Apr-99	1	2	1	1	1	1	0	1	2
04-Apr-99	NA	NA	NA	NA	NA	NA	NA	NA	NA
05-Apr-99	4	1	1	4	1	1	2	0	2
06-Apr-99	9	2	NA	8	2	NA	0	0	NA
07-Apr-99	7	1	NA	7	1	NA	85	0	NA
08-Apr-99	5	NA	NA	3	NA	NA	0	NA	NA
09-Apr-99	7	4	NA	6	4	NA	1	0	NA
	46	11	4	41	10	4	80	2	8
Total	61			55			90		
				89.13%	90.91%	100.00%	9 FP/day		
				90.16%					

A. TCP

Fig. 3 below shows one snap shot of a Run information for host 112.016.112.050 on 9<sup>th</sup> April for TCP packets which used 10-fold-cross-validation test mode for J48 classifier algorithm. Only 3 actual primary attribute values are used in this run which are ‘tcp source port’,

‘tcp destination port’ and ‘tcpflag’. 4 secondary attributes used in this run are ‘volume flag’, ‘direction’, ‘if both ports greater than 1023 flag’, ‘if both ports less than 1024 flag’ and the rest are primary attributes flags. There are 170,259 TCP packets destined for this host on this particular day and we managed to get a very good classification result as shown in the Confusion Matrix below with only 1 false positive and 3 false negatives which gives the percentage of correctly classified instances to 99.9977 %.

```

==== Run information ====
Scheme: weka.classifiers.trees.J48
Relation: 112-150-09apr-I-TCP-R1-
weka.filters.unsupervised.attribute.Remove-R1-4,8,10,12,15,17-
21,24-27,29,34-35,37-39,41-
weka.filters.unsupervised.attribute.Remove-R18
Instances: 170259
Attributes: 18 => tcpsrcport, tcpdestport, tcpflag, volumeanom,
direction, isbothportsgt1023, isbothportslt1024,
ethersizeisanom, iplengthisanom, ipfragdisanom,
ipsrcisanom, tcpsrcportisanom, tcpdestportisanom, tcpseqisanom,
tcpackisanom, tcpwindsizeisanom, score, class
Test mode: 10-fold cross-validation
==== Classifier model (full training set) ====
J48 pruned tree
Number of Leaves : 14
Size of the tree : 27
Time taken to build model: 21.08 seconds
==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances 170255 99.9977 %
Incorrectly Classified Instances 4 0.0023 %
Kappa statistic 0.9997
Mean absolute error 0
Root mean squared error 0.0028
Relative absolute error 0.0483 %
Root relative squared error 2.5892 %
Total Number of Instances 170259
==== Detailed Accuracy By Class ====
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
1 0 1 1 1 1
Normal
1 0 1 1 1 1 dos
0.999 0 1 0.999 0.999 0.999 probe
1 0 1 1 1 1 r2l
1 0 0.999 1 1 1 u2r
1 0 0.909 1 0.952 1 data
==== Confusion Matrix ====
a b c d e f <-- classified as
163991 0 0 0 0 1 | a = Normal
0 100 0 0 0 0 | b = dos
1 0 2623 0 2 0 | c = probe
0 0 0 315 0 0 | d = r2l
0 0 0 0 3216 0 | e = u2r
0 0 0 0 0 10 | f = data
    
```

Fig. 3 Run Information for TCP Packets on 9th April using J48 Tree Classifier Algorithm

Fig. 4. shows the corresponding J48 pruned tree for the run. From this figure we can see that a combination of the actual value of the primary attribute which is ‘tcpdestport’, the primary attribute anomaly flag which is ‘ip source is anomalous flag’ and a secondary attribute

which is ‘direction’ correctly classified R2L attack with 0 false negative. For U2R attack, an additional actual value of primary attribute which is ‘tcp destination port’ correctly classifies its class with 0 false negative. See the Confusion Matrix in Fig. 3.

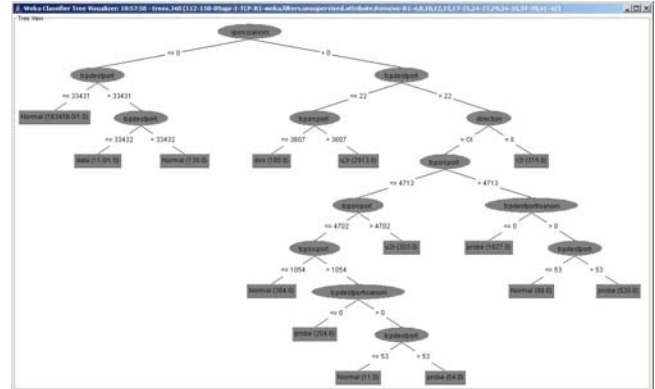


Fig. 4 J48 Tree for TCP Packets on 9th April

```

Rule 1
Antecedent
IF source IP is not anomalous
AND TCP destination port <= 33431
Consequent
THEN class is 'Normal'
...
Rule 14
Antecedent
IF source IP is anomalous
AND TCP destination port > 22
AND direction = 'II'
Consequent
THEN class is 'R2L'
    
```

Fig. 5 J48 Tree for TCP Packets on 9th April

From Fig. 4. we can see that the tree has a size of 27 with 14 leaves. From this tree we can extract the expert production rules. Since there are 14 leaves, 14 rules can be extracted from this tree. Fig. 5 shows some example of the rules extracted from the tree.

**B. UDP**

Fig. 6 below shows one snap shot of a Run information for host 112.016.112.050 on 9th April for UDP packets which used ‘evaluate on training data’ test mode for J48 classifier algorithm. Only 2 actual primary attribute values are used in this run which are ‘udp source port’ and ‘udp destination port’. There are 11,454 UDP packets destined for this host on this particular day and we managed to get a good classification result as shown in the Confusion Matrix below with 0 false positive and 70 false negatives which gives the percentage of correctly classified instances to 99.3889 %.

```

==== Run information ====
Scheme: weka.classifiers.trss.J48
Relation: 112-150-09apr-I-UDP-
weka.filters.unsupervised.attribute.Remove-R1-4,33,35
Instances: 11454
    
```



```

iptosisanom
iplengthisanom, ipfragidisanom, ipfragtrisanom,
ipprotocolisanom
ipsrcisanom, ipdestisanom, icmpptypecodeisanom,
icmpchecksumisanom, class
Test mode: 10-fold cross-validation
=== Summary ===
Correctly Classified Instances    79618    94.6751 %
Incorrectly Classified Instances  4478    5.3249 %
...
=== Confusion Matrix ===
  a    b    c <-- classified as
15108 4428 15 | a = Normal
  6   63153 1 | b = dos
 28    0 1357 | c = probe
    
```

Fig. 10 Run Information for all ICMP Packets using J48 Tree Classifier Algorithm without score

We made another run for ICMP packets to see the effect of taking out the anomaly score to the classification result. Fig. 10 above shows one snapshot of a Run information for host 112.016.112.050 for the whole duration of the two weeks testing period for ICMP packets which used 10-fold-cross-validation test mode for J48 classifier algorithm without anomaly score. The Confusion Matrix shows that the false positive has increased noticeably from only 8 when the ‘anomaly score’ field is included in the run to 4443 which reduced the Correctly Classified Instances percentage by 5.2654% that is from 99.9405% down to 94.6751%. This shows the importance of having a good statistical model to derive the anomaly score rating in reducing the false positives.

VI. COMPARISON WITH 1999 DARPA IDS EVALUATION BEST SYSTEMS RESULTS

We made a comparison between PbPHAD with the combined 1999 DARPA evaluation best systems in each category of attack results on poorly detected attacks as documented by Lippman et al. [16] This analysis was performed to determine how well all 18 evaluated intrusion detection system models submitted by 8 research groups taken together detect attacks regardless of false alarm rates. The best system was first selected for each attack as the system which detects the most instances of that attack which will serve as a rough estimation for upper bound on composite system performance. Our result is in column (g) as shown in Table 3 and 4 below.

Our analysis shows that PbPHAD managed to detect 4 out of 7 attacks as compared to only 4 out of 16 attacks detected by the composite best systems for the poorly detected category. This result shows a performance increment of 32.14%. For Non-detected attack category, PbPHAD managed to detect 4 out of 5 attacks which were not detected by all evaluated systems. This result shows a performance increment of 90.91%.

TABLE 3  
COMPARISON BETWEEN THE 1999 DARPA EVALUATION BEST SYSTEMS AND HOST 112.016.112.050 ON POORLY DETECTED ATTACKS

Ser	Name	Cat.	Total Instances	Instance Detected by Best System	Total Instances for Host 172.016.112.050	Instances Detected by Host 172.016.112.050
(a)	(b)	(c)	(d)	(e)	(f)	(g)
1	portsweep	Probe	13	3	4	3
2	tepreset	DoS	3	1	3	1
<b>Total</b>			<b>16</b>	<b>4</b>	<b>7</b>	<b>4</b>
<b>Percentage Detected</b>				<b>25.00%</b>	<b>57.14%</b>	
<b>Increment</b>						<b>32.14%</b>

Our IDS model failed to detect only one attack which was also not detected by the best composite system which is *resetscan*. The *resetscan* attack is a hard to detect attack as it did not have any anomaly that can be detected by our IDS and it is made up of only 2 out of 110537 incoming packets destined for host 112.016.112.050 for that particular day.

TABLE 4  
COMPARISON BETWEEN THE 1999 DARPA EVALUATION BEST SYSTEMS AND HOST 112.016.112.050 ON POORLY NON-DETECTED ATTACKS

Ser	Name	Cat.	Total Instances	Instance Detected by Best System	Total Instances for Host 172.016.112.050	Instances Detected by Host 172.016.112.050
(a)	(b)	(c)	(d)	(e)	(f)	(g)
1	ipsweep	Probe	7	0	3	3
2	queso	Probe	4	0	1	1
3	resetscan *	Probe	1	0	1	0
4	selfping	DoS	3	0	3	3
5	warezclient	DoS	3	0	3	3
<b>Total</b>			<b>18</b>	<b>0</b>	<b>11</b>	<b>10</b>
<b>Percentage Detected</b>				<b>0.00%</b>	<b>90.91%</b>	

TABLE 5  
OVERALL PERFORMANCE OF PbPHAD MODEL

predicted \ actual	Normal	Probe	DOS	U2R	R2L	Total Instances	% correct	FP	FN
Normal	981,667	77	9	4	-	981,756	100.0%	90	
Probe	179	9,913	-	2	-	10,094	98.2%		181
DOS	33	-	405,105	-	-	405,138	100.0%		33
U2R	256	-	171	13,506	-	13,933	96.9%		427
R2L	-	-	-	-	3,476	3,476	100.0%		-
Total	982,135	9,990	405,285	13,512	3,476	1,414,398			641
% correct	100.0%	99.2%	99.9%	100.0%	100.0%			0.01%	0.05%

VII. CONCLUSION

Our research has clearly shown the benefit of using KDD process in Modelling IDS. Table 5 exhibits the overall performance of PbPHAD model after KDD technique is applied to it as part of its complete life cycle IDS Modelling process. It is apparent that from 1,414,398 packets processed using KDD technique the results has shown actual high percentage of correct classification for Normal, U2R and R2L at 100%, DOS at 99.9% and the smallest percentage is for Probe at 99.2%.

The result shows that we managed to suppress the percentage of false positive rate to be very small at 0.01% which can be seen from Table 2 that it is only at 9 FP/day.

This research also shows the importance of having a good statistical model that can give good anomaly score rating to an anomalous packet. We have demonstrated how the score plays an important role in classifying the packets into their proper classes as shown in Fig. 10. From this research, we also show that one of the keys to having good classification results is to have secondary attributes intelligently chosen for the data structure which would greatly assist the classifier algorithm to yield beneficial knowledge from the data being mined which eventually would produce the corresponding good expert production rules needed in the IDS model. Comparison of PbPHAD with the 1999 DARPA composite system performance attested that our model has succeeded in discovering new dimension of attack space which complements the composite systems in terms of covering the whole dimension of attack space.

#### REFERENCES

- [1] John McHugh, A.C., and Julia Allen, *Defending Yourself: The Role of Intrusion Detection Systems*. IEEE Software, 2000(September/October): p. 42-51.
- [2] Mukkamala, S., A.H. Sung, and A. Abraham, *Intrusion detection using an ensemble of intelligent paradigms*. Journal of Network and Computer Applications, 2005. 28(2): p. 167-182.
- [3] Schultz, E.E. and E. Ray, *The future of intrusion prevention*. Computer Fraud & Security, 2007. 2007(8): p. 11-13.
- [4] Patcha, A. and J.-M. Park, *An overview of anomaly detection techniques: Existing solutions and latest technological trends*. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2007. Volume 51(Issue 12 (August 2007)): p. 3448-3470.
- [5] Negnevitsky, M., *Artificial Intelligence - A Guide to Intelligent Systems*. Addison Wesley, 2002.
- [6] S. Peddabachigari, A. Abraham, C. Grosan, C. Grosan, and J. Thomas, *Modeling Intrusion Detection System Using Hybrid Intelligent Systems*. Journal of Network and Computer Applications, Elsevier Science, 2005.
- [7] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, P. Neumann, H. Javitz, A. Valdes, and T. Garvey, *A Real-time Intrusion Detection Expert System (IDES)*. Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, February 1992.
- [8] D. E. Denning, *An Intrusion Detection Model*. In IEEE Transactions on Software Engineering, February 1997: p. 222-228.
- [9] J. P. Anderson, *Computer Security Threat Monitoring and Surveillance*. Technical report, James P Anderson Co., Fort Washington, Pennsylvania, April 1980.
- [10] D. Barbara, J. Couto, S. Jajodia, L. Popyack, and N. Wu, *ADAM: Detecting intrusions by data mining*. In Proc. of the IEEE Workshop on Information Assurance and Security, June, 2001.
- [11] S. Biles, *Detecting the Unknown with Snort and the Statistical Packet Anomaly Detection Engine ( SPADE )*. Technical Report TR2004-485, Department of Computer Science, Dartmouth College, Hanover, USA, 2003.
- [12] C. Yin, S. Tian, H. Huang and J. He, *Applying Genetic Programming to Evolve Learned Rules for Network Anomaly Detection*. In Advances in Natural Computation, First International Conference, ICNC 2005, Proceedings, Part III, 2005. 3612: p. 323-331.
- [13] J.R. Koza, *Genetic Programming*. MIT Press, 1992.
- [14] M. V. Mahoney, and P. K. Chan, *PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic*. Technical report, Florida Tech., technical report CS-2001-4, April 2001.
- [15] MIT, *Lincoln Laboratory 1999 DARPA Intrusion Detection Data Sets*. <[http://www.ll.mit.edu/IST/ideval/data/1999/1999\\_data\\_index.html](http://www.ll.mit.edu/IST/ideval/data/1999/1999_data_index.html)>.
- [16] R. P. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, *The 1999 DARPA Off-Line Intrusion Detection Evaluation*. MIT Lincoln Lab Technical Report, 2000.
- [17] *tcpdump*, Lawrence Berkeley National Laboratory Network Research Group <<http://www.nrg.ee.lbl.gov>>.
- [18] I. H. Witten, and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
- [19] Stevens, W.R., *TCP/IP Illustrated Volume 1*. Addison Wesley, 2003.
- [20] Weka, Software. *Machine Learning*. The University of Waikato, Hamilton, New Zealand. Available form: <<http://www.cs.waikato.ac.nz/ml/weka/>>.

**Solahuiddin B. Shamsuddin** graduated with a B Sc in Electrical Engineering from Wichita State University, Kansas, USA in 1986 and received Post Graduate Diploma in Systems Analysis and Design from University Technology MARA, Malaysia in 1991. He is now studying for his PhD degree in Network Security at the University of Bradford, United Kingdom.

In 2002 he joined the Malaysian National ICT Security and Emergency Response Centre (NISER) and became the Manager for the Malaysian Computer Emergency Response Team (MyCERT) in 2003. He served in the Royal Signal Corps of the Malaysian Army for 10 years prior to this.

**Professor Mike Woodward** graduated with a first class honours degree in Electronic and Electrical Engineering from the University of Nottingham in 1967 and received a PhD degree from the same institution in 1971 for research into the decomposition of sequential logic systems.

In 1970 he joined the staff of the Department of Electronic and Electrical Engineering at Loughborough University as a lecturer, being promoted to Senior Lecturer in 1980 and Reader in Stochastic Modelling in 1995. He remained at Loughborough until 1998 when he was appointed to the Chair in Telecommunications at the University of Bradford, where he also became the Director of the Telecommunications Research Centre. He served as the Head of the Department of Computing at the University of Bradford from 2002 to 2006. His current research interests include queueing networks, Internet congestion control, quality of service routing and mobile communications systems and he is the author of two books and over one hundred research papers on the above and related topics. He is currently Head of the Networks and Telecommunications Research Group and is supervisor to twelve full time research students.

Professor Woodward is a Fellow of the Institute of Mathematics and its Applications (FIMA) and is a Chartered Mathematician (CMath), Chartered Scientist (CSci) and a Chartered Engineer (CEng).