Anomaly-Based NIDS Using Artificial Neural Networks Optimised with Cuckoo Search Optimizer

K. Rithesh

Abstract Anomaly detection in network traffic is one of the major concerns for the researches and the network administrators. Presence of anomalies in network traffic could indicate a possible intrusion on the network, increasing the need for a fast and reliable network intrusion detection system (NIDS). A novel method of using an artificial neural network (ANN) optimised with Cuckoo Search Optimizer (CSO) is developed in this research paper to act as network monitoring and anomaly detection system. Two subsets of the KDD Cup 99 dataset have been considered to train and test our model, one of 2000 instances and the other of 10,000 instances, along with the complete dataset of 61,593 instances and I have compared the result with the BCS-GA algorithm and the fuzzy K-means clustering algorithm optimised with PSO in terms of precision, recall and f1-score, and the training time for the model with the selected database instances.

Keywords Network traffic · Anomaly detection · NIDS · Stream data analysis · Neural Network · Cuckoo Search Optimizer

1 Introduction

Intrusion in the field of the computer network is defined as any unauthorised activity on a system by exploiting its vulnerabilities, which could be improper security protocols or software bugs such as buffer overflow [1]. Uninterrupted services and data confidentiality are two of the important factors that determine the efficiency of a network system. General attack vectors like denial-of-service (DOS) attacks and buffer-overflow attacks, and protocol-specific attack vectors like ARP Poisoning and ICMP vulnerabilities tend to use the faults in the security system of the network for

K. Rithesh (✉)
Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, India
e-mail: 16co253.rithesh@nitk.edu.in

V. Sridhar et al. (eds.), *Emerging Research in Electronics, Computer Science and Technology*, Lecture Notes in Electrical Engineering 545,
https://doi.org/10.1007/978-981-13-5802-9_3

rendering the system passive and unresponsive or to access the privileges restricted to higher administrators [2].

Detecting a network intrusion depends on the defenders' level of understanding of the network and attacks associated with it. The various techniques used in intrusion detection can be classified into two categories: signature-based and anomaly-based intrusion detection.

Signature-based intrusion detection relies on the previously attained 'signature' of the network, which is a collection of features of the network, to classify whether the current network signature falls to the attack category or to the normal category [3]. While this technique is very well suited for detecting attacks if the attack signatures were previously available, it fails to identify any network with a new signature. And since the features of a network are always changing, this detection system does not guarantee accurate results.

Anomaly-based intrusion detection depends on the behaviour of the network to classify the current network. The model is pre-trained to identify different behaviours of the network and identify the current network [4]. When compared to signature-based intrusion detection technique, this technique is much slower, as it has to identify the behaviour of the current network, which involves analysis of various factors of the network. But it is very well suited for classifying new network behaviours as it does not depend on predefined properties once the model is trained.

Anomaly-based network intrusion detection system (NIDS) is the current focus of research, with the aim of building a faster and more accurate detection system. Statistical, clustering and classifying methods are commonly used to build the NIDS.

In this research, I have proposed a novel classifier-based anomaly detection for network intrusion—the conventional artificial neural networks optimised with the Cuckoo Search Optimizer algorithm.

Section 2 gives a brief review of the existing literature on the different models used for building NIDS. Section 3 gives the formal definition of the problem specifying some of the performance measures. Section 4 introduces the basic algorithms used to build the model proposed and the algorithm of the model proposed. The model is tested with the standard KDD Cup 99 dataset, and the results are analysed in Sect. 5. Section 6 concludes the research, briefing about the performance analysis of the model, and also lists out certain ideas which could help in improving the performance of the model as a part of future research work. The last section comprises the list of research papers referenced for the research undergone in this paper.

2 Literature Review

In this section, I have explored some of the existing machine learning tools and techniques for network anomaly detection.

The *Naive Bayes Classifier* described in [5] is a very efficient technique for anomaly detection if considered for fewer features, in accuracy and detection rate, but the performance drops with increase in the features (curse of dimensionality).

The *Genetic Network Programming* algorithm proposed in [6] has a pretty high detection rate when combined with fuzzy class-association-rule mining. It works well with both discrete and continuous values, but the algorithm is mainly used to extract the relevant rules of classification, for its high performance with less information available.

The combination of the two popular classifiers—*NBTree and Random Tree classifier*, as proposed in [7] is found to have a better result than the original random tree classifier, but the model takes more training time, and the performance is still under par with other hybrid models.

The innovation of using the *PSO-SVM classifier* for the network traffic anomaly detection, as described in [8] has paid off with good accuracy rate (~98%), for a small number of features. As the feature count increase, the accuracy drops sharply.

Time required to train the *Bayesian belief model* proposed in [9] is very less compared to other models and also gives a good precision rate for small feature count. But even with a slight increase in the features, the precision rate decreases rapidly.

The *MRMHC-SVM* algorithm [10] has a higher accuracy and robustness than the other intrusion detection methods and gives good result even if the dataset for training is small. But the time required for training the model is very high.

The popular *Neural Network with Backpropagation* [11] is highly adaptable to change in environment, and hence, training time is lesser than most of the models. But this method is seldom preferred because of its lower DR in comparison with other models.

A novel hybrid of *GA and Weighted KNN classifiers* is suggested in [12]. To build this model, I have extracted the features directly from the packets, and hence, it can be used as real-time NIDS. But though the accuracy for known attacks is very high, I could not obtain a good accuracy for unknown attacks.

Singh and Silakari [13] suggest the use of *KNN Classifier* for anomaly detection, which gained high accuracy rate when trained with a large dataset and with lesser features. But as the feature count increases, the accuracy drops by a large value.

A combination of *Decision Tree and SVM algorithm*, as proposed in [14] gives high accuracy of detection for specific attacks like DOS and Probing attacks, but is a poor algorithm for U2R and R2L attack detection. Also, the rate of convergence to the minima and its adaptation to new conditions was slow.

The use of *Cluster Centre and Nearest Neighbour (CANN)* for anomaly detection was researched upon in [15]. Performance of CANN was found to be higher than the KNN and SVM classifiers, providing higher detection rates and accuracy and a lower false alarm rate for a smaller feature set. But the accuracy decreases drastically for larger feature set.

Use of clustering algorithms like *K-Means Clustering* was studied in [16]. Higher accuracy of detection is observed, along with lower false alarm rate even for a smaller dataset. But the model is difficult to implement due to its complex structure and took a long time to train.

One of the recently developed models, *Unsupervised Niche Clustering* (*UNC*) was also researched upon for anomaly detection [17]. This algorithm, which takes

advantage of the fuzzy analysis for detection, has a high accuracy and detection rate when coupled with principal component analysis (PCA). But due to its complex structure and high training time, the model is seldom used.

3 Problem Description

The main problem tackled in this research is to devise a new model to detect network intrusion based on anomaly detection.

Also, certain factors were to be defined with which I can analyse the performance of the model proposed and to compare the proposed model with some of the other existing algorithms.

Some of the criteria for analysing the algorithm are:

1. *Precision and Recall*: Efficiency of the model in classifying the network is an important factor in analysing the model. The model should have a high precision and recall rate for it to successfully classify the network traffic as a normal or an attack one. More on the precision and recall is mentioned in Sect. 6.
2. *Training time*: Most of the time invested in building a model is spent in training it. While training is an important and essential part of a model, the model is to be trained in lesser time. Some complex algorithms have high precision rate, but take more amount of time in training and result evaluation, and hence are seldom used in this field, where real-time classification and anomaly detection is expected.
3. *Dataset size*: The dataset available for training the model is usually limited, especially when the model is to detect various types of attacks. Hence, the model must be able to perform even when the dataset available for training is small.

These above parameters define the problem to be solved by current researches. Some of the models try to satisfy one or two of the above factors, but research is still going on to come up with a model to satisfy all the factors.

4 The Proposed Solution

There are several approaches to tackle the problem defined, including the statistical, clustering and classifying techniques.

In this paper, a supervised learning classification algorithm is chosen for anomaly detection for the following reasons [18]:

- The classification models have a definite boundary of division and hence make faster decisions over the other models.
- The unsupervised clustering algorithms make strong assumptions on the nature of distribution of the traffic, such as the anomalous data are farther away from the

centres than the normal data and clusters with smaller sizes are anomalous. Supervised algorithms make minimal assumptions, as the training data gives sufficient information.
- The statistical models are very quick in decision-making, but are very slow in adapting to new environments.

A combination of two algorithms, a Neural Network for classifying the data, and an optimizer—Cuckoo Search Optimizer algorithm is used in this research.

4.1 Artificial Neural Network (ANN)

An artificial neural network (ANN) is modelled as a collection of nodes called as artificial neurons [19], the main inspiration being the neurons and its network in the brain of living beings. Each connection between the neurons transport signals from one another.

Each neural network consists of neurons grouped into layers. Each layer performs a specific function, including the input layer where the data is fed in, and the output layer where the classification result is obtained. The signal at the connection between two neurons is real-valued, and the output of each neuron is calculated based on a nonlinear function on the inputs for the neurons [20]. The output of the neuron is fed to the input of the neurons of the next layer, and the same process is repeated, until the outer layer is reached.

Learning in ANN is based on the modification of the function operating on the input of each neuron, based on the feedback mechanism [21]. There are various optimizer algorithms for this purpose, of which I am using a hybrid of two popular algorithms, which is discussed in the later part.

A simple model of ANN is represented by 'weights' on each of its inputs and a function which performs an operation on the inputs and the weights, returning the output for the next layer [21]. So learning of such ANN model involves modifying the weights based on the penalty given to each of the neuron.

Advantages of using ANN [19]:

- Adaptive capability of the ANN model is higher compared to most of the other classification algorithms. This will enable the model to adapt to new datasets and give better results when trained and tested.
- The model can organise the information (data) it receives during training, hence forming more generalised classification boundaries, and hence improving the results.
- Parallelism can be achieved in ANN, thus reducing time consumed in training and testing the model (which is considerably high for any heuristic algorithm).
- Fault tolerance rate is high enough, so that some of the capabilities of the original model can be retained in the damaged one.

4.2 Cuckoo Search Algorithm

Cuckoo Search is one of the popular optimizer algorithms currently used in various domains of engineering. Inspired from the brood parasitism nature exhibited by the cuckoo birds, it combines the advantages of swarm optimisation and genetic algorithms [22, 23].

i. *Cuckoo Search*: In the algorithm, the various possible solutions for the optimisation problem are considered as the eggs in a nest. Our aim is to develop new cuckoos which replace less fit eggs (solutions) with more fit ones [24]. The following assumptions are made during the construction of the algorithm:

- A cuckoo lays only one egg at a time and chooses a random nest to lay the egg.
- The nest with the highest fit eggs will be considered for the next generation.
- The host bird will detect the cuckoo's egg with a probability p_a.

For generating new cuckoos, along with the local random walk, the Cuckoo Search algorithm uses a global explorative random walk, where the parameter p_a is used for switching between them. For the local random walk, the Heaviside function is used, while for the global walk, I will be using the Levy flight [24, 25].

(I) Local random walk: $x_i^{t+1} = x_i^t + s * H(p_a - \varepsilon) * \left(x_j^t - x_k^t\right)$, where

x_i^t	ith solution of the tth generation.
x_i^{t+1}	ith solution of the $t + 1$th generation.
$H(u)$	Heaviside function.
ε	A random number generated from a uniform distribution.
S	The step size.
x_j^t and x_k^t	random solutions of the tth generation.

(II) Global random walk: $x_i^{t+1} = x_i^t + \alpha * L(s, \lambda)$, where

x_i^t	ith solution of the tth generation.
x_i^{t+1}	ith solution of the $t + 1$th generation.
α	boundary parameter = 0.01 * (UB − LB), UB = Upper Bound, LB = Lower Bound.
$L(s, \lambda)$	Levy flight function.

iii. *Levy flight*: Levy flight is defined as the random walk where the step length is calculated using the Levy distribution [25]. But the Levy flight can be approximated to:

(III) $L(s, \lambda) = \frac{1}{power(s, \lambda+1)}$, where

s	step length given by Mantegna's Algorithm
λ	Levy index

ii. *Step size*: For calculating the step size, I have used Mantegna's Algorithm [25].

(IV) $s = \frac{u}{v^{(1+\lambda)}}$, where

u normal distribution N(0, σ_u^2)
v normal distribution N(0, 1)

$$\sigma_u = \frac{\Gamma(1+\lambda) * \sin(\pi * \lambda/2)}{\Gamma((1+\lambda)/2) * \lambda * power(2, (\lambda+1)/2)}$$

ALGORITHM FOR CUCKOO SEARCH OPTIMIZER:

1. Initialise a random population of N host nests.
2. ***do****:*
3. Form a cuckoo by Levy flights (say i) and evaluate its fitness F_i
4. Choose a nest among N (say j) and evaluate its fitness F_j
5. ***if*** $F_i > F_j$*:*
6. Replace nest j by the new solution i
7. ***else****:*
8. Select j as the new solution
9. ***end if***
10. Remove a fraction p_a *of the worse nests and form new nests using random walks (I / II).*
11. ***while*** *(maximum number of iterations) or (minimum tolerance value) not reached.*
12. Sort the nests (based on fitness values) and find the best nest.

4.3 Construction of ANN-CSO Model

I have considered Cuckoo Search Algorithm for the optimisation of neural networks used to classify the dataset. The advantages of using genetic algorithms (GA) and swarm optimisation algorithms are discussed in [26] and [27], respectively. Some of the advantages of using Cuckoo Search optimizer are [23]:

- The selection criteria used in CSO is similar to that of genetic algorithms and Harmony Search.
- The step length in CSO is heavy-tailed, which supports any step length, and so randomisation is more efficient.
- The number of parameters in CSO is lesser compared to GA or PSO, and so it can adapt to changing environment faster.

In the model proposed here, I will be using the neural networks as the nonlinear objective function to be optimised by the Cuckoo Search Algorithm. A population of 15–40 neural networks with random weights and bias values are created. I will then find the cost of the network by training the model with the given dataset. The fitness for the neural network will be the inverse of the cost of the network. Then each of the networks will be considered as a possible solution (an egg in the case of Cuckoo Search Algorithm) and then the optimizer algorithm will be executed to get the best solution, i.e., the neural network with the least cost.

5 Experimental Result

5.1 Dataset

For training our model, I have used **KDD Cup '99** dataset. The dataset (corrected form) contains 311,029 instances of data, each containing of about 39 characteristics based on network packet data and the network traffic behaviour at the time of data mining. It also classified the instances into 4 different classes of attacks:

- DOS attack (223,298 instances)
- R2L attack (25,722 instances)
- Probe attack (377 instances)
- U2L attack (39 instances).

And about 61,593 instances of normal traffic data.

The model is trained using two sets of training data, one set containing 10,000 instances of data divided into 7500 training and 2500 testing data, and the other set containing 2,000 instances divided into 1500 training and 500 testing data, along with the original complete dataset split into 46,000 training and 15,593 testing data. Also, for detecting the attacks based on anomalies, the instances with U2L attack and Probe attack behaviour are very less in the original data, which is further reduced in our selection of training data (problems with overfitting and outlier handling will affect the performance). Hence the instances are classified into three groups:

- Normal Instances
- DOS attack instances
- R2L attack instances

5.2 Performance of the Model

The performance of the model is analysed based on its precision, recall and F1-score. For calculating the precision and recall, I need to define some terms [28]:

True Positive: When the prediction is positive and so is the actual data.
True Negative: When the prediction is negative and so is the actual data.
False Positive: When the prediction is positive, but the actual data is negative, also called ***false alarm***.
False Negative: When the prediction is negative, but the actual data is positive, also called ***miss***.

Precision is defined as the ratio of true positive to the total positive prediction.

$$Precision = True\ Positive/(True\ Positive + False\ Positive)$$

Recall is defined as the ratio of true positive to the total actual positive data.

$$Recall = True\ Positive/(True\ Positive + False\ Negative)$$

F1-score is the harmonic mean of precision and recall.

For the purpose of analysing our model with the existing models, I am comparing the performance of the model with two other existing models for anomaly detection: BCS-GA algorithm proposed in [29] and fuzzy K-means classifier using PSO algorithm proposed in [30].

Table 1 gives the precision, recall and F1-score on training and testing the model using the dataset of 2000 instances, 10,000 instances and the complete database. Table 2 gives the comparative performance and Table 3 gives the training time of our model over the two models considered for comparative analysis.

Figures 1, 2 and 3 give the graphical representation of the performance recorded in Table 2.

Table 1 Performance of the proposed model when trained for 1500 dataset instances and tested for 500 instances, when trained for 7500 dataset instances and tested for 2500 instances, and when trained for 46,000 dataset instances and tested for 15,593 instances

Dataset size	2,000 (1500 training + 500 test)			10,000 (7500 training + 2500 test)			61,593 (46,000 training + 15,593 test)		
Nature	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Normal	0.76	0.87	0.81	0.98	0.98	0.98	0.99	0.98	0.98
DOS	0.97	0.99	0.98	0.99	1.00	0.99	0.98	0.99	0.98
R2L	1.00	0.39	0.56	0.97	0.77	0.86	0.96	0.85	0.90

Table 2 Comparison of the performance of our hybrid model with the PSO-K-means model and the BCS-GA model. All the models were trained on the same dataset of 10,000 instances

Nature	Model	Precision	Recall	F1-Score
Normal	PSO-K-means proposed in [30]	0.96	0.97	0.96
	BCS-GA proposed in [29]	0.98	0.99	0.98
	ANN-CSO proposed in this paper	0.98	0.98	0.98
DOS	PSO-K-means proposed in [30]	0.98	0.97	0.97
	BCS-GA proposed in [29]	0.99	1.00	0.99
	ANN-CSO proposed in this paper	0.99	1.00	0.99
R2L	PSO-K-means proposed in [30]	0.95	0.75	0.84
	BCS-GA proposed in [29]	0.97	0.79	0.87
	ANN-CSO proposed in this paper	0.97	0.77	0.86

Table 3 Comparison of the time taken to train the three models considered for the performance analysis. The models were trained on the same computational system with the same dataset

Model	1000 dataset size (s)	2000 dataset size (s)	10,000 dataset size (s)
PSO-K-means proposed in [30]	1123	2207	11092
BCS-GA proposed in [29]	1175	2320	11502
ANN-CSO proposed in this paper	1054	2085	10214

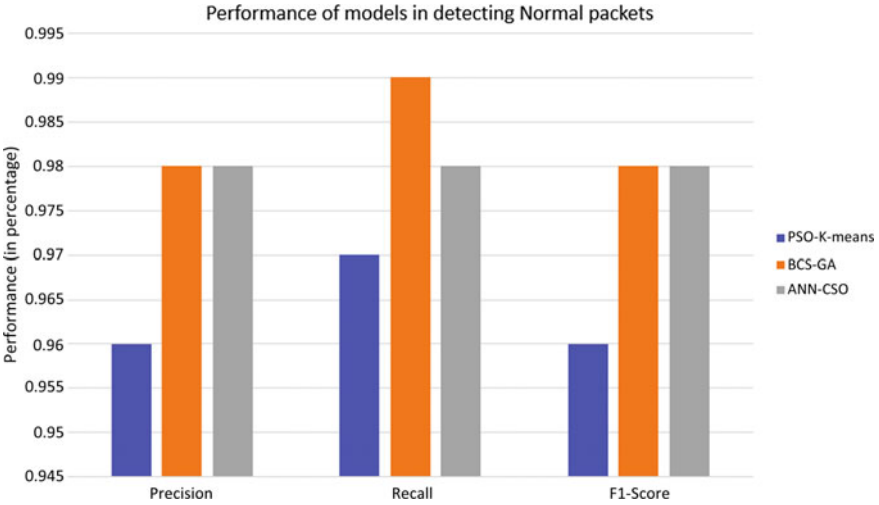

Fig. 1 Column graph compares the performance of the three models in the detection of normal packets

5.3 Analysis of Performance

From the results of training and testing the model against the dataset chosen, I can analyse the model chosen. The precision and recall for smaller dataset (2000) is low, as shown in Table 1. This is due to overfitting and insufficient handling of outliers. But the performance finds a significant increase when the model is trained for larger dataset (10,000). Also, there is no significant improvement in the performance as the dataset instances is increased from 10,000 to the complete dataset (61,593).

Table 2 suggests that the ANN-CSO model have a better precision and recall than the PSO-K-means model for all the three classes. Also, the precision and recall for normal and DOS attack detection is higher than that for R2L attack. This is due to the relatively smaller dataset for R2L attacks compared to the normal and the DOS attacks.

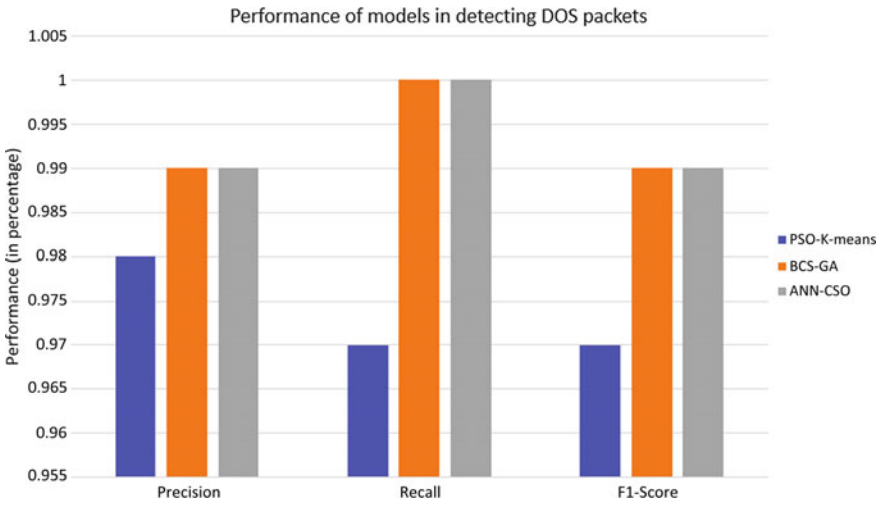

Fig. 2 Column graph compares the performance of the three models in the detection of DOS attack packets

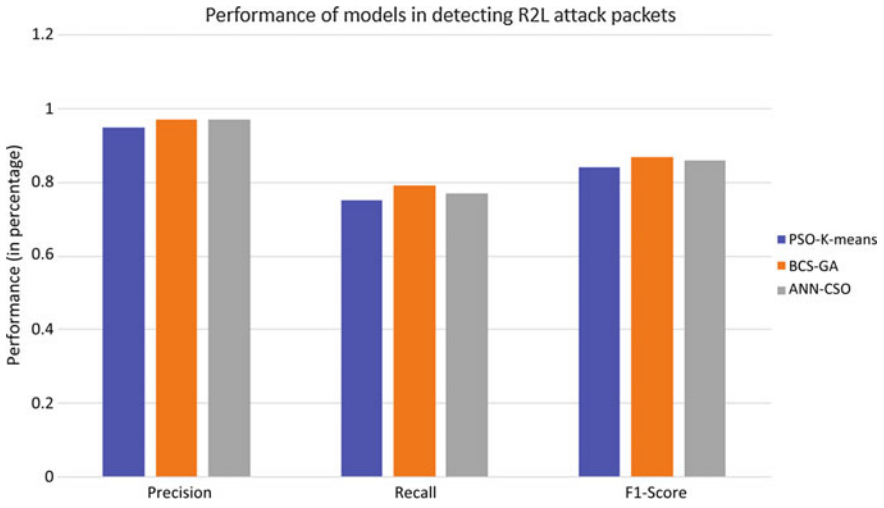

Fig. 3 Column graph compares the performance of the three models in the detection of R2L attack packets

The precision and recall of the ANN-CSO model and the BCS-GA hybrid model are very close to each other, though the hybrid proposed in [30] has slightly higher values. But the training time for the hybrid is more than the time for training the ANN-CSO model proposed in this paper.

6 Conclusion and Future Work

In this paper, I have presented a novel technique for network anomaly detection using artificial neural network optimised with cuckoo search algorithm. Our model has exploited the advantage of cuckoo search algorithm and the neural network classifier. Analysing the experimental results, I can conclude that the ANN-CSO model suggested in this paper has a higher precision and recall than the PSO-K-means model. Also, the model proposed takes significantly lesser training time compared to the BCS-GA hybrid model.

In our model, I have assumed that each nest in the cuckoo search algorithm has only one egg, and also each cuckoo lays only one egg. This restriction can be eased and allow many eggs in a single nest, and each cuckoo can lay many eggs based on different levy flights. This could allow for multiple solution optimisations, which could give better results than the single solution optimisation proposed in this paper, provided the step length and the Levy flight parameters are properly controlled. Such modifications can be made on the model, and the results can be analysed.

References

1. Yang H, Xie F Lu Y (2006) Research on network anomaly detection based on clustering and classifier. In: 2006 International conference on computational intelligence and security, Guangzhou, pp 592–597
2. Common Types of Network Attacks—Microsoft Docs
3. Holm H (2014) Signature based intrusion detection for zero-day attacks. In: 2014 47th Hawaii international conference on system sciences, Waikoloa, HI, pp 4895–4904
4. Zhang W, Yang Q, Geng Y (2009) A survey of anomaly detection methods in networks. In: 2009 International symposium on computer network and multimedia technology, Wuhan, pp 1–3
5. Almansob SM Lomte SS (2017) Addressing challenges for intrusion detection system using naive Bayes and PCA algorithm. In: 2017 2nd international conference for convergence in technology (I2CT), Mumbai, pp 565–568
6. Mabu S, Chen C, Lu N, Shimada K Hirasawa K (2011) An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. IEEE Trans Syst Man Cybern Part C (Appl Rev) 41(1):130–139
7. Kevric J, Jukic S, Subasi A (2016) An effective combining classifier approach using tree algorithms for network intrusion detection. Neural Comput Appl 1–8
8. Lei Y (2017) Network anomaly traffic detection algorithm based on SVM. In: 2017 International conference on robots and intelligent system (ICRIS), Huai'an, pp 217–220. https://doi.org/10.1109/ICRIS.2017.61
9. Thakong M, Wongthanavasu S (2007) Packet header anomaly detection using bayesian belief network. ECTI Trans Comput Inf Technol 3(1)

10. Li W, Duan M, Chen Y (2008) Network anomaly detection based on MRMHC-SVM algorithm. In: 2008 IEEE international multitopic conference, Karachi, pp 307–312
11. Al-Janabi STF, Saeed HA (2011) A neural network based anomaly intrusion detection system. In: 2011 Developments in e-systems engineering, Dubai, pp 221–226
12. Su M-Y (2011) Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers. Expert Syst Appl 38(4):3492–3498
13. Singh S, Silakari S (2009) An ensemble approach for feature selection of cyber attack dataset. Int J Comput Sci Inf Secur P12-(IJCSIS) 6(2)
14. Peddabachigari S, Abraham A, Grosan C, Thomas J (2007) Modeling intrusion detection system using hybrid intelligent systems. J Netw Comput Appl 30(1):114–132
15. Lin WC, Ke SW, Tsai CF (2015) CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl-Based Syst
16. Li H (2010) Research and implementation of an anomaly detection model based on clustering analysis. In: 2010 International symposium on intelligence information processing and trusted computing, Huanggang, pp 458–462
17. Leon E, Nasraoui O Gomez J (2004) Anomaly detection based on unsupervised niche clustering with application to network intrusion detection. In: Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753), vol 1, pp 502–508
18. Jidiga GR, Sammulal P (2014) Anomaly detection using machine learning with a case study. In: 2014 IEEE international conference on advanced communications, control and computing technologies, Ramanathapuram, pp 1060–1065
19. Callegari C, Giordano S, Pagano M (2014) Neural network based anomaly detection. In: 2014 IEEE 19th international workshop on computer aided modeling and design of communication links and networks (CAMAD), Athens, pp 310–314
20. Han S-J, Cho S-B (2005) Evolutionary neural networks for anomaly detection based on the behavior of a program. IEEE Trans Syst Man Cybern Part B (Cybernetics) 36(3):559–570
21. Andropov S, Guirik A, Budko M, Budko M (2017) Network anomaly detection using artificial neural networks. In: 2017 20th conference of open innovations association (FRUCT), St. Petersburg
22. Naik M, Nath MR, Wunnava A, Sahany S, Panda R (2015) A new adaptive Cuckoo search algorithm. In: 2015 IEEE 2nd international conference on recent trends in information systems (ReTIS), Kolkata, pp 1–5
23. Majumdar D, Mallick S (2016) Cuckoo search algorithm for constraint satisfaction and optimization. In: 2016 Second international conference on research in computational intelligence and communication networks (ICRCICN), Kolkata, pp 235–240
24. Zhao P, Li H (2012) Opposition-based Cuckoo search algorithm for optimization problems. In: 2012 Fifth international symposium on computational intelligence and design, Hangzhou, pp 344–347
25. Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 World congress on nature and biologically inspired computing (NaBIC), Coimbatore, pp 210–214
26. Hamamoto AH, Carvalho LF, Sampaio LDH, Abrão T, Proença ML (2018) Network anomaly detection system using genetic algorithm and fuzzy logic. Expert Syst Appl 92:390–402
27. Ali MH, Al Mohammed BAD, Ismail A, Zolkipli MF (2018) A new intrusion detection system based on fast learning network and particle swarm optimization. IEEE Access 6:20255–20261
28. Koehrsen W (2018) Beyond accuracy: precision and recall—towards data science
29. Ghosh P, Jha S, Dutta R, Phadikar S (2018) Intrusion detection system based on BCS-GA in cloud environment. In: Shetty N, Patnaik L, Prasad N, Nalini N (eds) Emerging research in computing, information, communication and applications. ERCICA 2016. Springer, Singapore
30. Ensafi R, Dehghanzadeh S, Akbarzadeh TMR (2008) Optimizing fuzzy k-means for network anomaly detection using PSO. In: 2008 IEEE/ACS international conference on computer systems and applications, Doha, pp 686–693