A hybrid data mining anomaly detection technique in ad hoc networks

Yu Liu,* Yang Li and Hong Man

Department of Electrical and Computer Engineering,
Castle point on Hudson,
Hoboken, NJ 07030, USA
E-mail: yliu@stevens.edu
E-mail: yli1@stevens.edu E-mail: hman@stevens.edu
*Corresponding author

Wei Jiang

Department of Systems Engineering and Engineering Management,
Stevens Institute of Technology,
Castle point on Hudson,
Hoboken, NJ 07030, USA
E-mail: wjiang@stevens.edu

Abstract: Ad hoc network security mainly relies on defence mechanisms at each mobile node due to lack of infrastructure. For this reason, various intrusion detection techniques have been proposed for ad hoc networks. Developing Intrusion Detection Systems (IDS) for individual nodes in ad hoc network is challenging for a number of reasons, including resource constraints at each node and the difficulties to locate attack source for prompt response. In this paper, we propose a hybrid data mining anomaly detection technique for node-based IDS. Specifically, we incorporate two data mining techniques, that is, association-rule mining and cross-feature mining, to characterise normal behaviours of mobile nodes and detect anomalies by finding deviance from the norm. The advantage of our hybrid approach is that association-rule mining and cross-feature mining usually complement each other in time scale and sensitivity to different attack types. We investigate features of interest from both the medium access (MAC) layer and the network layer. Our intention of using the MAC layer features is to localise the attack source within one-hop perimeter. To preserve the precious energy of mobile nodes, we propose two compact feature sets, that is, direct feature set and statistical feature set, that target on short-term and long-term profiling of normal node behaviours, respectively. Considering the characteristic of audit data collected upon different feature sets, we apply association-rule mining to the short-term profiling and cross-feature mining to the long-term profiling. We validate our work through ns-2 simulation experiments. Experimental results show the effectiveness of our method.

Keywords: anomaly detection; ad hoc network; association rule; cross-feature; data mining; Bayesian network.

Reference to this paper should be made as follows: Liu, Y., Li, Y., Man, H. and Jiang, W. (2007) 'A hybrid data mining anomaly detection technique in ad hoc networks', *Int. J. Wireless and Mobile Computing*, Vol. 2, No. 1, pp.37–46.

Biographical notes: Yu Liu is a PhD candidate in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. Her main research interests are network vulnerability assessment, intrusion detection, game theory for wireless networks and peer-to-peer network security.

Yang Li is a PhD candidate in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. His main research interests include medium access control and routing protocol design in wireless ad hoc networks and quality of service and differential service for wireless network.

Hong Man is an Assistant Professor in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. He received his PhD from Georgia Institute of Technology in 1999. His current research interests include image analysis, medical imaging and multimedia networking.

Wei Jiang joined Stevens Institute of Technology as an Assistant Professor in 2003. Before that, he worked for four years at AT&T labs. He received his PhD from Hong Kong University of Science and Technology in 2000. His current research interests are statistical methods for quality control, data mining and enterprise intelligence.

1 Introduction

Wireless ad hoc networks are formed dynamically by an autonomous system of mobile nodes connected through wireless links without the aid of any infrastructure or centralised administration. While the self-organising nature of ad hoc networks provides convenient communication methods for mobile users, they lack perimeter defence mechanisms, which creates significant challenges to network security. It is even more challenging for an *open environment* ad hoc network where no prior trust relationship exists among network nodes. The network operation relies on altruism and cooperation of neighbour nodes to participate network functions such as forwarding and routing.

In an ad hoc network, security threats generally come in two forms: selfishness and maliciousness. A selfish node may chip in far less energy in relaying packets than anyone else and its behaviour is similar to a packet dropping attack which can result in denial of service to some nodes and can even significantly degrade the overall network performance (Michiardi and Molva, 2002). A malicious node, on the other hand, intentionally aims at harming the network operation by mounting attacks against different network layers to either compromise individual node(s) or disrupt the overall network operation. Moreover, a malicious node behaviour can endanger the security of Wireless Local Area Networks (WLANs) and wired networks if a rogue mobile node can form an ad hoc network with a legitimate WLAN or wired network station.

Attacks on ad hoc networks, similar to WLANs and wired networks, can be categorised into local attacks and network-based attacks. In this work, we focus our attention on detecting network-based attacks. Network-based attacks can be launched against individual layers of the network protocol stack. Considering the self-organised peculiarity of ad hoc networks, various intrusion detection techniques are proposed to combat attacks at the network layer (Deng et al., 2003; Hu et al., 2003; Huang and Lee, 2003; Kachirski and Guha, 2002; Tseng et al., 2003). Nonetheless, intrusion detections at the network or higher layers usually capture end-to-end attack behaviour and expose attack source via IP address. Because of the infrastructureless and dynamic nature in ad hoc networks, locating attack source is essential for initiating fast response. On the other hand, resource constraints limit the feasibility of applying existing IP traceback techniques to locate attack source efficiently (Thing and Lee, 2004). Consequently, most existing intrusion detection techniques have not been able to emphasise the localisation of attack source.

In this paper, we propose a node-based IDS that mostly uses the MAC layer traffic data for intrusion detection. In contrast to the network or higher layers, the MAC layer is close to the bottom of the network protocol stack, so most network-based attacks will have a direct impact on its operation. Thus, using the MAC layer traffic data for intrusion detection will be sensitive to large varieties of network attacks and will incur less detection delay. MAC layer address is considered as the identity of attack source. This facilitates our Intrusion Detection Systems (IDS) to localise attack source(s) within one-hop perimeter, which will clearly make any further location discovery more accurate.

In terms of intrusion detection approach, we opt for using anomaly detection techniques, because signature detection techniques may be impractical for ad hoc networks due to the difficulties of specifying, distributing and updating signatures of attacks. However, anomaly detection techniques are usually prone to high false positive rates. Therefore, a well-defined feature set becomes an extremely important aspect to these techniques. On the other hand, in ad hoc networks, resource constraints (i.e. computational power, storage capacity and battery life) of mobile nodes prevent the use of a rich and diversified feature set to profile normal node behaviours.

Our approach is to specify two compact feature sets, a direct feature set and a statistical feature set, that aim at profiling short-term and long-term node behaviours, respectively. Accordingly, we propose a new hybrid data mining anomaly detection technique that utilises association-rule mining over the audit data collected upon the direct feature set (i.e. short-term profiling) and utilises cross-feature mining over the audit data collected upon the statistical feature set (i.e. long-term profiling). The advantage of our hybrid approach is that cross-feature mining is able to capture both frequent and infrequent event patterns, which complement association-rule mining in which only frequent-enough event patterns can be captured. Moreover, because of the use of statistical features, the size of audit data used in cross-feature mining is considerably smaller than the one used in association-rule mining. Therefore, cross-feature mining is suitable for long-term always-on profiling and can preserve the precious node energy.

The main contributions of our work are as follows.

1 We proposed a new hybrid data mining anomaly detection technique that incorporates association-rule mining and cross-feature mining techniques.

2 We investigated two compact feature sets which are defined over direct and statistical measures of the network data, respectively and targeted at short-term and long-term node behaviour profiling correspondingly.

3 We are able to localise an attack source within a one-hop perimeter using the proposed direct feature set which is able to incorporate cross-layer intelligence from both the MAC layer and the network layer.

4 We present a novel collaborative detection scheme by using Bayesian network to correlate local and global alerts and make collaborative decision accordingly. In addition, we also propose using Bayesian network to evaluate multiple attack sources presented in multiple anomalies to reduce false positive rate.

5 We implemented the proposed algorithm and performed a comprehensive simulation on the ns-2 simulator (Broch et al., 1998).

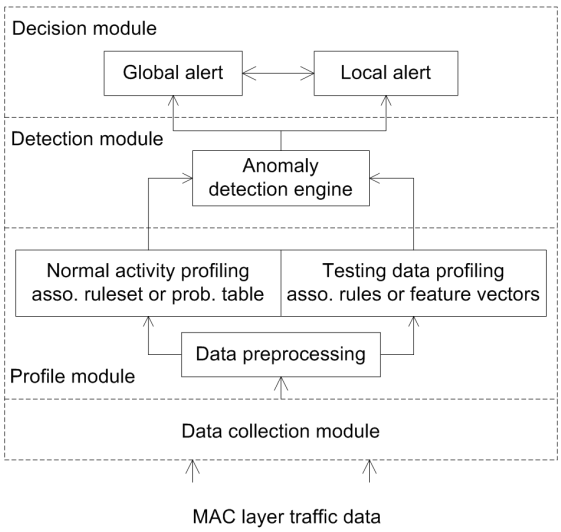
The organisation of this paper is the following. In Section 2, we present the model architecture of the proposed IDS. In Section 3, we discuss feature of interest from the MAC layer data. In Section 4, we describe the anomaly detection technique, and illustrate our collaborative detection scheme.

In Section 5, we provide the simulation results and the performance evaluation. We conclude this paper in Section 6.

2 The model architecture

Because of the lack of centralised administration, we propose a node-based IDS that relies solely on local information collected within its radio transmission range. The conceptual model architecture of the proposed IDS is illustrated in Figure 1. It comprises the following four major components: data collection module, profile module detection module and decision module.

Figure 1 The proposed model architecture



- *Data collection module*: this module collects the audit data according to the proposed two feature sets within its neighbourhood.
- *Profile module*: this module has two subsystems.
  - Preprocessor transforms audit data to either 'market basket' format (Agrawal and Srikant, 1994) for the succeeding association-rule mining process or "quantised feature vector format" for the succeeding cross-feature mining process.
  - Profiler uses Apriori algorithm (Agrawal and Srikant, 1994) to find association patterns (rules) from feature values collected from the direct feature set and uses cross-feature mining algorithm (Huang and Lee, 2003) to find inter-correlations among feature values collected from the statistical feature set. Short-term normal behaviour profiling refers to the association-rule analysis and a normal profile consists of an aggregated association rule set of multiple training data segments, where each rule is associated with four metrics: $minimum$ support and confidence levels and $maximum$ support and confidence levels. On the other hand, long-term normal behaviour profiling refers to the cross-feature analysis and a normal profile consists of probability distributions of feature vectors. A decision threshold θ is chosen systemically according to these inter-correlation probability distributions and is used to predict the normality of feature vectors constructed from testing data.
- *Detection module*: anomaly detection is to detect deviance from the norm. In this module, test data profiles are compared with the expected normal profiles. Specifically, in the association-rule analysis, any new rule or rule with deviations beyond the corresponding threshold interval $[minimum - \epsilon, maximum + \epsilon]$ is considered as an anomaly rule. In the cross-feature analysis, any new feature vector or feature vector with inter-correlation probability that is less than the decision threshold θ is considered as an abnormal feature vector.
- *Decision module*: in the association-rule analysis, any anomaly rule can trigger a local alert when the support and confidence level is high. The detecting node can then send a global alert to warn its neighbours. When the support and confidence level is low, the detecting node can make collaborative decision by gathering intelligence (global alerts) from its neighbours.
  In practice, multiple anomaly rules may be generated from audit data. Some rules come from the same attack source and the others come from different attack sources. For the former scenario, we raise alert once using the rule with the highest support level. For the latter scenario, we use Bayesian network to evaluate multiple attack sources and raise an alert according to the largest posterior probability of the attack sources.
  In cross-feature analysis, a single abnormal feature vector can be used to trigger a local alert. An alternative is to test the normality of a data interval (i.e. a relative long-term period). For each testing data interval, if the total number of abnormal feature vectors is above a threshold τ, the testing interval is considered as abnormal and consequently trigger a local alert.

3 Feature of interest

A variety of intrinsic features are obtainable from the MAC layer data. However, there is a tradeoff between effectiveness and efficiency in choosing a feature set for intrusion detection. A rich and diversified feature set can help the IDS to detect various type of attacks, but it also means more resource will be consumed for detection process. Taking into account the scarcity of node energy in ad hoc networks, we opt to a compact feature set.

Selecting features of interest for intrusion detection is specific to the underlying network protocols. Hereafter, we assume the most commonly considered 802.11 MAC protocol, which uses a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme. A 4-way RTS/CTS/DATA/ACK handshake exchange is performed for every data packet transmission except for broadcast packets.

Intuitively, features with large value variations cannot be directly used to profile normal node behaviours. On the other hand, some of these values are indeed relevant to node normal behaviour. For instance, the *duration* values in the RTS/CTS frames indicate the communication channel utilisation and

a malicious node behaviour could be unfairly occupy the communication channel by launching network-based attacks. To cope with this problem, we here propose two feature sets, direct feature set and statistical feature set, where the former extracts information directly from the header fields of the MAC control and data frames and the latter gleans statistical measures of the network traffic data.

3.1 Direct feature set

In view of the 802.11 MAC protocol, there are three types of MAC frames: management frames, control frames and data frames. Management frames cope with authentication and association. Since IDS is a passive defense mechanism, we will not use any information from them. In the control frames (RTS, CTS and ACK frame), both RTS and CTS carry the proposed *duration* of the DATA frame, which is then used by neighbour nodes to set the duration in the Network Allocation Vector (NAV) resided at each node. We leave the characterisation of the NAV values in the statistical feature set and only incorporate the type of control frames in the direct feature set. Considering a MAC DATA frame, because either a routing data packet (RtDataPkt) or a routing control packet (RtCtrlPkt, which includes Route Request, Route Reply and Route Error packet) is encapsulated, we tag the type of the network layer packet type (RtPktType) to each MAC DATA frame. The proposed direct feature set and its value space is illustrated in Table 1.

Table 1 Direct feature set

Feature	*Value space*
Flow direction (Dir)	SEND, RECV, DROP
Send address (SA)	$sa_i, \forall i \in$ node set S
Destination address (DA)	$da_j, \forall j \in$ node set S
MACFrameType	RTS, CTS, DATA, ACK
RtPktType*	RtDataPkt, RtCtrlPkt

*This feature applies to MAC DATA frame only.

Note that the proposed direct feature set is actually a cross-layer feature set. That is, we incorporate features from both the MAC layer and the network layer by tagging network layer packet type to each MAC DATA frame.

The advantage of a cross-layer feature set is threefold. Firstly, it allows each mobile node to monitor both the MAC and network layer simultaneously and extends the detection capabilities to both the MAC layer and the network layer. In fact, most of the network layer attacks will cause immediate affects at the MAC layer and detection at the MAC layer can be more direct and more prompt. Secondly, it enables our IDS to detect attacks that are unable or hard to detect by using single layer information. For example, it's very hard to detect our simulated blackhole attack (described in Section 5.2) using information from the network layer only. This is because after the attack source broadcasts the falsified routing information, all neighbour nodes update their routing table accordingly. From a neighbour node point of view, this routing change is quite normal. Thirdly, it can be used to localise attack source within one-hop perimeter due to the use of MAC address as a feature for detecting attack and hence further location discovery will be more accurate and incurs less delay than using IP address to identify attack source.

3.2 Statistical feature set

The statistical feature set is defined over a set of statistical measurements gleaned from the MAC traffic data over a fixed sampling interval, for example, 5 s. Table 2 illustrates the proposed statistical feature set and its value space. Here are the reasons we select these features:

- *NAV value*: NAV is a counter residing at each node that represents the amount of time that the channel will be occupied by the current sending node. Briefly, NAV values indicate the busy/idle status of the communication channel.
- *Transmission traffic rate (xmitTrafficRate) and reception traffic rate (recvTrafficRate)*: these two values refer to the total number of inbound and outbound traffic data (bytes) per sampling interval. The traffic data includes the RTS/CTS/DATA/ACK frames. The two values are node specific and are also relevant to the node density and traffic pattern in the neighbourhood.
- *RTS retransmission count (reXmitRTS) and DATA retransmission count (reXmitDATA)*: these two values can be considered as channel congestion indicators. Excessive number of retransmission of RTS and DATA frames may be caused by network-based attacks such as flooding attack.
- *Active neighbour node count (nigbrNodeCount)*: this value specifies the number of active neighbour nodes of the monitoring node. Here active means that a node has data transmission activities.
- *Forwarding node count (fwdNodeCount)*: this value specifies the number of active forwarding nodes of the monitoring node.

Table 2 Statistical feature set

Feature	*Value space*	*Unit*
Time	Ignored in classification	second
NAV	Continuous	second
xmitTrafficRate	Continuous	byte
recvTrafficRate	Continuous	byte
reXmitRTS	Discrete	count
reXmitDATA	Discrete	count
nigbrNodeCount	Discrete	count
fwdNodeCount	Discrete	count

Taking into account the different data characteristic in the audit data collected upon the aforesaid two different feature sets, we apply two different data mining techniques to profile normal behaviours of mobile nodes and detect anomalies by finding deviations from the expected normal profiles.

4 Anomaly detection

We assume an ad hoc network is in an open environment where no trusted authority existed among the network nodes. Besides the use of 802.11 MAC protocol, we consider one of the on-demand routing protocols, Ad hoc on Demand Distance Vector (AODV), in the network layer. In addition, we assume there is no explicit congestion control mechanism in the transport layer.

4.1 Association-rule analysis

Association rule describes associations of features (attributes) within transaction records of an audit dataset. Let $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$ denote a set of n transaction records and $\mathbf{F} = \{F^1, F^2, \ldots, F^k\}$ denote a set of k features defined over $\mathbf{T}$. A feature vector, denoted as $\mathbf{f} = \{f_1, f_2, \ldots, f_k\}$, is a collection of k-tuple items whose value assignments corresponding to the k features in $\mathbf{F}$, that is, f_k represents a value from the feature F^k. A transaction record T_i comprises a timestamp value and a feature vector $\mathbf{f}$. Let A and B denote two disjointed item subset in T_i. Let the support of A, denoted by $\sup(A)$, represent the percentage of transactions containing A in T and the support of both A and B, denoted by $\sup(A \cup B)$. An association rule is (Agrawal and Srikant, 1994)

$$A \rightarrow B, (s, c) \tag{1}$$

where $s = \sup(A \cup B)$ is the support value of the rule and $c = \sup(A \cup B)/\sup(A)$ is the confidence. The rule holds if $s \geq$ minsup and $c \geq$ mincon f, where minsup and mincon f denote predefined minimum support and confidence threshold, respectively.

According to the direct feature set given in Table 1, a transaction record is an instantiation of the following feature set:

$$< \mathit{Timestamp}, \mathit{Dir}, \mathit{SA}, \mathit{DA}, \mathit{MACFrameType}, \mathit{RtPktType} >$$

An example association rule is

$$(\text{sa7}, \text{RtDataPkt} \rightarrow \text{da16}, \text{RECV}), (0.2, 1)$$

which describes an event pattern related to the RECV flows of a monitoring node (a neighbour node of both node 7 and 16). It says that 20% of transaction records matches the event pattern of 'node 7 sends data packet to node 16' and the data packets received by node 16 are 100% of the time from node 7.

In our work, we use Apriori algorithm to find association patterns from a set of transaction records. Apriori algorithm is a primitive and successful algorithm to find association patterns from a large database. However, the algorithm usually generate a large number of rules which is prone to false alarms. We first discard rules that do not contain any node identity (sa_i or da_j). In addition, we propose to use Maximal Frequent Itemsets (MFI) (Burdich et al., 2001) criteria to further prune rules that are formed with redundant frequent itemsets. A MFI is defined as a frequent itemset for which none of its immediate supersets are frequent. This pruning process dramatically reduces the size of normal rule-based profile, yet still captures the frequent association patterns from a dataset. In our experiments, the MFI pruning can reduce the number of association rules by 20 to 40%.

The size of audit data set collected from the MAC layer data for association-rule analysis is usually large even for a short time interval, say 50 s. This is because of the massive number of packet-level transactions in the MAC layer. Thus, building a normal profile with association rules extracted from long time intervals puts pressure on the precious resources of mobile nodes. For this reason, we only use association rules to build short-term normal profiles. A training dataset consists of multiple data segments of short time intervals. Association rules are extracted from each data segment and then are aggregated into a rule set which is considered as the normal profile. In the aggregation process, each association rule is recorded with $minimum$ and $maximum$ support and confidence levels.

Testing data (real-time activities) segments are collected over the same time interval as used in the above single training data segment. Association rules generated from the testing data segments are compared with the expected normal profiles. Any new rule or rule with deviations beyond the corresponding threshold interval $[minimum - \epsilon,\ maximum + \epsilon]$ is considered as an anomaly rule.

4.2 Cross-feature analysis

The cross-feature analysis technique was first presented by Huang et al. (2003). The basic idea is to explore inter-feature correlations from training dataset and use these correlations to build normal profiles.

As defined in the preceding subsection, for a data set $\mathbf{T}$ of n transaction records and a set $\mathbf{F}$ of k features defined over $\mathbf{T}$. A feature vector $\mathbf{f} = \{f_1, f_2, \ldots, f_k\}$ is an instantiation of k-tuple features in $\mathbf{F}$ and a transaction record T_i comprises a timestamp value and a feature vector $\mathbf{f}$. In the cross-feature analysis, timestamp values are ignored and feature vectors are used to construct normal profiles. The total number of feature vectors over $\mathbf{F}$ is $\prod_k(|F_k|)$. Note that only partial feature vectors may be instantiated from a specific training data set when constructing a normal profile. Cross-feature analysis involves the following steps:

1 For each feature vector $\mathbf{f}$ in a training data set, compute a classifier C_i for each f_i using $\{f_1, f_2, \ldots, f_{i-1}, f_{i+1}, \ldots, f_k\}$. C_i in terms of $p_i(f_i|f_1, f_2, \ldots, f_{i-1}, f_{i+1}, \ldots, f_k)$ is learned from the training data set. Specifically, given a classification tree produced from C4.5, suppose that n is the total number of examples in a leaf node, n_i is the total number of examples with class label f_i in the same leaf, then $p(f_i|x) = n_i/n$ is the probability that x is an instance of class f_i, that is, $x \in \mathbf{f} \setminus f_i$.

2 Compute the average probability $p = \sum_i p_i/k$ for each feature vector $\mathbf{f}$ and save it in a probability distribution matrix Δ. The probability p represents a inter-correlation level among feature values in $\mathbf{f}$. A 'normal' feature vector should have relative larger p value than an 'abnormal' feature vector, since all features in $\mathbf{f}$ make contributions to p. Thus, p values are

used to predict the normality of feature vectors by choosing a decision threshold θ. For any feature vector in Δ, if its p value is less than θ, it is considered as a training error and may result in false positive alarm (testing error) in the testing phase. A normal profile is built with all feature vectors in Δ with their p values greater than θ.

3 Given a testing data set, compare each feature vector $\mathbf{v} = \{v_1, v_2, \ldots, v_k\}$ with the expected normal profile built in the preceding step. Any new feature vector or feature vector with $p < \theta$ is considered as an abnormal feature vector.

Cross-feature analysis is applied to the audit data upon the statistical feature set specified in Table 2. We note that some features have continuous value space. As stated in the previous step 1, every feature value in a feature vector is used as a classifier during cross-feature analysis. Therefore, all continuous and discrete feature with infinite value space need to be discretised and quantised into a finite number of bins. The number of bins allocated to each feature directly impact the accuracy of classification model. The simplest quantisation method is to divide the range of the attribute values into a fixed number of equal-width bins. However, because the characteristic and probability distributions of each feature value space are inherently different, one should apply a non-uniform quantisation to each feature value space. A heuristic approach is to randomly select several training data segments and plot histogram to roughly estimate the quantisation bin intervals. An example of quantisation bins used in our simulation is given in Section 5.

There are two advantages of cross-feature analysis. Firstly, it is able to capture both frequent and infrequent event patterns. This is because an infrequent event pattern will contribute little to the inter-correlation probability p and small p value will likely be classified as an abnormal vector and on the other hand, a frequent event pattern will contribute more to the probability p and a large p value will likely be classified as a normal vector. So the cross-feature analysis can complement the association-rule analysis for which only frequent-enough event patterns are extracted from both training data and testing data. Secondly, the cross-feature analysis is able to capture (relatively) long-term node behaviour owing to the fact that data is sampled in a certain frequency. For instance, assume feature sample interval is 5 sec, audit data of a 2000 sec time span will only constitute 400 sampling points. Hence the size of audit data collected upon the statistical feature set is dramatically smaller than the one collected upon the direct feature set given the same time interval. Nevertheless, the inclusion of association-rule analysis can help to identify attack source(s).

4.3 Intrusion response

Here intrusion response refers to associating anomalies with alerts. As described in Section 2, both association-rule analysis and cross-feature analysis can trigger local alert if any deviance from the expect normal profiles is detected. Because cross-feature analysis is used to perceive the normality of neighbourhood environment and it cannot be used to identify specific attack source, we decide to only use anomalies from association-rule analysis to trigger global alert. In particular, a detecting node can send a global alert to its neighbours when it detect anomaly rules with high support and confidence levels. On the other hand, when support and confidence levels are low, the detecting node can make collaborative decision by gathering intelligence (global alerts) from its neighbour nodes.

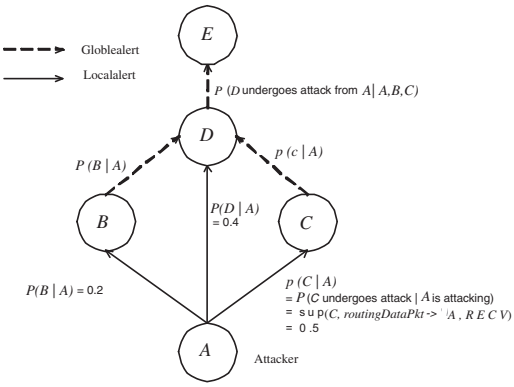
We use Bayesian network to correlate local and global alerts in the decision module and make collaborative decision accordingly. A Bayesian network is defined by a Directed Acyclic Graph (DAG) over nodes representing random variables and arcs signifying conditional dependencies between pairs of nodes. In our model, we define a Bayesian network with a set $\mathbf{X} = \{X_1, \ldots, X_n\}$ of variables that represent a monitoring node and a set of neighbouring nodes. Each variable X_i takes on a binary value, where a true state corresponds to 'being attacked' (for monitoring node) or 'attacking' (for neighbouring node) and a false state corresponds to the opposite. Let S be a network structure that encodes the conditional relationship between variables in $\mathbf{X}$ and $\mathbf{P}$ is a set of local probability distributions associated with each variable. The posterior distribution with respect to S is

$$P(H|O) \propto P(O|H) \times P(H) \tag{2}$$

where H denotes a set of subjective beliefs that we are interested in and $P(H)$ is the set of prior probabilities on S. $O = \{X_1 = x_1, \ldots, X_n = x_n\}$ denotes a set of observations (evidences) on $\mathbf{X}$. $P(O|H)$ is called marginal likelihood of O. Here, subjective beliefs in our model could be one or a set of attacking nodes (that are suspicious or interesting to evaluate).

Here we briefly illustrate collaborative decision making through an example network. As shown in Figure 2, A and E are not in the vicinity of each other. A adjoins three neighbouring nodes B, C and D. E is neighbouring with D. The attacker A stages flooding attack by sending spurious data packets against B, C and D. Suppose D learns an anomaly rule *(D,* RtDataPkt $\rightarrow$ *A,* RECV*),* (0.4, 1) from its local detection module. B and C likewise learn the similar rule with different support values, for example, 0.2 and 0.5, respectively. We associate the support values of these anomaly rules as the marginal probabilities in a Bayesian network, for example, $P(B|A)$, $P(C|A)$ and $P(D|A)$. We call such marginal probability as direct local alert.

Figure 2 Alert aggregation at node D



Within the decision module, each node maintains an intrusion response matrix M. An example matrix at node B is illustrated in Table 3. Each row contains specific local and global threat levels pertaining to an attack source. In particular, the first column records the attacker identification; the second column records direct local alert that computed from its own detection engine; the third column records neighbouring alert(s), each in turn is a pair value of (s, p), where s represents the MAC address of the neighbouring node who sends the global alert, p is the global alert (a probability) from this neighbouring node; finally, the fourth column records the updated local alert value by computing the probability of union set of the local alert and the incoming global alert(s). That is,

$$p(x_i = 1|\mathbf{pa}_i) = 1 - \prod_j (1 - p(x_i = 1|x_j)) \quad (3)$$

where X_i denotes a detecting node and $\mathbf{pa}_i$ represents the set of corresponding parent nodes, $j \in \mathbf{pa}_i$ and $p(x_i = 1|x_j)$ is a local or global alert.

Table 3 Intrusion response matrix at node B in Figure 2

AttackFrom	*DirectLocal*	*NeighbourAlert*	*UpdateLocal*
A	0.2	0	0.2
A	0.2	(D, 0.4)	0.52
A	0.2	(D, 0.4), (C, 0.5)	0.72
...	...	...	

Suppose at node D, its direct local alert for attacker A is 0.4 and it also receives two global alert: $(B, 0.2)$ and $(C, 0.5)$, then D can update its local alert $P(D|A, B, C)$ by Equation (3).

In addition to the four column fields in M, another column field, TotalTimeLive (TTL), is necessary to keep the information in M up-to-date.

In order to avoid massive global alerts, we piggyback a global alert on a CTS packet and the global alert could either be the value in the directLocal column or updateLocal column depending on whether the requesting node (i.e. the sending node of RTS frame) is in the neighbourAlert column. The algorithm is described as in Algorithm 1. Note that the overhead is kept to a minimum because global alert is only sent to neighbours and because of the use of piggyback technique.

Once an attacker carries out an attack, several neighbouring nodes of the attacker may detect anomalies about the attacker. From Algorithm 1, we see that the aggregated intrusion alert becomes more vivid and can quickly converge at all the neighbouring nodes of the attacker. Therefore, the attacker can be ruled out within one-hop perimeter. However, we have not yet deal with the possibility that the attacking node (or its colluding node) may attempt to subvert the algorithm. Determining the credibility of each neighbour node requires prior knowledge. A rough prior assessment can be accomplished by using the intrusion response matrix maintained at each node, but this still subjects to uncertainty.

Bayesian network is also being used to evaluate multiple attack sources. Suppose, in Figure 2, node D detects anomalies from A, B and C simultaneously and $P(D|A)$, $P(D|B)$ and $P(D|C)$ are known from the anomalies rules. The trick is that only A is the true attacker. Both B and C unconsciously forward packets from A to D. In such a scenario, we would be interested in computing the posterior probabilities $P(A, B, C|D = 1)$. Suppose the prior probability distributions of A, B and C, that is, $P(A)$, $P(B)$ and $P(C)$, can be estimated as previously described, then for the aforesaid example, we would expect $P(A = 1, B = 0, C = 0|D = 1)$ is the largest posterior probability from all 8 combinations of A, B and C, that is, when D undergoes attack, it is likely that A is the attacking node.

Algorithm 1 Intrusion Response Algorithm

```
/* receive global alert */
if receive CTS with a global alert (attacker_id = a, send_addr = s, p(s|a) = p) then
  if attackFrom[a] ≠ NULL then
    if s is in neighbourAlert[a] then
      discard the global alert
    else
      add (s, p) to neighbourAlert[a]
    end if
  else
    add a new entry (a, s, p) in the intrusion response matrix M
  end if
end if
/* update local alert */
for each a in M do
  if attackFrom[a] ≠ NULL then
    if neighborAlert[a] = ∅ then
      updataLocal[a] = directLocal[a]
    else
      compute updataLocal[a] using equation (3)
    end if
  end if
end for
/* send global alert */
for each a in M do
  globalAlert[a] = 0
end for
if receive RTS from node s then
  for each a in M do
    if s is in neighborAlert[a] then
      globalAlert[a] = directLocal[a]
    else
      globalAlert[a] = updateAlert[a]
    end if
  end for
end if
```

5 Performance evaluation

5.1 Simulation environment

The simulation is conducted on the ns-2 simulator. Table 4 lists the ns-2 parameters in our simulation.

In the simulation, each node starts its move from a random location to a random destination with a randomly selected speed that uniformly distributed between [0, maxspeed]. Once the destination is reached, the node stays there for as

long as specified by *pause time*, then another destination location is chosen. Dynamic network topology and different mobility scenarios are modelled by varying the *maxspeed* and the *pause time*. To prevent all flows start from the beginning at the same time, each source node chooses its starting time for sending packets from the range of [0, s time].

Table 4 Ns-2 simulation environment

Parameter	*Value/choice*
Topology	500 m × 500 m
Node movement	Random waypoint model
Max movement speed	5 m/s
Radio range	250 m
Sending capacity	2 Mbps
Node set count	30
Total number of flows	25
Avg. trans. rate per flow	2 packet/s, 512 byte/packet
Send buffer at each node	A fixed 64-packet
Training data interval	2000 s for both analysis
Testing data interval	50 s/asso-rule, 200 s/cross-fea.
Feature sampling interval	5 s, ignored in asso-rule

5.2 Simulated attacks

Common network-based attacks are resource consumption (power exhaustion, storage and CPU exhaustion and network bandwidth exhaustion) attacks such as flooding and deprivation attacks, routing-disruption attacks such as blackhole and grayhole attacks (Hu and Perrig, 2004). The goal of attacker is to degrade the performance of the network or individual nodes instead of gaining privileges of a particular machine. We simulate the following four attacks against the network layer.

- *Flooding*: an attacking node sends spurious data packets to one destination node. If the destination node is not in the vicinity of the attacking node, the data packets may be delivered to the destination node in multiple hops and they may also take various paths. In such case, all the intermediate nodes become victims.
- *Blackhole*: an attacking node advertises itself as having the best path to any node in the network. After the neighbour nodes receive the advertisement, they update their routing tables and redirect all packets to the attacker.
- *Sleep deprivation*: an attacking node advertises falsified routing control information about one of its neighbouring node, that is, a victim node. That is, the attacker tells everyone that the victim node has the best path to any node in the network. As a result, the victim node could be kept busy by forwarding and dropping (if no route) packets of the neighbour nodes.
- *Packet dropping*: an attacking node drops all data packets passing through it. The attacker continues to participate routing functions to show its existence.

5.3 Detection of simulated attacks

In this section, we show our experimental results over the simulated attacks. Here, detection rate is defined as the ratio of the number of attacks being detected correctly to the total number of attacks occurred during a particular time frame. In our experiments, similar to (Huang and Lee, 2003), we consider a detection is successful as long as one node detects abnormal behaviour in the neighbourhood. False positive rate is defined as the ratio of the number of attack-free events falsely being identified as anomalies to the total number of normal events. In addition, we calculate the average number of nodes that have detected the attack per simulation run for each pause time. This number represents how many nodes involved in an intrusion detection of neighbourhood. The higher this number is, the more success of the detection in the neighbourhood a node achieves.

5.3.1 Data preprocessing

Due to the busty transmission behaviour of mobile nodes, we observe that the total number of transaction records collected according to the direct feature set has large variations among different nodes and different simulation runs. Because the support value of an association rule is directly influenced by the total number of transaction records in each training or testing data interval, a dataset with a small number of transactions is likely to produce rules with large support values. When these support values are used in a global alert, they may mislead the neighbouring nodes, since we associate them with marginal probabilities in the Bayesian network. Therefore, we decide to activate the detection module for association-rule analysis only if the total number of inbound traffic rate reaches a certain percentage (e.g. 5% in our experiments) of the overall traffic rate in the neighbourhood. This condition largely reduces the false alarm rate.

For the cross-feature analysis, Table 5 lists the non-uniform quantisation bins used for the statistical feature set in our simulation. The index of bin number is used to encode the quantised feature value.

Table 5 Quantisation bins of statistical features

Feature	*Quantisation bin*
NAV	[0, 1), [1, 3), [3, 5), [5, inf)
xmitTrafficRate	[0, 102. 4k), [102, 4k, 204.8k), [204.8k, 3072.2k), [307.2k, inf)
recvTrafficRate	same as xmitTrafficRate
reXmitRTS	[0, 3), [3, 5), [5, 7), [5, inf)
reXmitDATA	[0, 7), [7, inf)
nigbrNodeCount	[0, 7), [7, 15), [15, 23), [5, 29]
fwdNodeCount	[0, 0], [1, 1], [2, 2], [3, 3], (3, 29]

5.3.2 Experiment results

Tables 6 and 7 show the experimental results of simulated attacks using both association-rule and cross-feature analysis. The detection rate and the false positive rate are the average value for five different mobility levels, that is, *pause time*

is set to 0, 10, 30, 60 and 200. (Note that mobility is in reverse proportion to the pause time). The false alarm rate in Table 7 refers to the training errors. From both tables, we can see that our IDS can effectively detect the simulated attacks with relatively low false positives. The detection rate of association-rule analysis is higher than cross-feature analysis is because the former has short time interval but larger attack data size.

Table 6 Experiment results of association-rule analysis

Attack type	*Detection rate %*	*False alarm rate %*
Flooding	100	2.8
Blackhole	99.3	0.3
Sleep deprivation	90	0.7
Packet dropping	93	0.5

Table 7 Experiment results of cross-feature analysis

Attack	*Detection rate %*	*False alarm rate %*
Flooding	100	0.29
Blackhole	83.33	0.29
Sleep deprivation	85.33	0.29
Packet dropping	72.00	0.29

In our experiments, we investigate the influence of mobility on detection rate. As expected, the detection rate decreases as mobility of network increases. Figures 3 and 4 are two example plots from the cross-feature analysis. Figure 3 shows detection rate of blackhole attack versus mobility and Figure 4 illustrates the average number of nodes detecting the attack per simulation versus mobility. As illustrated in Figure 3, detection rate increases as mobility decreases (i.e. the pause time increases). From Figure 4, we see that when mobility decreases, the average number of nodes that have detected the attack increases. This means that more nodes are aware of intrusions or abnormal behaviours in the neighbourhood. This result is expected because a node's neighbourhood tends toward stable when mobility of network decreases.

Figure 3 Detection rate of blackhole attack versus mobility

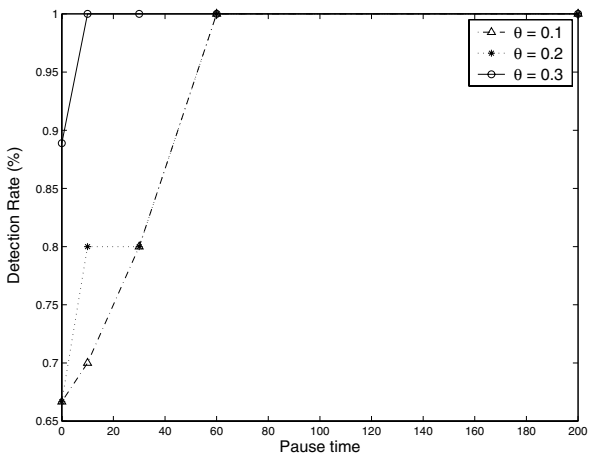
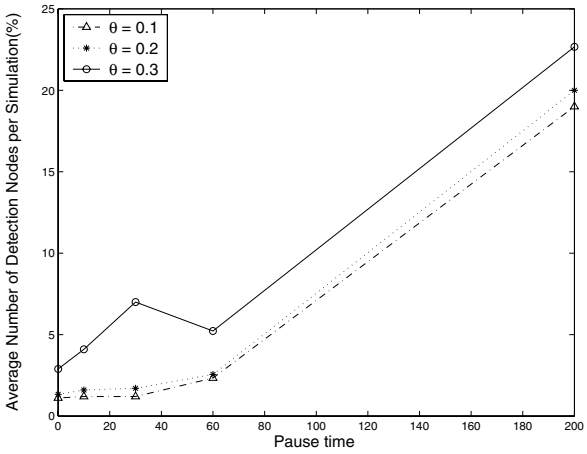


Figure 4 Average number of nodes detecting blackhole attack per simulation versus mobility



The influence of mobility from the association-rule analysis is similar to the cross-feature analysis. That is, mobility does impact the performance of IDS. We conducted an additional set of simulations with the *maxspeed* setting to 10 m/s (instead of 5 m/s as listed in Table 4) to confirm this conclusion. For this set of experiments, we simulate one set of normal training data in an interval of 2000 sec, which is twice as long as the interval used in the 5 m/s dataset. The simulated normal test data and attack datasets are of 200 sec. Ten simulation runs are performed for five different pause time on both test and attack data. Five pause time are 0, 10, 30, 60 and 2000/200, where 2000 represents a still network during training data interval and 200 represents a still network during testing data interval. Because our IDS focuses on most of the resource consumption attacks, which usually occur over a time interval, we use a sliding window (e.g. 50 sec) of intervals with 5 sec overlap to determine whether an attack takes place. Therefore, we segment both training and test data sets into intervals of 50 sec and these segments are used to profile and to detect anomalies as described in Section 4.1. As a result, the detection rates of the four simulated attacks listed in Table 6 are all lower than those obtained on 5 m/s data sets. Specifically, the detection rate of flooding, blackhole, deprivation and packet dropping attacks drop to 91.78%, 71.34%, 40.6% and 77.56%, respectively while keeping the false positive rate less than 0.5%.

In general the experimental results are very promising and competitive to several other reported works in term of detection rates on certain attacks (e.g. Huang and Lee, 2003). However it is difficult to make direct comparisons on these results. To the best of our knowledge, we are the first to exploit MAC layer features for ad hoc IDS. Besides detection accuracy, we are also addressing in attack source localisation and collaborative detection. MAC layer features can provide some unique insight on local activities that may be related to global security breach and our work brings a different perspective to ad hoc IDS research.

Both short-term and long-term profiling brings different characterisation of node behaviours, however the corresponding long-term audit data size is considerably smaller than the short-term audit data size. In practice, for the sake of conserving energy, the proposed hybrid

approach can efficiently incorporate association-rule analysis and cross-feature analysis to maximise detection power while minimising resource consumption. For example, the association-rule analysis can be activated only after the cross-feature analysis detects the abnormality of an audit data segment and the detection results can also be correlated using Bayesian network. In future, we plan to further integrate these two detection modules to improve the system performance. For example, we will investigate the effectiveness of using the outcome (i.e. rules) of association-rule analysis as the income (i.e. features) of cross-feature analysis .

We should point out that our work mainly focuses on detecting network resource consumption attacks, for example, the aforesaid simulated attacks. This set of attacks is especially attractive to attackers because of the limited energy and bandwidth resources of ad hoc networks. Our proposed technique is not suitable to detect MAC spoofing attacks, although it relies on using MAC address to identify attack source. To counteract spoofing attacks, ingress filtering is an easy and effective mechanism for wired networks. However, it does not fit in ad hoc networks due to the lack of centralised administration and control. Instead, authentication mechanism may be used to combat spoofing attacks. Hu et al. (2003) proposed an efficient authentication protocol, called TIK, that can be used along with a MAC layer protocol to efficiently protect against replay, spoofing and wormhole attacks. They showed that TIK does not require significant additional processing overhead at the MAC layer. It is obvious that authentication protocols cannot impede most resource consumption attacks. Therefore, the concept of defense-in-depth is also applied to ad hoc networks.

6 Conclusion

In this paper, we have presented a node-based anomaly IDS for ad hoc networks using unsupervised association-rule mining and cross-feature mining techniques. Two feature sets are defined from the MAC layer network data, that is, the direct feature set and the statistical feature set. They aim at short-term and long-term profiling, respectively. The direct feature set incorporates cross-layer information by tagging the network packet type to each MAC DATA frame, hence easing the complexity of the proposed IDS and extending its detection ability to both the MAC layer and the network layer. In addition, this feature set can be used to localise attack source within one-hop perimeter based on MAC's single hop communication nature. We also propose to use MFI criteria to prune the association rules, which dramatically reduces the size of rule set and reduces the number of redundant alerts. False positive rate is further reduced through the decision module of the IDS where intelligence gathered from neighbour nodes is used to make a collaborative decision and Bayesian network is used to evaluate multiple attack sources. Cross-feature analysis takes an energy-efficient approach by coarsely monitoring the network behaviour of neighbourhood and association-rule analysis provides more accurate detection performance. These two detection modules complement each other in time scale and sensitivity to different attack types. Simulation results demonstrated that the proposed IDS is effective with respect to the simulated attacks.

References

Agrawal, R. and Srikant, R. (1994) 'Fast algorithms for mining association rules', *Proceedings of 20th International Conference on Very Large Databases*, Santiago, Chile, September, pp.487–499.

Broch, J., Maltz, D., Johnson, D., Hu, Y-C. and Jetcheva, J. (1998) 'A performance comparison of multi-hop wireless ad hoc network routing protocols', *Proceedings of Fourth ACM/IEEE Annual International Conference on Mobile Computing and Networking (MobiCom'98)*, October, pp.85–97.

Burdich, D., Calimlim, M. and Gehrke, J. (2001) 'MAFIA: a maximal frequent itemset algorithm for transactional databases', *Proceedings of 17th International Conference on Data Engineering (ICDE)*, April, pp.443–452.

Deng, H., Zeng, Q-A. and Agrawal, D.P. (2003) 'SVM-based intrusion detection system for wireless ad hoc networks', *Proceedings of IEEE 58th Vehicular Technology Conference (VTC'03)*, October, Vol. 3, pp.2147–2151.

Hu, Y-C., Perrig, A. and Johnson, D.B. (2003) 'Packet leashes: a defense against wormhole attacks in wireless networks', *Proceedings of IEEE INFOCOM 2003*, April, Vol. 3, pp.1976–1986.

Hu, Y-C. and Perrig, A. (2004) 'A survey of secure wireless ad hoc routing', *IEEE Security and Privacy Magazine*, Vol. 2, No. 3, pp.28–39.

Huang, Y. and Lee, W. (2003) 'A cooperative intrusion detection system for ad hoc networks', *Proceedings of ACM First Workshop on Security of Ad Hoc and Sensor Networks*, October, pp.135–147.

Huang, Y., Fan, W., Lee, W. and Yu, P.S. (2003) 'Cross-feature analysis for detecting ad-hoc routing anomalies', *Proceedings of 23th International Conference on Distributed Computing Systems (ICDCS)*, May, pp.478–487.

Kachirski, O. and Guha, R. (2002) 'Intrusion detection using mobile agents in wireless ad hoc networks', *Proceedings of IEEE Workshop on Knowledge Media Networking*, July, pp.153–158.

Michiardi, P. and Molva, R. (2002) 'Simulation-based analysis of security exposures in mobile ad hoc networks', *Proceedings of European Wireless Conference (EW2002)*, February.

Thing, V.L.L. and Lee, H.C.J. (2004) 'IP traceback for wireless ad-hoc networks', *Proceedings of IEEE 60th Vehicular Technology Conference (VTC'04)*, September, Vol. 5, pp.3286–3290.

Tseng, C., Balasubramanyam, P., Ko, C., Limprasittiporn, R., Rowe, J. and Levitt, K. (2003) 'A specification-based intrusion detection system for AODV', *Proceedings of First ACM Workshop on Security of Ad Hoc and Sensor Networks*, October, pp.125–134.