SAT: an SVM-based Automated Trust Management System for Mobile Ad-hoc Networks

Wenjia Li
Department of Computer Sciences
Georgia Southern University
Statesboro, GA 30460
wenjiali@georgiasouthern.edu

Anupam Joshi, Tim Finin
Computer Science and Electrical Engineering
University of Maryland, Baltimore County (UMBC)
Baltimore, MD 21250
{joshi, finin}@umbc.edu

***Abstract*—Mobile Ad-hoc Networks (MANETs) are extremely vulnerable to a variety of misbehaviors because of their basic features, including lack of communication infrastructure, short transmission range, and dynamic network topology. To detect and mitigate those misbehaviors, many trust management schemes have been proposed for MANETs. Most rely on pre-defined weights to determine how each apparent misbehavior contributes to an overall measure of trustworthiness. The extremely dynamic nature of MANETs makes it difficult, however, to determine a set of weights that are appropriate for all contexts. We describe an automated trust management scheme for MANETs that uses machine learning to classify nodes as malicious. Our scheme is far more resilient to the context changes common in MANETs, such as those due to malicious nodes altering their misbehavior patterns over time or rapid changes in environmental factors, such as the motion speed and transmission range. We compare our scheme to existing approaches and present evaluation results obtained from simulation studies.**

***Keywords*-security; trust management; mobile ad hoc network; Support Vector Machine**

I. Introduction

In general, a mobile ad hoc network (MANET) is composed of a *dynamic* set of nodes that rely on each other to relay packets due to the lack of a fixed networking infrastructure. MANETs have been widely used in a variety of military scenarios, such as soldiers exchanging information for situational awareness on the battlefield, search teams coordinating in combat search and rescue efforts, and real-time enemy detection around a troop station.

Compared with traditional infrastructure-based networks, MANETs are more susceptible to malicious attacks and random failures due to their unique features such as constrained node energy, error-prone communication media, and dynamic network topology. Therefore, security is a key concern for MANETs. Security threats in MANETs come from both malfunction of mobile devices and subversion to these devices by enemies.

To cope with the security threat posed by misbehaving nodes in MANETs, a variety of solutions have been studied in the past decade, such as misbehavior detection mechanisms [1], [2], [3] and trust management schemes [4], [5], [6]. Trust management schemes are used to evaluate a node's behavior and determine whether it is trustworthy or not in terms of how cooperative it is. The majority of the current trust management schemes rely on a pre-defined formula to determine the trustworthiness of each node. However, a fixed formula is not able to accurately distinguish the difference on a node's trustworthiness in many scenarios. Figure 1 depicts such a scenario in which a fixed formula with pre-determined weights cannot precisely describe the trustworthiness of mobile nodes.

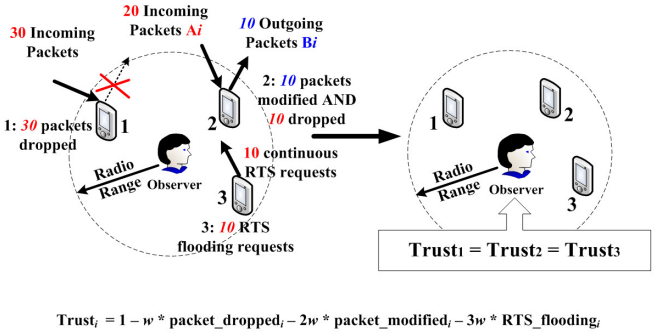


Fig. 1. This figure depicts a scenario where the use of a fixed formula with pre-defined weights is not sufficient to accurately evaluate node trust.

In Figure 1, we find that the observer uses a fixed formula with pre-defined weights to evaluate node trust. According to the formula, node 1, 2 and 3 are *equally* trustworthy in this scenario. However, it is true that nodes 1, 2 and 3 are equally trustworthy because they have exhibited different types of misbehaviors with varying frequencies. In addition, the weights ought to be adjusted according to the nature of the misbehavior and the context in which the misbehavior occurs, which is not feasible because of the dynamic nature of MANETs. Thus, we argue that a pre-determined formula cannot precisely indicate the trustworthiness of mobile nodes in MANETs.

We describe *SAT*, SVM-based Automated Trust Management, as an approach to evaluating the trustworthiness of MANET nodes that uses the support vector machine technique to learn how to combine evidence. Unlike the traditional trust

Partial support for this work was provided by MURI award FA9550-08-1-0265 from the Air Force Ofce of Scientic Research. This work was done when the first author was a PhD candidate in the Department of Computer Science and Electrical Engineering at University of Maryland, Baltimore County

management schemes such as [7], [4], [5], [8], the SAT scheme neither uses a fixed formula to calculate the trustworthiness of mobile nodes, nor does it rely on a set of pre-defined weights to punish various misbehaviors at different paces. Instead, a SVM classifier is trained in SAT scheme and then used to determine the trustworthiness of the nodes in an automated manner.

This work offers two contributions. The first is the use of an *automated* trust management scheme using the Support Vector Machine (SVM) technique [9] to automatically determine how each misbehavior should be punished and what the trustworthiness is for each mobile node according to various contexts. The second is the articulation of a set of *sophisticated* attack patterns of the adversaries, which rarely have been discussed in previous trust management schemes [4], [5], [8]. The SAT scheme assumes that adversaries can alter their attack patterns from time to time. In addition, adversaries can choose to conduct misbehaviors for various length of time. By this means, we believe that the attack patterns used in the simulation can better represent the actual behaviors of adversaries in practice.

II. Related Work

In the past decade, many research efforts have been made to address the security needs for MANETs by means of trust management [10]. The main goal of trust management is to evaluate the actions of other nodes, and build a reputation for each node based on the node evaluation result. The reputation can then be used to determine the trustworthiness for other nodes. The trustworthiness can be utilized to make choices on which nodes to cooperate with, or even take action to punish an untrustworthy node if necessary.

Trust is divided into *direct* trust and *indirect* trust [11]. Direct trust stems from the *first-hand* observations locally obtained by a node itself, while indirect trust refers to the *second-hand* observations released by other nodes. In MANETs, direct trust cannot always provide comprehensive evaluation of the target node due to exterior circumstances such as channel conditions, temporary unavailability, interference, etc. At this time, indirect trust is used to provide secondary information to help evaluate the actual trustworthiness of the target node.

In [4], Buchegger et al. proposed the CONFIDANT protocol to encourage the node cooperation and punish misbehaving nodes. Michiardi et al. [5] presented a mechanism with the name CORE to identify selfish nodes, and then compel them to cooperate in the following routing activities. Patwardhan et al. [8] studied an approach in which the reputation of a node is determined by data validation.

In our previous research work [7], [12], we proposed a *multi*-dimensional trust management scheme for MANETs. In this framework, the trustworthiness of a node is judged from different *perspectives* (i.e., *dimensions*), and each dimension of the trustworthiness is derived from various sets of misbehaviors according to the nature of those misbehaviors. However, each dimension of trustworthiness is still derived from a pre-defined formula with a set of fixed weights, which still cannot well adapt to complicated scenarios or context changes.

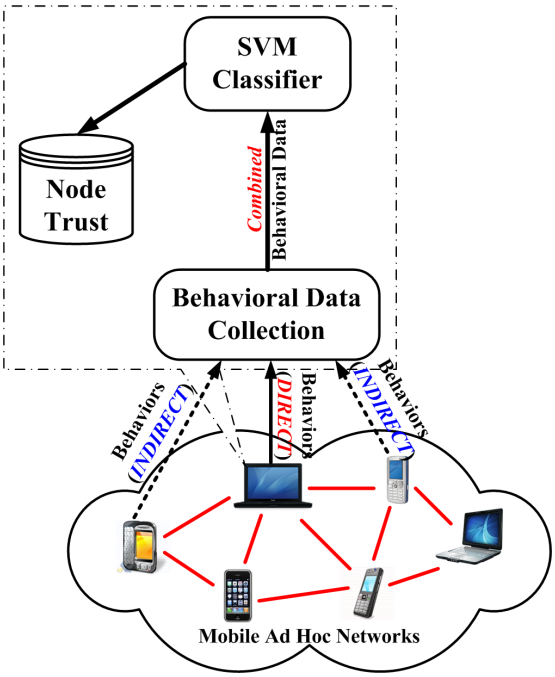


Fig. 2. The SAT Scheme

III. SAT: SVM-based Automated Trust Management

The SAT framework has two functional modules: *Behavior Data Collection* and *Trust Management* as shown in Figure 2.

A. Behavioral Data Collection

The behavioral data collection module is responsible for the collection of node behaviors and formation of behavioral dataset. In this paper, a node's behavior is described in terms of the ratio of the amount of this behavior over the total amount of packets that the node has received, such as *packet drop rate* (PDR), *packet modification rate* (PMR) and *RTS flooding rate* (RTS).

We use network simulations to generate behavioral dataset and train a SVM classifier. Because the adversaries and their misbehaviors are pre-defined in these simulations, the behavioral data are collected and then labeled according to the ground truth regarding adversaries. The trained SVM classifier can then be distributed and deployed to mobile devices to classify nodes in MANETs in which they participate. An example training dataset is shown in Table I. Here, we create a m-dimensional feature vector for each node. In the example shown in Table I, $m = 3$.

During the testing stage, the Behavioral Data Collection module on each node first observes and records the behaviors of their neighbors. It also receives and integrates node behaviors reported by other nodes. In this paper, we use Dempster-Shafer Theory [13] to combine the observations, as discussed in details in our previous work [14].

$Node\ ID$	PDR	PMR	RTS
1	90%	10%	0
2	2%	0	0
3	30%	60%	10%
4	5%	0	0
5	10%	0	90%
...	...	...	...

TABLE I
AN EXAMPLE OF THE TRAINING DATASET USED TO CLASSIFY MANET NODES AS MALICIOUS OR NON-MALICIOUS.

B. Trust Management Using Support Vector Machine

Support Vector Machine technique [9] is used in our SAT scheme to evaluate the trustworthiness of nodes in MANETs. More specifically, we use SVM^{rank} [15] in our experiments to determine the trustworthiness of nodes in a ranked list.

Initially, each node observes and records neighbor behaviors, and these local observations are fed into the SVM^{rank} classifier to produce the initial trustworthiness in a ranked list. Because each node can only observe behaviors of its direct neighbors, the local observations are then exchanged among nodes so that each node can also know the behaviors of other nodes that are out of its radio range. The local observations and foreign observations obtained from other nodes are fused together using Dempster-Shafer Theory and thus an updated behavioral dataset is generated. If the updated behavioral dataset makes SVM^{rank} classifier produce a ranked list with different order than the previous one, then the updated behavioral dataset is propagated to all neighbors. Once there is not any change in the trust evaluation result when they receive foreign behavioral data, the procedure terminates. At this point, all the nodes have the same belief of node trustworthiness.

IV. PERFORMANCE EVALUATION AND ANALYSIS

In this section, we examine the performance of the SAT scheme, and its performance is compared to that of the baseline mechanism. The baseline mechanism that we choose here is the Multi-dimensional Trust management framework (*mTrust*) discussed in our prior work [7], and our prior work has shown that *mTrust* framework outperforms other well-known mechanisms [7].

A. Adversary Model

In this paper, we assume that the adversary can conduct multiple misbehaviors at the same time, and it can mix these misbehaviors at any ratio. In addition, the adversary can alter this ratio over time, and it can carry out the set of misbehaviors for any arbitrary length of time. For instance, an adversary A may determine at time t_1 that it should equally conduct the four types of misbehaviors (i.e., RTS flooding, packet dropping, packet modification, and packet misroute); while at time t_2, A changes its misbehavior pattern to solely perform RTS flooding attack.

$Parameter$	$Value$
Simulation area	$600m \times 600m$
Num. of nodes	50, 100, 200
Transmission range	$60m, 90m, 120m$
Mobility pattern	*Random Waypoint*
Node motion speed	$5m, 10m, 20m$
Num. of bad nodes	5, 10, 20
Simulation time	900s

TABLE II
SIMULATION PARAMETERS

$Node\ ID$	$Start$	End	$Drop$	$Modify$	RTS
1	$0s$	$900s$	80%	20%	0
2	$0s$	$900s$	0	50%	50%
3	$0s$	$900s$	30%	30%	40%
4	$0s$	$900s$	20%	10%	70%
5	$0s$	$900s$	10%	0	90%

TABLE III
AN EXAMPLE OF MISBEHAVIOR SETUP FOR THE TRAINING STAGE.

B. Simulation Setup

We use GloMoSim 2.03 [16] as the simulation platform, and table II lists the parameters used in the simulation scenarios.

We use four parameters to evaluate the correctness and efficiency of the SAT scheme: *Precision*, *Recall*, *Communication Overhead* (CO), and *Convergence Time* (CT). CO and CT are defined as follows.

$$CO = \frac{Number\ of\ Packets\ for\ the\ Framework}{Total\ Number\ of\ Packets\ in\ the\ Network}$$

$$CT = Time\ taken\ to\ form\ a\ unique\ global\ view$$

Overall, the experiments can be divided into two phases: the training stage and the testing stage. In the training stage, there are totally 100 nodes in the simulated MANET, with five of them being misbehaving nodes and the rest 95 nodes being well-behaved nodes. These five misbehaving nodes conduct a mixed set of various misbehaviors including packet dropping, packet modification, RTS flooding, and fake observation spreading, and the mixture rates for these misbehaviors may vary among the misbehaving nodes. However, the mixture rate for each misbehaving node is fixed throughout the training stage. Table III depicts an example setup of misbehaviors for our simulation. Note that *Start* and *End* denote the start time and end time for the specified set of misbehaviors respectively. Each value in the columns of *Drop*, *Modify* and *RTS* stands for the percentage of the corresponding misbehavior over the whole set of misbehaviors in the specified time range, and the percentages of these three misbehaviors sum to one. Just take the first entry as an example: 80% of misbehaviors conducted by node 1 will be packet dropping and 20% of that be packet modification during the time range between 0s and 900s. The "rumor spreading" misbehavior will be defined separately.

In the testing stage, we vary the number of misbehaving nodes by 5, 10 or 20 in different experimental scenarios, not only the mixture rates for different nodes vary, the mixture rate for the same misbehaving node conducts may also differ

over time. In this way, we guarantee that the training dataset differs significantly from the testing dataset. In addition, some adversaries may alter their attack patterns so as to make their behaviors less deviated from the normal behaviors. We will further discuss different attack patterns that are considered in our experiments in Section IV-C2.

Each simulation scenario has 30 runs with distinct random seeds, which ensures a unique initial node placement for each run. Each experimental result is the average over the 30 runs for this simulation scenario.

C. Simulation Results

The performance of *SAT* is observed and compared to that of *mTrust* in several simulation scenarios. The simulation results show that: (1) In general, *SAT* achieves a good performance in terms of proper evaluation of node trustworthiness, quickly convergence, and acceptable communication cost; and (2) *SAT* outperforms *mTrust* especially in the scenarios in which there are some *insidious* adversaries that periodically alter their attack patterns. The simulation results are presented in details below.

1) Overall Performance of SAT: To evaluate the performance of the SAT scheme, we observe the performance of *SAT* as well as that of *mTrust* in the following four scenarios: different number of nodes, different radio ranges, different percentage of misbehaving nodes, and different node motion speeds. Given that the simulation area remains the same in all these scenarios, we can observe from these scenarios the effect of node density, radio range, percentage of adversary , and node mobility, respectively. Note that each of the misbehaving nodes mixes all the misbehaviors with a fixed rate in these experimental scenarios. In addition, there are 5 adversaries in the network except for the third scenario (i.e., different percentage of misbehaving nodes), in which the number of adversaries is 5, 10 or 20. The simulation results are showed in the following Figure 3 and Figure 4.

Figure 3 shows that *SAT* yields an overall higher precision than *mTrust* in all these four cases.We also find from Figure 4 that *SAT* outperforms *mTrust* in all cases in terms of recall.

Moreover, we also observe the performance of *SAT* and *mTrust* in terms of communication overhead and convergence time. The simulation results show that *SAT* produces a good performance in that it converges in a short period of time with a small communication overhead.

2) Effect of Various Attack Patterns: In this simulation scenario, We compare the performance of *SAT* and *mTrust* under different attack patterns. In the traditional trust management schemes, adversaried are generally supposed to conduct a fixed set of misbehaviors throughout the simulation. However, in practice an *insidious* adversary may choose to periodically alter its attack pattern so that its trustworthiness are harder to get punished. For example, an adversary can change the duration as well as the mixture rate of its misbehaviors from time to time, which makes its behaviors not so diverse from the normal behaviors. More specifically, we identify two novel attack patterns, namely *Short-term* and *Everchanging*, in our simulation scenarios. Table IV and V give an example of *Short-term* and *Everchanging* attack patterns, respectively.

$Node\ ID$	$Start$	End	$Drop$	$Modify$	RTS
16	$0s$	$100s$	30%	60%	10%
33	$150s$	$300s$	0	10%	90%
34	$50s$	$250s$	40%	30%	30%
44	$400s$	$680s$	50%	50%	0%
45	$650s$	$820s$	20%	0	80%

TABLE IV
AN EXAMPLE OF *Short-term* ATTACK PATTERN

$Node\ ID$	$Start$	End	$Drop$	$Modify$	RTS
16	$0s$	$200s$	10%	70%	20%
16	$200s$	$600s$	30%	0%	70%
16	$600s$	$900s$	90%	10%	0
33	$0s$	$500s$	0	10%	90%
33	$500s$	$900s$	30%	30%	40%
...	...	...	...	...	...

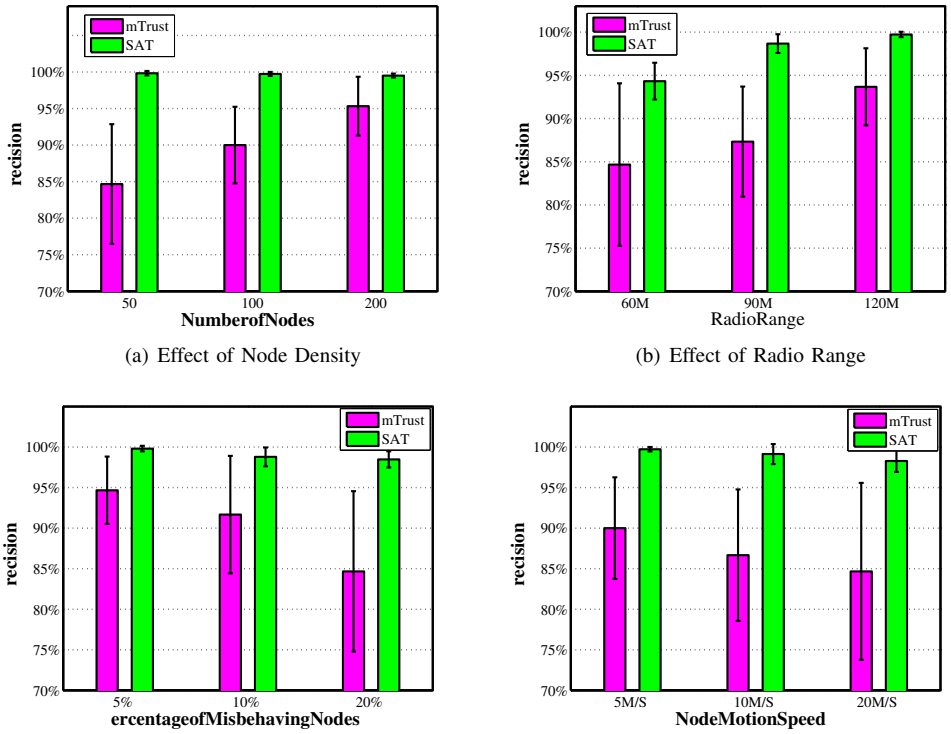
TABLE V
AN EXAMPLE OF *Everchanging* ATTACK PATTERN

To observe the possible effects of various attack patterns on trust management, we deploy these two novel attack patterns in our simulation. To the best of our knowledge, traditional trust management mechanisms assume that the adversaries generally do not change their attack patterns. Therefore, the *Comprehensive* attack pattern is used as the baseline for comparison purpose, in which each adversary chooses a different mixture rate of misbehaviors and keeps this rate unchanged throughout the whole simulation. The simulation results are shown in Fig. 5.

From Fig. 5(a) and Fig. 5(b) we observe that *SAT* always achieves a better performance than *mTrust* in terms of a higher precision score as well as a higher recall score. Fig. 5(c) depicts that it generally takes less time for *SAT* than *mTrust* to converge. Moreover, we see from Fig. 5(d) that *SAT* usually introduces a smaller communication overhead than *mTrust* does under all three attack patterns. Therefore, we can conclude from Fig. 5 that when compared to *mTrust*, *SAT* always yields a better performance under all attack patterns in that *SAT* achieves higher precision and recall in shorter time and with lower communication cost. Given that these novel attack patterns can help adversaries better hide themselves from being caught, it is more likely that the SAT scheme will work better in practice.

V. CONCLUSION

The purpose of trust management schemes is to properly evaluate the trustworthiness of nodes and thus identify and mitigate misbehaviors. We presented an automated trust management scheme for MANETs that uses an SVM-based classifier to assess the trustworthiness of nodes. In addition, we have studied several novel attack patterns that have rarely been discussed by the traditional mechanisms. Simulation results have proven that our approach outperforms the previous schemes and is more resilient to some extremely adverse cases,



(a) Effect of Node Density

(b) Effect of Radio Range

(c) Effect of Adversary Percentage

(d) Effect of Node Mobility

Fig. 3. Precision of *SAT* V.S. *mTrust*

such as networks with a large fraction of misbehaving nodes or where nodes are moving quickly, as in a highway-based vehicular MANETs.

REFERENCES

[1] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 275–283.

[2] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 255–265.

[3] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T. Karygiannis, "Secure routing and intrusion detection in ad hoc networks," in *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, 2005. PerCom 2005.* IEEE, March 2005, pp. 191–199.

[4] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the confidant protocol," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2002, pp. 226–236.

[5] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*. Deventer, The Netherlands, The Netherlands: Kluwer, B.V., 2002, pp. 107–121.

[6] Q. He, D. Wu, and P. Khosla, "Sori: a secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proceedings of 2004 IEEE Wireless Communications and Networking Conference, WCNC '04.*, vol. 2, March 2004, pp. 825–830 Vol.2.

[7] W. Li, A. Joshi, and T. Finin, "Coping with node misbehaviors in ad hoc networks: A multi-dimensional trust management approach," in *Proceedings of the Eleventeenth International Conference on Mobile Data Management, 2010. MDM '10.* IEEE Computer Society, May 2010.

[8] A. Patwardhan, A. Joshi, T. Finin, and Y. Yesha, "A data intensive reputation management scheme for vehicular ad hoc networks," in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops, Mobiquitous '06.*, July 2006, pp. 1–8.

[9] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines : and other kernel-based learning methods*, 1st ed. Cambridge University Press, March 2000.

[10] J. Cho, A. Swami, and I. Chen, "A survey on trust management for mobile ad hoc networks," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1 –22, 2010.

[11] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*. New York, NY, USA: ACM, 2004, pp. 1–10.

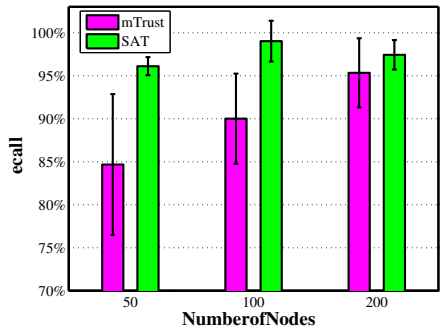
[12] W. Li, J. Parker, and A. Joshi, "Security through collaboration and trust in manets," *ACM/Springer Mobile Networks and Applications (MONET)*, pp. 1–11, 2010 (Online First).

[13] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ, USA: Princeton University Press, 1976.

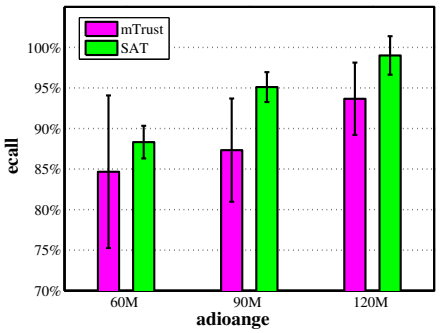
[14] W. Li and A. Joshi, "Outlier detection in ad hoc networks using dempster-shafer theory," in *Proceedings of the Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09.* IEEE Computer Society, May 2009, pp. 112–121.

[15] T. Joachims, "Training linear svms in linear time," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 217–226. [Online]. Available: http://doi.acm.org/10.1145/1150402.1150429

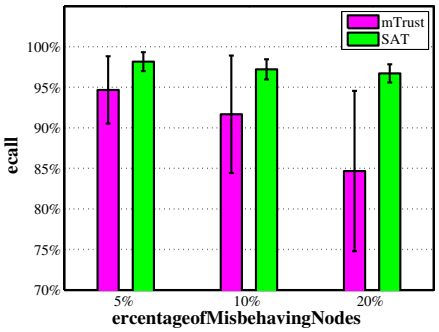
[16] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: a library for parallel simulation of large-scale wireless networks," *ACM SIGSIM Simulation Digest*, vol. 28, no. 1, pp. 154–161, 1998.



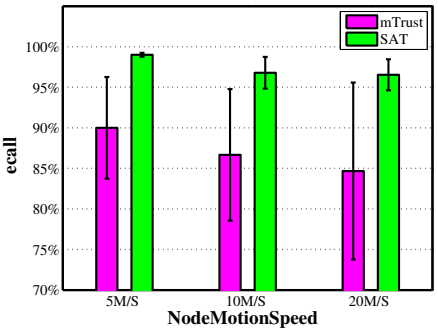
(a) Effect of Node Density



(b) Effect of Radio Range

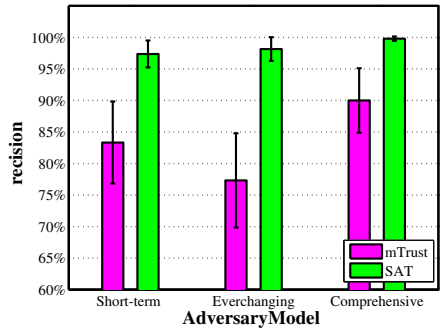


(c) Effect of Adversary Percentage

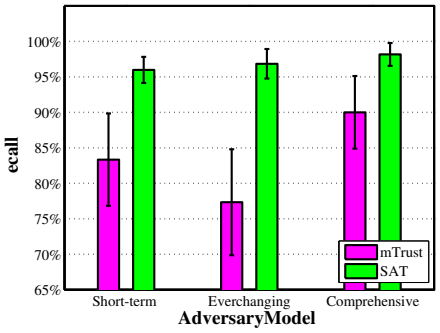


(d) Effect of Node Mobility

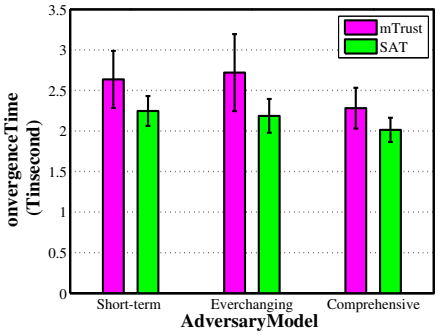
Fig. 4. Recall of *SAT* V.S. *mTrust*



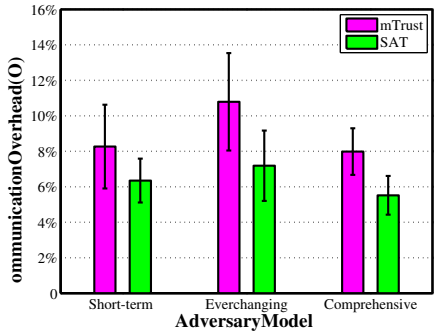
(a) Precision



(b) Recall



(c) Convergence Time (CT)



(d) Communication Overhead (CO)

Fig. 5. Effect of Various Attack Patterns on *SAT* and *mTrust*