Evaluating Intrusion Sensitivity Allocation with Support Vector Machine for Collaborative Intrusion Detection

Li, Wenjuan; Meng, Weizhi; Kwok, Lam For

Published in:
Information Security Practice and Experience

Link to article, DOI:
10.1007/978-3-030-34339-2_26

Publication date:
2019

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Li, W., Meng, W., & Kwok, L. F. (2019). Evaluating Intrusion Sensitivity Allocation with Support Vector Machine for Collaborative Intrusion Detection. In *Information Security Practice and Experience* (pp. 453-63). Springer. https://doi.org/10.1007/978-3-030-34339-2_26

Evaluating Intrusion Sensitivity Allocation with Support Vector Machine for Collaborative Intrusion Detection

Wenjuan Li[1], Weizhi Meng[2](✉), and Lam For Kwok[1]

[1] Department of Computer Science, City University of Hong Kong, Hong Kong, China
[2] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark
weme@dtu.dk

Abstract. The aim of collaborative intrusion detection networks (CIDNs) is to provide better detection performance over a single IDS, through allowing IDS nodes to exchange data or information with each other. Nevertheless, CIDNs may be vulnerable to insider attacks, and there is a great need for deploying appropriate trust management schemes to protect CIDNs in practice. In this work, we advocate the effectiveness of intrusion sensitivity-based trust management model and describe an engineering way to automatically allocate the sensitivity values by using a support vector machine (SVM) classifier. To explore the allocation performance, we compare our classifier with several traditional supervised algorithms in the evaluation. We further investigate the performance of our enhanced trust management scheme in a real network environment under adversarial scenarios, and the experimental results indicate that our approach can be more effective in detecting insider attacks as compared with similar approaches.

Keywords: Collaborative intrusion detection · Intrusion sensitivity · Supervised learning · Trust management · Insider threat

1 Introduction

To protect various computer or network assets, intrusion detection systems (IDSs) are one of the most commonly adopted solutions in practice [19]. As intrusions are becoming more complicated, collaborative intrusion detection networks (CIDNs) are proposed to enhance the detection performance of a single IDS [22,23]. A CIDN allows various IDS nodes to exchange data and learn with each other.

However, CIDNs may be vulnerable to insider attacks due to the distributed architecture, in which an intruder can control an internal node within the network. For instance, if an attack successfully hijack one internal node, then more attacks can be launched via this compromised node. Insider threat can greatly

S.-H. Heng and J. Lopez (Eds.): ISPEC 2019, LNCS 11879, pp. 453–463, 2019.
https://doi.org/10.1007/978-3-030-34339-2_26

degrade the security of collaborative systems and networks. Therefore, it is very important to design appropriate trust management schemes to help safeguard CIDNs.

Motivations. In real scenarios, it is found that IDS nodes may have different detection capability regarding one particular attack. This may be caused by different configuration and settings, i.e., one node has more detection rules than the other. In the previous work [8], the authors defined a term called *intrusion sensitivity* to describe the capability of identifying specific attacks. The use of *intrusion sensitivity* is expected to enhance the detection performance by highlighting the impact of expert nodes. To the best of our knowledge, there are few studies focusing on exploring the influence of intrusion sensitivity in practice. In [10], their results indicated that the application of intrusion sensitivity can provide more efficient detection of pollution attacks. However, how to assign the sensitivity values in an automatic way still remains a challenging issue.

Contributions. Previous work [11] showed that using supervised learning algorithms is a good way to help intelligently allocate the sensitivity values, while it requires at least 60 labeled alarms to achieve good accuracy. Motivated by this observation, in this article, we target on this challenge and enhance the intrusion sensitivity-based trust management scheme by leveraging a support vector machine (SVM) classifier to reduce the required labeled alarms for allocating sensitivity values. Further, we provide an engineering way of allocating the sensitivity based on SVM in a real scenario. The contributions can be summarized as follows.

– We improve the intrusion sensitivity-based trust management scheme in [11] by using a support vector machine (SVM) classifier to allocate sensitivity values. Our experimental results indicate that our approach can reduce the required number of labeled alarms, as compared with several traditional algorithms like decision tree and KNN.
– We introduce an engineering way of implementing both the allocation of sensitivity values and the derivation of satisfaction level for the received feedback for CIDNs. In practice, expert knowledge is very helpful and important to ensure the quality of allocation.
– We collaborate with an IT organization and evaluate the performance of our enhanced trust management scheme in a real network environment under adversarial scenarios, like newcomer and betrayal attack. Our experimental results indicate that our approach can achieve better performance in identifying untruthful insider nodes as compared with similar approaches.

The reminder of this article is structured as follows. We review related studies on distributed and collaborative intrusion detection in Sect. 2. Section 3 introduces the basic architecture of intrusion sensitivity-based trust management scheme for CIDNs, and presents an engineering way of allocating sensitivity values using the SVM classifier. In Sect. 4, we present and analyze evaluation results. Section 5 concludes our work.

2 Related Work

Distributed or collaborative intrusion detection schemes are usually vulnerable to insider attacks (or internal attacks). To construct an effective trust management scheme is a necessary and important solution. For this purpose, some trust management models have been designed in the literature. An Overlay IDS was proposed by Duma *et al.* [3], aiming to defend distributed intrusion detection against insider attacks. The major limitation is that it could not be effective in detecting malicious nodes that have good reputation before in a fast manner. This is because all nodes have the same impact regardless of the behavior changes.

Fung *et al.* [4] then introduced a kind of challenge-based CIDN, which the reputation was measured by identifying the satisfaction levels between the received feedback and the expected answers. They also enhanced the detection using a forgetting factor, which highlights the recent behavior of a node. Then, Dirichlet-based trust model [5] was proposed to help improve the trust evaluation and the balance between detection and false rates. Several other related studies on collaborative intrusion detection can be referred to [2,6,7,14,16,20,22].

Discussion. CIDNs have been gradually adopted by many organizations, but it is very important to protect the security of these mechanisms against insider threat. Most existing research provided many approaches to improve the detection like [6], whereas they did not consider different detection capability of IDS nodes in practice.

In the previous work [8], the authors firstly defined a notion of *intrusion sensitivity* by noticing the different levels of detection sensitivity among IDS nodes. This term aims to help enhance the trust computation and alarm aggregation by highlighting the influence of expert nodes, those who have stronger detection accuracy regarding certain attacks. However, allocating the value manually is time-consuming and error-prone with the increasing size of nodes. How to automatically allocate the values remains a challenging issue.

For this issue, relevant work [11] had shown that supervised learning can help allocate sensitivity values in an automatic way. Motivated by this observation, in this work, we target on this issue and enhance the intrusion sensitivity-based trust management model by using an SVM classifier. This classifier can achieve the same accuracy by reducing the required labeled alarms. We also provide an engineering way of implementing the value allocation and investigate the performance in a real network environment. Our experimental results demonstrate that our approach can help defend CIDNs against insider attacks more effectively than similar approaches.

3 Intrusion Sensitivity-Based Trust Management Model for CIDNs

This section introduces the basic architecture of CIDNs like components and interaction, and then describe how to use the SVM classifier to allocate sensitivity values for IDS nodes.

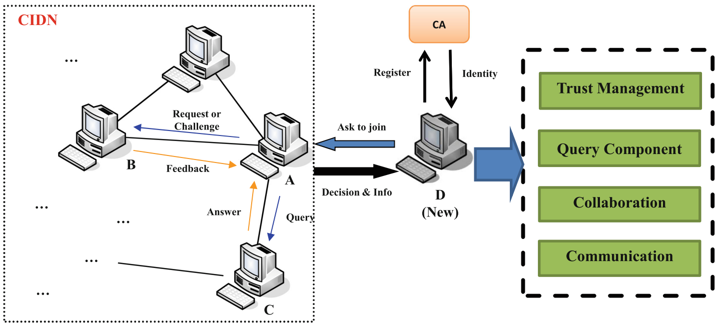

Fig. 1. The CIDN architecture including exchanged messages and major components.

3.1 CIDN Architecture

Figure 1 depicts the architecture of CIDNs like exchanged messages and major components, i.e., *trust management component*, *query component*, *collaboration component* and *communication component*.

Node Registration. Such kind of CIDN allows each node selecting its partner nodes based on its own rules or policies, and recording them in a list. If a node wants to join, the first step is to obtain an identity from a certificate authority (CA). As shown in Fig. 1, a new node D should deliver a joining request to CIDN nodes, say node A. Based on the predefined policies, node A can make a decision whether to accept node D or not.

Trust Management Component. The main task of this component is to manage trust computation. In this work, we adopted the feedback-based trust (or challenge-based trust) [4], in which the reputation is measured by identifying the satisfaction level regarding the received feedback.

Collaboration Component. This component handles the communication among different nodes. There are three types of messages can be used in a challenge-based CIDN, such as *normal requests*, *challenges* and *feedback*. More details can refer to previous work [11].

Query Component. This component is used to measure the intrusion sensitivity of other nodes. A query containing a set of alarms can be sent to the target node. Based on the feedback, it can decide the sensitivity level accordingly.

Communication Component. This component mainly handles the connection with other nodes, i.e., building and maintaining a P2P connection among IDS nodes.

3.2 Trust Computation and Evaluation

To measure the reputation of an IDS node, a *challenge* can be sent to the target node periodically via a random generation process (i.e., the time of sending is random). To facilitate the comparison with similar approaches, in this work, we adopt and update the trust computation based on relevant studies [4,11], as below.

$$T^{i,j}_{value} = w_s \frac{\sum_{k=0}^{n} F^j_k \lambda^{tk}}{\sum_{k=0}^{n} \lambda^{tk}} \tag{1}$$

where $F^j_k \in [0,1]$ represents the satisfaction level for the received feedback k, n means the total amount of received feedback, λ represents a *forgetting factor* that highlights more weight to the recent response and behavior, w_s is a *significant value*, which can be varied based on the total amount of received feedback. That is, if the number of received feedback is smaller than a value m, then $w_s = \frac{\sum_{k=0}^{n} \lambda^{tk}}{m}$; otherwise we set w_s to 1.

Then, we adopt the following weighted majority approach to derive the reputation of a node j.

$$T_j = \frac{\sum_{T \geq r} T^{i,j}_{value} D^j_i I^i_s}{\sum_{T \geq r} T^{i,j}_{value} D^j_i} \tag{2}$$

where $T^{i,j}_{value}(\in [0,1])$ means the reputation level of node i according to node j, $D^j_i(\in [0,1])$ indicates the relationship between these two nodes and the in-between hops, r represents a threshold to filter those nodes whose reputation is smaller than the threshold, $I^i_s(\in [0,1])$ indicates the value of intrusion sensitivity of node i.

3.3 Intrusion Sensitivity Allocation in an Engineering Way

Traditionally, research studies often measure the detection capability among various nodes using a normal distribution, but it cannot reflect the real-world applications [4,5]. We advocate that *intrusion sensitivity* provides a metric to evaluate the detection capability of an IDS node in practice.

In the above CIDN architecture, sensitivity values can be derived by sending *queries* to other nodes. However, as human efforts are error-prone and expensive, it is still a big challenge on how to allocate the values in an intelligent and automatic way [8].

Focused on this challenge, in this work, we advocate the use of supervised learning to help allocate the values based on expert knowledge in [11], while we propose to use a multi-class support vector machine (SVM) classifier [12] to enhance the allocation performance. The merits of using such classifier are shown below [1]:

- SVM provides flexibility in selecting the form of the threshold, which does not need to be linear and even not require the same functional form for each data item. This is because its function can operate locally and be non-parametric.
- SVM is robust especially for a small amount of data items. There is no assumption needed about the functional form of the transformation, i.e., human expertise judgement beforehand is not needed.

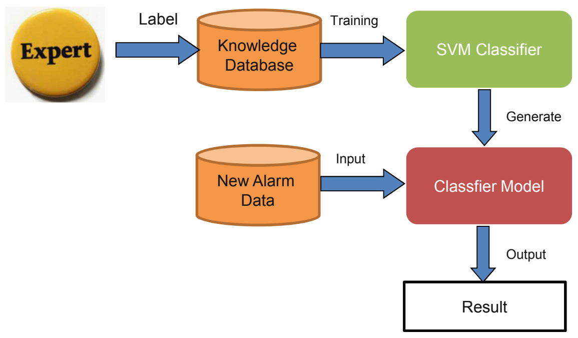

Fig. 2. The allocation of sensitivity values using SVM classifier in an engineering way.

– SVM can provide a unique and robust solution, i.e., a good out-of-sample generalization. In other words, SVM can be still robust even under biased training samples as long as selecting an appropriate generalization grade.

Similar to the previous studies [11], in this work, we also invited three security experts (with more than six years' experience) from the participating organizations (in our evaluation) to help label some alarm items. Figure 2 shows how to allocate the sensitivity values using the SVM classifier and expert knowledge in an engineering way.

4 Evaluation

4.1 Classifier Performance

In this part, we compare the performance of SVM with three commonly used supervised classifiers in allocating the values of intrusion sensitivity, including k-nearest neighbors (KNN), back-propagation neural networks (BPNN) and decision tree (DT).

– *KNN.* This is a kind of instance-based learning, which can classify new instances based on the similarity to the known items. The detailed steps can refer to the previous work [11].
– *BPNN.* This classifier is a kind of supervised classifier that can minimize the error by adjusting the weight values via the process of back propagation. We use the typical BPNN developed in [15], which has three layers like input, output and hidden layer.
– *DT.* This is a popular classifier that can generate a model to predict the label of an item by using a tree-like structure. In the comparison, we employ the algorithm developed in [24].

In this comparison, similar to [11], we investigate different alarm numbers in training like 30, 40, 50 and 60 alarms. We define *intrusion sensitivity* (I_s^i) to be

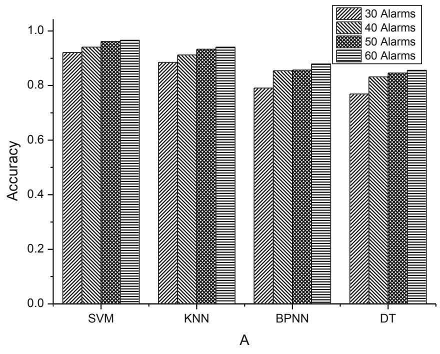

Fig. 3. The classification accuracy among different classifiers.

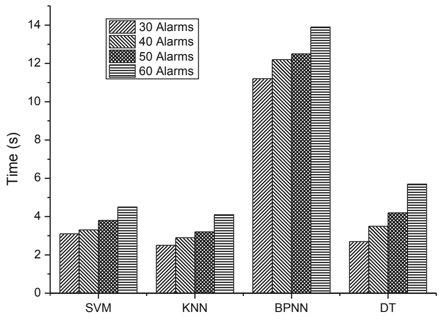

Fig. 4. The time consumption among different classifiers.

ten levels such as expert (1.0), excellent (0.9), very high (0.8), high (0.7), good (0.6), neural (0.5), not good (0.4), low (0.3), very low (0.2), and bad (0.1). In the evaluation, we mainly considered Snort, which is an open-source signature-based IDS [18,21]. Its alarm has three priority levels: high, medium and low.

In particular, we collected 300 labeled alarms that were labeled by security experts. In the phase of training, each classifier was trained with a set of labeled alarms. The process is similar to [11], we trained the classifier with 60 alarms (randomly selected from the database) for labeling 30 new alarms, while we trained the classifier with 120 alarms (randomly selected from the database) for labeling 60 new alarms. We repeated this experiment for ten times (via cross-validation) to avoid some bias. Figures 3 and 4 shows the classification accuracy and time consumption, respectively. The main observations are discussed below:

– *Classification accuracy.* It is found that SVM could reach better classification accuracy than other three classifiers, i.e., it can achieve an accuracy rate of 0.941, 0.961, and 0.966 for 40, 50, and 60 alarms, respectively. In the comparison, our SVM classifier can achieve the same accuracy by reducing the required labeled alarms in the training phase, i.e., SVM can achieve the accuracy of above 0.96 for 50 alarms, while KNN [11] requires 60 alarms for reaching the same accuracy.
– *Time consumption.* Intuitively, inputting more alarms would require more time consumption in both training and classification. It is visible that KNN could normally reach the smallest time consumption among all classifiers, whereas the time consumption of SVM is very close to KNN. There is no significant difference between KNN and SVM.

Our results demonstrate that SVM can achieve the best detection accuracy among all classifiers, and can make a good balance between accuracy and time consumption. It is worth noting that SVM can help reduce the required number of labeled alarms as compared with the results in previous work [11].

Table 1. Some parameter settings in the evaluation.

Parameters	Value	Description
μ_1	15/day	Arrival rate for challenges
μ_2	5/day	Arrival rate for queries
λ	0.9	Forgetting factor
r	0.8	Trust threshold
$T_{dir,initial}$	0.5	Trust value for newcomers
m	10	Lower limit of received feedback
$k1$	5	Satisfaction levels
$k2$	10	Intrusion sensitivity levels

4.2 Evaluation in a Practical Environment

It is found that most existing trust management models have not been studied in a real network. In this part, we therefore aim to evaluate our enhanced trust management scheme in a practical CIDN environment by collaborating with an IT organization.

There are up to 71 nodes in this CIDN environment, and our trust management model was implemented with the help of security administrators from the participating organization due to privacy concerns. In this evaluation, we mainly consider two typical insider attacks like newcomer attack and betrayal attacks, as compared with the performance of DSOM trust model [3] and challenge-based trust model [4]. These two are the most relevant approaches to our work. We adopted the same satisfaction mapping method in [11].

To facilitate the comparison, similar to [11], we adopt that each *challenge* is comprised of 5 alarms ($c = 5$) and each *query* contains 50 alarms ($q = 50$). Some parameters are summarized in Table 1.

Defending Against Newcomer Attack and Betrayal Attack. The type of attack (also called re-entry attack) indicates a situation where a malicious node tries to register as a new user to erase its bad record. By contrast, betrayal attack is a major type of insider attacks, in which a trusted node (with high reputation) turns into a malicious node, i.e., behaving harmfully to the network. In this part, we investigate the performance of our trust management model against both newcomer and betrayal attack.

In practice, cyber-criminals often launch a newcomer attack to leverage the reputation, and then conduct a betrayal attack when the node obtains high reputation. After the trust values become stable in the network, we randomly selected 5 nodes in collaboration with security administrators, to conduct a betrayal attack from the 51st day, by sending malicious packets and false alarm rankings. The results of nodes' reputation under different trust models are shown in Fig. 5 and Fig. 6, respectively. We discuss the main observations as below.

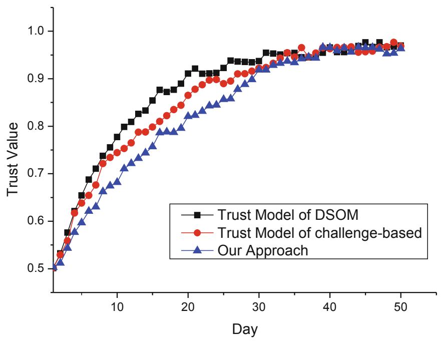

Fig. 5. The trust value of newcomers under different trust models.

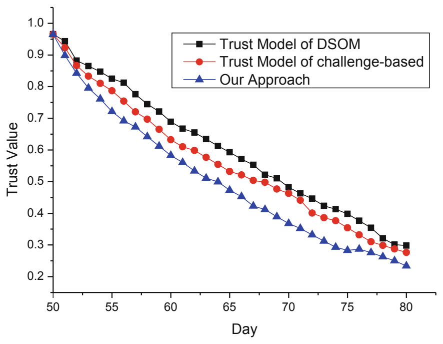

Fig. 6. The trust value of malicious nodes under betrayal attack.

- In our network settings, new nodes can become trusted only by increasing its trust values above the threshold of 0.8 (see Table 1); otherwise, it cannot affect the trust evaluation and alarm aggregation process. According to Fig. 5, it is found that the nodes under DSOM and challenge-based trust model could increase their reputation faster than our approach, i.e., our approach requires 4 days and 8 days more in comparison with challenge-based and DSOM model, respectively. This indicates that our approach is relatively less vulnerable to newcomer attack.
- Under betrayal attack, when a node becomes malicious, Fig. 6 shows the trust values under different trust models. It is visible that challenge-based trust model could outperform DSOM model by decreasing the trust value of malicious nodes faster. This is because challenge-based approach employed a forgetting factor. In comparison, our approach could reduce malicious nodes' reputation faster than the other two approaches. This is mainly because our approach applies intrusion sensitivity to emphasize the impact of expert nodes.

Overall, the results demonstrate that our trust management scheme can outperform the other two similar approaches by decreasing the trust values of malicious nodes faster. The main reason is that our approach applies *intrusion sensitivity* to highlight the impact of expert nodes. In this case, our approach is more sensitive to malicious behavior and more robust against insider attacks like betrayal attack. Our observation is also confirmed by the security administrators from the participating organization after repeating the experiments five times.

5 Conclusion

In this work, we advocate the effectiveness of sensitivity-based trust management model and develop an engineering way to automatically allocate the sensitivity values by using a support vector machine (SVM) classifier. In the evaluation, we compare the SVM classifier with three typical supervised classifiers in value allocation, and found that SVM can provide better accuracy and make a better balance between accuracy and time consumption than other classifiers. We further investigate our trust management model in a real network environment by collaborating with an IT organization. Our results demonstrate that our model can reach better detection performance than similar approaches under both newcomer and betrayal attack, by reducing the trust values of malicious nodes faster.

Acknowledgments. This work was partially supported by National Natural Science Foundation of China (No. 61802077).

References

1. Auria, L., Moro, R.A.: Support vector machines (SVM) as a technique for solvency analysis. DIW Berlin Discussion Paper no. 811 (2008)
2. Bao, F., Chen, I.R., Chang, M., Cho, J.H.: Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection. IEEE Trans. Netw. Serv. Manage. **9**(2), 169–183 (2012)
3. Duma, C., Karresand, M., Shahmehri, N., Caronni, G.: A trust-aware, P2P-based overlay for intrusion detection. In: Proceedings of DEXA Workshop, pp. 692–697 (2006)
4. Fung, C.J., Baysal, O., Zhang, J., Aib, I., Boutaba, R.: Trust management for host-based collaborative intrusion detection. In: De Turck, F., Kellerer, W., Kormentzas, G. (eds.) DSOM 2008. LNCS, vol. 5273, pp. 109–122. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87353-2_9
5. Fung, C.J., Zhang, J., Aib, I., Boutaba, R.: Robust and scalable trust management for collaborative intrusion detection. In: Proceedings of IM, pp. 33–40 (2009)
6. Li, J., Li, R., Kato, J.: Future trust management framework for mobile ad hoc networks. IEEE Commun. Mag. **46**(2), 108–114 (2008)
7. Liu, X., Zhu, P., Zhang, Y., Chen, K.: A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure. IEEE Trans. Smart Grid **6**(5), 2435–2443 (2015)
8. Li, W., Meng, W., Kwok, L.F.: Enhancing trust evaluation using intrusion sensitivity in collaborative intrusion detection networks: feasibility and challenges. In: Proceedings of the 9th International Conference on Computational Intelligence and Security (CIS), pp. 518–522 (2013)
9. Li, W., Meng, W., Kwok, L.-F., Ip, H.H.S.: PMFA: toward passive message fingerprint attacks on challenge-based collaborative intrusion detection networks. In: Chen, J., Piuri, V., Su, C., Yung, M. (eds.) NSS 2016. LNCS, vol. 9955, pp. 433–449. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46298-1_28
10. Li, W., Meng, W.: Enhancing collaborative intrusion detection networks using intrusion sensitivity in detecting pollution attacks. Inf. Comput. Secur. **24**(3), 265–276 (2016)
11. Li, W., Meng, W., Kwok, L.F., Ip, H.H.S.: Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model. J. Netw. Comput. Appl. **77**, 135–145 (2017)

12. LIBSVM Tools: Multi-label classification. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/multilabel/
13. Meng, Y., Kwok, L.F.: Adaptive false alarm filter using machine learning in intrusion detection. In: Wang, Y., Li, T. (eds.) Practical Applications of Intelligent Systems. AINSC, vol. 124. Springer, Berlin (2011). https://doi.org/10.1007/978-3-642-25658-5_68
14. Meng, Y., Li, W., Kwok, L.: Evaluation of detecting malicious nodes using bayesian model in wireless intrusion detection. In: Lopez, J., Huang, X., Sandhu, R. (eds.) NSS 2013. LNCS, vol. 7873, pp. 40–53. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38631-2_4
15. Paola, J.D., Schowengerdt, R.A.: A detailed comparison of backpropagation neural network and maximum-likelihood classifiers for urban land use classification. IEEE Trans. Geosci. Remote Sens. **33**(4), 981–996 (1995)
16. Qin, Z., Jia, Z., Chen, X.: Fuzzy dynamic programming based trusted routing decision in mobile ad hoc networks. In: Proceedings of the 5th IEEE International Symposium on Embedded Computing (SEC), pp. 180–185 (2008)
17. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Commun. ACM **43**(12), 45–48 (2000)
18. Roesch, M.: Snort: lightweight intrusion detection for networks. In: Proceedings of Usenix Lisa Conference, pp. 229–238 (1999)
19. Scarfone, K., Mell, P.: Guide to intrusion detection and prevention systems (IDPS). NIST Special Publication 800–94, Feburary 2007
20. Shamshirband, S., Anuar, N.B., Kiah, M.L.M., Patel, A.: An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique. Eng. Appl. Artif. Intell. **26**(9), 2105–2127 (2013)
21. Snort, Homepage. http://www.snort.org/
22. Vasilomanolakis, E., Karuppayah, S., Muhlhauser, M., Fischer, M.: Taxonomy and survey of collaborative intrusion detection. ACM Comput. Surv. **47**(4), 55 (2015)
23. Wu, Y.S., Foo, B., Mei, Y., Bagchi, S.: Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS. In: Proceedings of ACSAC, pp. 234–244 (2003)
24. Yuan, Y., Shaw, M.J.: Induction of fuzzy decision trees. Fuzzy Sets Syst. **69**(2), 125–139 (1995)