An Intrusion Detection System Based on Machine Learning for CAN-Bus

Daxin Tian[1,3,4], Yuzhou Li[1,3,4], Yunpeng Wang[1,3,4(✉)], Xuting Duan[3], Congyu Wang[3], Wenyang Wang[2], Rong Hui[2], and Peng Guo[2]

[1] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, XueYuan Road No. 37, Beijing 100191, China
ypwang@buaa.edu.cn
[2] China Automotive Technology and Research Center, Automotive Engineering Research Institute, East Xianfeng Road No. 68, Tianjin 300300, China
[3] Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems and Safety Control, School of Transportation Science and Engineering, Beihang University, XueYuan Road No. 37, Beijing 100191, China
[4] Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Si Pai Lou. 2, Nanjing 210096, China

Abstract. The CAN-Bus is currently the most widely used vehicle bus network technology, but it is designed for needs of vehicle control system, having massive data and lacking of information security mechanisms and means. The Intrusion Detection System (IDS) based on machine learning is an efficient active information security defense method and suitable for massive data processing. We use a machine learning algorithm—Gradient Boosting Decision Tree (GBDT) in IDS for CAN-Bus and propose a new feature based on entropy as the feature construction of GBDT algorithm. In detection performance, the IDS based on GBDT has a high True Positive (TP) rate and a low False Positive (FP) rate.

Keywords: CAN-Bus · Information security · IDS · Machine learning
GBDT · Entropy · Detection performance

1 Introduction

CAN-Bus (Controller Area Network Bus) [1, 2] is a kind of field bus and the most widely used vehicle bus network technology. Now, the vehicle is equipped with a large number of electronic equipment, in addition to the basic electronic control, media systems, as well as intelligent advanced auxiliary driving system, even that mobile phones and other intelligent devices connect to the infotainment system, these systems and devices will be from the car CAN-Bus to obtain data [3]. With the rapid progress of the automotive industry and the Internet, in the near future, Internet technology will be applied to every car and the electronic devices, intelligent information systems are likely to become hackers inbound vehicle network system approach [4]. Vehicle information security [5] is not only related to data confidentiality authenticity and integrity, but also related to traffic safety, which is directly related to human life and

Y. Chen and T. Q. Duong (Eds.): INISCOM 2017, LNICST 221, pp. 285–294, 2018.
https://doi.org/10.1007/978-3-319-74176-5_25

property safety. Therefore, the well-designed security system is a very significant and urgent to every modern car especially the IoV (Intelligent Connected Vehicle).

In the field of information security protection, there are passive security measures and active security measures. Passive security includes data encryption [6, 7], security authentication [8], firewall [9] and others. Active security is mainly based on Intrusion Detection System (IDS) [10] which is the essence of greatly much data, behavior analysis and detection in network, so as to find abnormal network behavior process. There are much data which contains ECUs' conditions, latencies, and behaviors in CAN-Bus, IDS is very suitable for a real-time security protection of CAN-Bus.

Vehicle information security is closely relate to the life and property safety of drivers and passengers. There are several researches to do works on IDS for vehicle CAN-Bus. Paper [11, 12] proposes a rate-based algorithm to detect the anomaly network behaviors, but it is too simple for complex CAN-Bus data, and the period selection of the algorithm is a difficult problem. Paper [13, 14] use entropy based message ID and frequency to be as the algorithm of IDS, but it can not detect the contents of the CAN-Bus message which is full of control commands, sensor information and other vehicle system key information. Paper [15] proposes a protocol-level security specifications for IDS in CAN-Bus, but the CAN-Bus protocol is the top secret of automobile enterprise and is almost impossible to get the protocol from every automobile enterprise of the industry. Therefore, the protocol-level is not versatility and unpractical.

In this paper, we propose an IDS based on a machine learning algorithm—Gradient Boosting Decision Tree (GBDT) [16] for CAN-Bus. GBDT is suitable for data detection which has great volume and few features. In feature engineering, we create a new entropy-based feature based on characteristics of CAN-Bus data to reflect the stability of the entire data, and that could be more robust. We get a very high True Positive (TP) rate and a quite low False Positive (FP) rate [17] in detection performance with a short time, and that means it has a great performance for detection and efficiency.

In this paper, we proposed the IDS based on GBDT for CAN-Bus in Sect. 2; detection performance based on real car CAN-Bus data and analysis is in Sect. 3; conclusion and outlook are in Sect. 4.

2 Gradient Boosting Decision Tree (GBDT) Algorithm in Intrusion Detection System (IDS) for CAN-Bus

Intrusion detection method is to design the network behavior classifier to distinguish the data set, simulation or network of normal and abnormal data, in order to achieve the alarm function of attract behaviors in CAN-Bus data. Machine learning can learn the existing intrusion or normal mode, the characteristics of the network packet probability deduction or fuzzy matching, so that unknown intrusion, in order to improve the intrusion detection of adaptive.

2.1 Regression Decision Tree (DT)

Decision tree [18] is a supervised learning model, which expresses the logical relationship between attributes and results in a tree diagram, mainly used to solve the

problem of classification and regression. In this paper, we use regression decision tree to be as the base algorithm.

In a given dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, we assume that the input space is divided into M regions $R_1, R_2, \ldots R_J$, each unit has a fixed output value c_m, and the regression decision tree model is:

$$y = f(x) = \sum_{j=1}^{J} c_j I(x \in R_j) \tag{1}$$

The prediction error of the training data set is:

$$\sum_{x_i \in R_j} (y - f(x_i))^2 \tag{2}$$

Here R_m is the region, if x_i is belong to R_m, the value is c_m, $I()$ is an indicator function that returns 1 when the formula inside parentheses is right, otherwise it returns 0. i and j are count constants.

2.2 Boosting Decision Tree

The boosting algorithm [19] is a method of integrating several classifiers into a classifier, it can be expressed as:

$$f(X) = w_0 + \sum_{j=1}^{J} w_j \Phi_j(X) \tag{3}$$

Here w is weight, Φ is the set of weak classifier (regression classification), in fact, is an additive model (the linear combination of primary functions). j is count constant.

The boosting decision tree is an iterative multiple regression tree to make a common decision. When the squared error loss function is used, each tree of regression tree learns the conclusion and residuals of all previous trees and fits to get a current residual regression tree. The meaning of residuals is as follows:

$$r = y - f \tag{4}$$

Here r is residual, y is true value and f is prediction value.

Thus, given the current model $f_{m-1}(x)$, it is only necessary to simply fit the residuals of the current model. Now boosting algorithm for decision tree is described as follows:

Boosting Decision Tree Algorithm:

Input: the regression decision tree $f(x)$.
Output: boosting decision tree $f_M(x)$.
Step1 initialize the decision model $f_0(x) = 0$;
Step2 for m : 1 to M do begin

Step3 calculate the residuals:

$$r_{im} = y_y - f_{m-1}(x_i), i = 1, 2, \ldots N(1);$$

Step4 fit r_{mi} to get a regression decision tree $T(x; \varnothing m)$;
Step5 update boosting decision tree:

$$f_m(x) = f_{m-1}(x) + T(x; \varnothing m) \tag{5}$$

Step6 get boosting decision tree:

$$f_M(x) = \sum_{m=1}^{M} T(x; \varnothing m) \tag{6}$$

Step7 end;

Here r_{im} is the residuals of m.th regression decision tree. $T(x; \varnothing m)$ is a regression decision tree about r_{mi}. i and m are count constants.

2.3 Gradient Boosting Decision Tree (GBDT)

The boosting decision tree uses the additive model and the forward stepwise algorithm to realize the optimization process of learning. When the loss function is a square loss or an exponential loss, the optimization of each step is very simple, such as the square loss function in residual regression tree (Table 1).

Table 1. Gradients for commonly used loss funictions.

Setting	Loss function	Gradient
Regression	$\frac{1}{2}[y_i - f(x_i)]^2$	$y_i - f(x_i)$
Regression	$\lvert y_i - f(x_i)\rvert$	$sign[y_i - f(x_i)]$
Regression	Huber	$y_i - f(x_i)$ for $\lvert y_i - f(x_i)\rvert \le \delta_m$ $\delta_m sign[y_i - f(x_i)]$ for $\lvert y_i - f(x_i)\rvert > \delta_m$ where $\delta_m = a\text{th} - \text{quantile}\{\lvert y_i - f(x_i)\rvert\}$
Classification	Deviance	kth component: $I(y_i = \varsigma_k) - p_k(x_i)$

But for the general loss function, often each step is not so easy to optimize, as in the table above the absolute value loss function and Huber loss function. In response to this problem, Freidman [16] proposed a gradient boosting algorithm: using the declining method of the steepest descent, that is, using the negative gradient of the loss function of the current model as an approximation of the residuals of the lifting tree algorithm in the regression problem, fitting a regression tree (Table 1).

The GBDT algorithm as follow:

Gradient Boosting Decision Tree Algorithm:

Input: the model $f(x)$.
Output: GBDT $\hat{f}(x)$.
Step1 initialize $f_0(x)$

$$f_0(x) = \arg\min_{y} \sum_{i=1}^{N} L(y_i, \gamma) \tag{7}$$

Step2 for $m : 1$ to M:

(a) for $i = 1, 2, \ldots, N$ compute

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}} \tag{8}$$

(b) fit a regression tree to the targets r_{im} giving terminal regions $R_{jm}, j = 1, 2, \ldots J_m$;
(c) for $j = 1, 2, \ldots J_m$ compute

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma) \tag{9}$$

(d) update $f_m(x)$

$$f_m(x) = f_{m-1}(x) + \sum\nolimits_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}) \tag{10}$$

Step3 output $\hat{f}(x) = f_M(x)$;
Step4 end;

Here r_{im} is the residuals of m.th regression decision tree. $L()$ is loss function, R_{jm} is the region m of tree j and γ_{jm} is the value of R_{jm}. i, m and j are count constants.

GBDT is one of the most popular machine learning algorithms that can handle various types of data flexibly and efficiently, and is suitable for almost any classification and regression problem. We use GBDT as the classification for IDS for its great efficiency, robustness and characteristics of CAN-Bus data in this paper.

2.4 Feature Engineering for CAN-Bus Data

There is a significant characteristic in CAN-Bus data and few features (about 8) in data. Obviously, too few features can not show the complexity of the data, will affect the IDS detection performance [20]. In this paper, we artificially construct an entropy-based [21] feature based on the ID and time of the data, it reflects the stability of the entire CAN-Bus data and conducive to the detection of abnormal behavior.

We let the random variable X, the all possible results of X are: $x_1, x_2, \ldots x_n$, the probability of each result is $p_1, p_2, \ldots p_n$, the entropy is:

$$H(X) = -\sum_{i=1}^{n} p_i \log_b p_i$$
$$0 \leq H(X) \leq \log |X| \tag{11}$$

In this paper, ID and time of CAN-Bus data reflects the whore stability, and we select period $T_i = 0.5s$, $i = 1, 2 \ldots, n$ for the entropy-based algorithm:

$$H_{T_i}(can\_ID\_j) = p_j \ln(p_j) \Big/ \sum_{j=1}^{N} p_j \ln(p_j) \tag{12}$$

Here N is the number of different ID in period T_i, p_i is proportion of one CAN_ID in all CAN-IDs in period T_i. t is a constant.

Since we get the new feature $H_{T_i}(can\_ID)$ in feature construction for CAN-Bus data and it will get better performance for IDS.

2.5 The Processes of the Novel IDS Based on GBDT for CAN-Bus

The core of IDS is the classification, and we use GBDT algorithm to classify the CAN-Bus data. The classification contains different tress for every ID and the general processes as follows (Fig. 1):

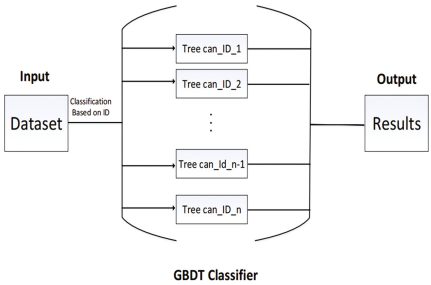


Fig. 1. The process of general GBDT classifier has many decision trees based on can_ID and we use dataset to be an input and we get the classification results.

We divide the IDS into two processes: in the train process, we construct new entropy-based feature and give every message a label for distinguishing between the normal and the abnormal, and get the known CAN-Bus behavior data and marked data. And then Preprocesses: discretization, feature extraction. Finally, we use the train set for GBDT training; in the test process, the Test Set includes unmarked CAN-Bus behavior data, and then discretization, and finally we use GBDT classifier to get behavior classification results. The processes of the IDS for CAN-Bus as follows (Fig. 2):

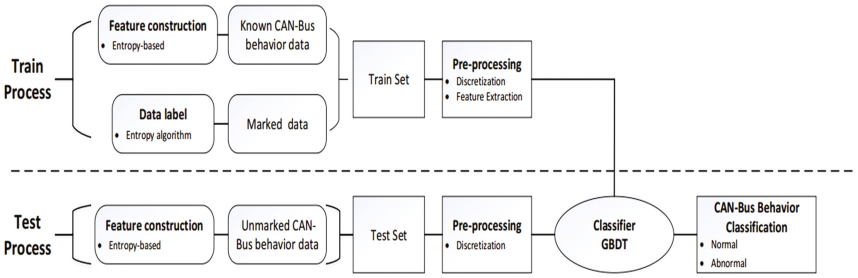


Fig. 2. The whole process: Train Process and Test process, the feature construction is in both train and test process, and data label is only in train process for marking data.

3 Detection Performance and Analysis

3.1 Dataset for Detection

In this paper, the dataset of detection is from a real domestic car—Alsvin CHANA. It contains 750,000 messages and the CAN-Bus speed is 250 kbits/s. When we collected data, the car was in a low speed and normal conditions for driving safety. Because the data is from real car and it may involve information security privacy and related legal issues, we make mosaics on some sensitive information in as follows (Fig. 3):

Abs Time(Sec)	Rel Time (Sec)	Status	Er	Tx	Description	Network	Node	PT	Trgt	Src	B0	B1	B2	B3	B4	B5	B6	B7	B8
59.39907855	0	2.9E+14	F	F		HS CAN		308	F	F		1	3					0 9C	D3
59.39931673	0.00023818	67371008	F	F		HS CAN		311	F	F		1	40				6B		0
59.39954698	0.000230253	3.17E+14	F	F		HS CAN		312	F	F		1 2F						96 2F	3C
59.40538901	0.00584203	67371008	F	F		HS CAN		418	F	F		1	0				F		5
59.40563726	0.000248253	67371008	F	F		HS CAN		419	F	F		1	0					0	0
59.40908122	0.003443956	2.99E+14	F	F		HS CAN		308	F	F		1	3					0 9C	D3
59.4093194	0.00023818	67371008	F	F		HS CAN		311	F	F		1	40				B		0
59.40954977	0.000230372	67371008	F	F		HS CAN		312	F	F		1 2F						96 2F	3C
59.41538787	0.005838096	67371008	F	F		HS CAN		418	F	F		1	0				F		5
59.41829187	0.002903998	67371008	F	F		HS CAN		508	F	F		1	2					0	0
59.41917604	0.000884175	67371008	F	F		HS CAN		308	F	F		1	3					0 9C	D3
59.41941422	0.00023818	67371008	F	F		HS CAN		311	F	F		1	40				B		0
59.41964048	0.000226259	4.24E+14	F	F		HS CAN		312	F	F		1 2F						96 2F	2A
59.42538649	0.005746007	67371008	F	F		HS CAN		418	F	F		1	0				F		5
59.42910254	0.003716052	4.22E+14	F	F		HS CAN		308	F	F		1	3					0 9C	D1
59.42934078	0.00023824	67371008	F	F		HS CAN		311	F	F		1	40				XB		0
59.42956692	0.00022614	67371008	F	F		HS CAN		312	F	F		1 2F						96 2F	2A
59.43538511	0.005818188	67371008	F	F		HS CAN		418	F	F		1	0				FF		5

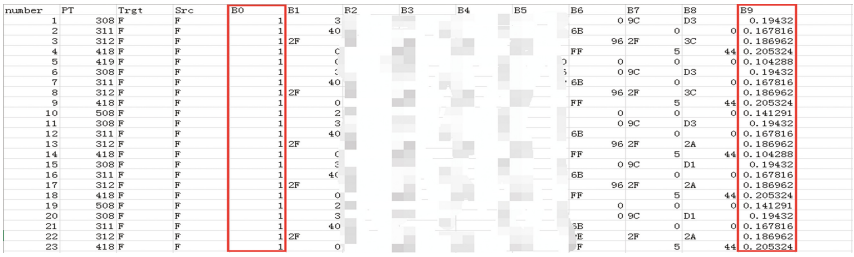
Fig. 3. The fig shows that there are 8 main features, ID, Abs Time and etc.

3.2 Feature Construction and Abnormal Samples

From Sect. 3.1, we find that every message has 8 main features (B1, B2, …, B8) and we artificially construct 2 new features: B0 and B9. B0 is to show that whether the message is normal or abnormal. B9 is an entropy-based feature which we describe in Sect. 2.4.

As we collect data when the real car is in a normal condition, we assume that the messages are all normal and the value of B0 should be 1. Usually, hacker will modify some features' value for tentative attacks, so we change the features' value randomly in the range of 0–255 to get abnormal messages and the value of B0 is 0 (Fig. 4).

We select 562,500 messages as the train set and number the ratio of normal messages and abnormal is 1:1, it means that there are 281,250 normal messages and



number	PT	Trgt	Src	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
1	308	F	F	1	3						0 9C	D3	0.19432
2	311	F	F	1	40					6B		0	0 0.167816
3	312	F	F	1	2F						96 2F	3C	0.186962
4	418	F	F	1	C					FF		5	44 0.205324
5	419	F	F	1	C					0	0	0	0 0.104288
6	308	F	F	1	3						0 9C	D3	0.19432
7	311	F	F	1	40					6B		0	0 0.167816
8	312	F	F	1	2F						96 2F	3C	0.186962
9	418	F	F	1	0					FF		5	44 0.205324
10	508	F	F	1	2					0		0	0 0.141291
11	308	F	F	1	3						0 9C	D3	0.19432
12	311	F	F	1	40					6B		0	0 0.167816
13	312	F	F	1	2F						96 2F	2A	0.186962
14	418	F	F	1	C					FF		5	44 0.104288
15	308	F	F	1	3						0 9C	D1	0.19432
16	311	F	F	1	40					6B		0	0 0.167816
17	312	F	F	1	2F						96 2F	2A	0.186962
18	418	F	F	1	0					FF		5	44 0.205324
19	508	F	F	1	2					0		0	0 0.141291
20	308	F	F	1	3						0 9C	D1	0.19432
21	311	F	F	1	40					3B		0	0 0.167816
22	312	F	F	1	2F					9E	2F	2A	0.186962
23	418	F	F	1	0					F		5	44 0.205324

Fig. 4. 2 new features in data and are shown in red boxes (Color figure online)

281.250 abnormal messages. In test set, we select 187,500 messages and the ratio of normal message and abnormal ones is 1:1, is similar to train set. So there are 93,750 normal messages and 93,750 abnormal ones.

3.3 Performance Analysis

In detection performance, the experimental platform environment is: Operation System: Windows 7 ultimate, CPU 3.00 GHz, RAM 8 GB, Hard Disk 500G; Programming tools: Spyder (Python 2.7), Dataset: Real vehicle CAN-Bus data (75% Train Set and 25% Test Set), the detection performance results as follows:

FP and TP are base index of an IDS and we could find that all FPs in Fig. 5 are higher than 95%. In fact, we use the weighted average to calculate the FP value and the accurate value of TP is 97.67% and the FP is 1.20%, that means the IDS based on GBDT has a great performance and could protect information security of CAN-Bus.

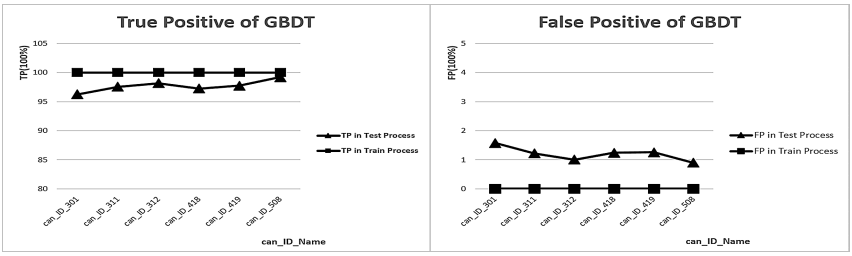


Fig. 5. The True Positive TP (TP) and False Positive (FP) of GBDT in IDS detection performance for CAN-Bus. We could find that the TPs are almost higher than 95% and FPs are lower than 1.5%.

4 Conclusion and Outlook

For the information security of CAN-Bus and traffic safety, we use IDS based on GBDT for CAN-Bus. With the popularity of intelligence connected vehicle, more and more devices and systems will connect CAN-Bus and get data from it. It is very reasonable to develop effective IDS to detect the attacks of hacker to ensure the

security. In this paper, the IDS we propose could detect the abnormal behaviors in massive CAN-Bus data and has a high True Positive and quite low False Positive in detection performance, that means the IDS based on GBDT has a great performance and could protect information security of CAN-Bus, even the life and property of drivers and pedestrians.

In this IDS, we have a lot of improvement in the performance of classification, in the future work, how to find more new useful and artificial features, and improve relationship of features between features, and other machine learning algorithm, which can further enhance the classifier's ability in detection performance for complex Internet of Vehicle (IoV).

Acknowledgments. This research was supported by the National Key Research and Development Program of China (2016YFB0100902).

References

1. Senn, S.: Analysis and application for CAN-bus controller integrated in AVR MCU, pp. 2661–2674 (1996)
2. Ricci, C.P.: Controller area network bus (2013)
3. Taha, A.E.M., Nasser, N.: Utilizing CAN-Bus and smartphones to enforce safe and responsible driving, pp. 111–115 (2015)
4. Guerrero-Ibanez, J.A., Zeadally, S., Contreras-Castillo, J.: Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. IEEE Wirel. Commun. **22**, 122–128 (2015)
5. Huang, C.H., Chen, H.Y., Huang, T.F., Tzeng, Y.Y., Li, P.Y., Wu, P.S.: A self-adaptive system for vehicle information security applications. In: IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, pp. 188–192 (2015)
6. Matsui, M.: The first experimental cryptanalysis of the data encryption standard. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 1–11. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_1
7. Biryukov, A., Cannière, C.D.: Data encryption standard (DES) (2005)
8. Lowe, G.: An attack on the Needham-Schroeder public-key authentication protocol. Inf. Process. Lett. **56**(3), 131–133 (1995)
9. Manner, J., Karagiannis, G., Mcdonald, A.: NSIS Signaling Layer Protocol (NSLP) for quality-of-service signaling. IETF **31**(2), 152–160 (2010)
10. Huang, M.Y., Jasper, R.J., Wicks, T.M.: A large scale distributed intrusion detection framework based on attack strategy analysis. Comput. Netw. **31**(23–24), 2465–2475 (1998)
11. Hoppe, T., Kiltz, S., Dittmann, J.: Security Threats to Automotive CAN networks – practical examples and selected short-term countermeasures. In: Harrison, Michael D., Sujan, M.-A. (eds.) SAFECOMP 2008. LNCS, vol. 5219, pp. 235–248. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87698-4_21
12. Cheng, K., Zhang, C.: Feature-based weighted Naive Bayesian classifier. Comput. Simul. **23** (10), 92–94 (2006)
13. Müter, M., Asaj, N.: Entropy-based anomaly detection for in-vehicle networks. In: Intelligent Vehicles Symposium, pp. 1110–1115 (2011)
14. Robnikšikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Mach. Learn. **53**(1), 23–69 (2003)

15. Larson, U.E., Nilsson, D.K., Jonsson, E.: An approach to specification-based attack detection for in-vehicle networks. In: Intelligent Vehicles Symposium, pp. 220–225 (2008)
16. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annal. Stat. **29**, 1189–1232 (2001)
17. Hamid, Y., Sugumaran, M., Journaux, L.: Machine learning techniques for intrusion detection: a comparative analysis. In: International Conference on Informatics and Analytics (2016)
18. Xu, M., Watanachaturaporn, P., Varshney, P.K., Arora, M.K.: Decision tree regression for soft classification of remote sensing data. Remote Sens. Environ. **97**, 322–336 (2005)
19. Takimoto, E., Maruoka, A.: Top-down decision tree learning as information based boosting. Theor. Comput. Sci. **292**, 447–464 (2003)
20. Iqbal, M.R.A., Rahman, S., Nabil, S.I., Chowdhury, I.U.A.: Knowledge based decision tree construction with feature importance domain knowledge. In: International Conference on Electrical & Computer Engineering, pp. 659–662 (2012)
21. Liang, J., Shi, Z., Li, D., Wierman, M.J.: Information entropy, rough entropy and knowledge granulation in incomplete information systems. Int. J. Gen. Syst. **35**(6), 641–654 (2006)