A Comprehensive Survey of Machine Learning-Based Network Intrusion Detection

Radhika Chapaneri and Seema Shah

Abstract In this paper, we survey the published work on machine learning-based network intrusion detection systems covering recent state-of-the-art techniques. We address the problems of conventional datasets and present a detailed comparison of modern network intrusion datasets (UNSW-NB15, TUIDS, and NSLKDD). Recent feature-level processing techniques are elaborated followed by a discussion on supervised multi-class machine learning classifiers. Finally, open challenges are pointed out and research directions are provided to promote further research in this area.

1 Introduction

Cyber-attacks against individuals, businesses, and nations hit headlines almost every day as computers, mobile communication devices, IoT devices, and networks have become an integral part of life. Unfortunately, these devices are attacked by network intruders and cyber-criminals with malicious intent. Powerful security defense mechanisms involving intrusion detection systems are essential for detecting such cyber-attacks. Since the administrator cannot manually inspect each packet entering or leaving the network, it is desirable that the machine learns itself the behavior of incoming packets and detects any suspicious or anomalous behavior.

The aim of this paper is to present a comprehensive survey of machine learning-based network intrusion detection. There are multiple surveys previously published in this area: [1] focusses on four major categories (statistical, information theory, classification, and clustering), [2] covers various methods, systems, and

R. Chapaneri (✉) · S. Shah
Mukesh Patel School of Technology Management and Engineering,
NMIMS University, Mumbai, India
e-mail: radhika.chapaneri@nmims.edu

S. Shah
e-mail: seema.shah@nmims.edu

S. C. Satapathy et al. (eds.), *Smart Intelligent Computing and Applications*,
Smart Innovation, Systems and Technologies 104,
https://doi.org/10.1007/978-981-13-1921-1_35

validity measures of network anomaly detection, [3] presents various distance and similarities measures, and [4] compares available commercial NIDS. Even though all these surveys provide a valuable contribution, this paper covers various facets of machine learning-based NIDS focusing on recent trends in terms of datasets, feature analysis, and machine learning techniques. The **contributions** in this paper are (a) comparison of modern datasets with conventional datasets relative to current network scenarios and attacks, (b) recent features transformation and selection techniques are compared along with feature analysis of recent dataset, (c) overview of various machine learning techniques is used in the field of network intrusion detection in last 5 years, and (d) open challenges in this area are pointed out and research directions are suggested to tackle these challenges.

Section 2 provides an overview of intrusion detection systems, Sect. 3 presents comparative analysis of various datasets, Sect. 4 focuses on feature-level processing, and Sect. 5 surveys various machine learning techniques. Section 6 presents open challenges to be solved in this area followed by conclusions in Sect. 7.

2 Overview of Network Intrusion Detection System (NIDS)

Performing an intrusion or attack is a multistage process consisting of (a) reconnaissance, (b) vulnerability identification and scanning, (c) gaining access and compromising system, (d) maintaining access and creating backdoors, and (e) covering tracks.

Table 1 presents the characteristics of various intrusions; the detailed taxonomy of tools and attacks can be found in [5]. IDS is used to classify incoming data either as malicious or as normal traffic. There are three approaches for deployment of IDS: host-based (HIDS, machine-specific), network-based (NIDS, network-specific and its interfaces), and hybrid IDS [6]. The detection approaches can be categorized into signature-based and anomaly-based. In signature-based detection (also known as misuse detection) approach, a database of the malicious pattern (signature) is created; the matching algorithm is applied on incoming data, and if the pattern is matched with known signature, an alarm is raised. A major advantage of this approach is the low false positive rate (FPR), but it fails to detect novel zero-day attacks; also, the signature database is required to be constantly updated. In contrast, the anomaly-based approach creates a model (i.e., learns the characteristics) of normal behavior and detects deviations or anomaly stimulated due to the attack. With this approach, generalization of behavior can be achieved in the testing phase allowing identification of attack in real time. A drawback of this approach is that it suffers from high FPR and hence low detection accuracy considering modern attacks on network. Therefore, this area is growing and a challenging area of research.

Table 1 Network intrusion characteristics

Intrusions	Characteristics
Probe/ Reconnaissance	Attacker gathers information about the network, including number of machines, type of OS.
Exploit	Exploits known vulnerability of networks
Backdoors	Security mechanism is bypassed stealthily to access network data
U2R	Gain illegal access to root account to manipulate confidential resources
R2L	Gain local access as a user of a targeted machine remotely
Worms	Malicious code replicates itself to spread to other computers on the network
DoS/DDoS	Attacker disrupts the normal computing operation and makes the service unavailable to legitimate users

3 Comparative Analysis of Network Intrusions Datasets

Datasets play an important role in evaluating the performance of IDSs since they can be used for repeating experiments and validating new techniques. The authors in [8] have set several guidelines for creating NIDS datasets: (a) realistic network and traffic, (b) labeled dataset, (c) total interaction capture, and d) diverse intrusion scenarios. IDS datasets typically consist of following categories of features: (a) **basic features** which include features of TCP/IP connection, (b) **time-based features** which capture temporal characteristics of network data; for example, features that examine all connections having the same destination in last 2 s so that DoS and probe attack (which involves sending a huge amount of traffic data to the same host) can be easily discriminated and identified, and (c) **content features** which are extracted using domain knowledge to capture suspicious behavior in the data, for example unauthorized logged in as root or number of failed login attempts.

3.1 Conventional Datasets

The first benchmark dataset of IDS was KDD99 released by DARPA. Extensive test bed network was created, and different types of attacks were simulated and categorized into four groups: DOS, U2R, R2L, and Probe. The dataset consists of 41 features and a label to specify either normal or attack (and type of attack). Even though widely used, this dataset has inherent flaws as pointed out in [8]. To avoid the limitations of KDD99 dataset, NSL KDD dataset was created having the following advantages [9]: (a) Redundant records were removed to decrease classifier bias, (b) duplicate records were removed, (c) new train and test sets were created containing reasonable amount of samples for full set training and testing, and (d) a number of samples are inversely proportion to difficulty level of the particular attack category.

3.2 Modern Datasets

Since the complexity of network and types of attacks have changed over time, the NSL KDD dataset is therefore not relevant for the current evaluation of modern network scenarios. Hence, efforts are geared toward creating benchmark datasets that can capture modern attack traces either via simulation or via real-time network traffic. Recent datasets are discussed followed by a comparative analysis of two specific datasets with NSL KDD in Table 2.

Table 2 Comparison of benchmark and recent datasets (UNSW-NB15, TUIDS)

Parameters	NSL KDD	UNSW-NB 15	TUIDS
Year of creation	KDD99—1999 NSL KDD—2009	2015	2015
Modern network traffic and attacks	No	Yes	Yes
Type of network traffic	Simulated	Simulated	Real
Attack traffic generation	Simulated	IXIA perfect storm tool	Modern tools in test bed network
Duration of data collected	7 weeks	16 h and 15 h	7 days
Feature extraction tools	BroIDS	BroIDS, Argus, C# algorithms	Distributed feature extraction framework
Feature categories	Basic, content, time, connection	Basic, content, time, connection	Basic, content, time, connection
# Features	41	47	50
# Attack categories	4	9	2
Number of samples (training set)	**N**:67,343; **D**:45,927; **P**:11,656; **U**:52; **R**:995	**N**: 56,000; **F**: 18,184, **A**: 2,000; **B**: 1,746; **D**: 12,264; **E**: 33,393; **G**: 40,000; **R**: 10,491; **S**: 1,133; **W**: 130	**N**: 71,785; **D**: 42,592; **P**: 7550
Number of samples (testing set)	**N**:9,710; **D**:7,458; **P**:2,422; **U**:67; **R**:2887	**N**: 37,000; **F**: 6,062, **A**: 677; **B**: 583; **D**: 4,089; **E**: 11,132; **G**: 18,871; **R**: 3,496; **S**: 378; **W**: 44	**N**: 47,895; **D**: 30,613; **P**: 7757
Publicly available	Yes	Yes	Yes

N: Normal; **D**: DoS; **P**: probe; **U**: U2R; **R**: R2L; **A**: analysis; **B**: backdoor; **E**: exploits; **F**: fuzzers; **G**: generic; **R**: reconnaissance; **S**: shellcode; **W**: worms

(a) **UNSW NB15** [10]: This dataset represents modern low footprint attacks and was developed by Australian Center for Cyber Security (ACCS). IXIA perfect storm tool was used to create hybrid of modern network scenarios and nine classes of attack traffic. Since it contains information about new attack types updated continuously from CVE (dictionary of publicly known information security vulnerabilities), the dataset captures modern network intrusion characteristics. Two simulation scenarios were conducted, first for 16 h with 1 attack/sec and second for 15 h with 10 attacks/sec. Labeling of the dataset was done via the IXIA report. From network traces, 43 features were extracted that include basic (14), content (8), time (9), connection (7), and additional (5) features. The description of each feature is provided in [10], and the distribution of number of samples of each category is provided in Table 2. UNSW NB15 is a complex dataset [11] as it represents modern network and attack traffic and can be used for reliable evaluation.

Figure 1a illustrates the analysis of feature values of this dataset where all features are normalized via z-score. We observe that some features are redundant; for example, values of feature indices 14 to 19 are almost similar for all 10 categories and hence such features need to be discarded via some feature selection technique. Also, using boxplot analysis of Fig. 1b, we observe that there is a significant variation of feature values among the ten categories which will result in better discrimination for classifiers.

(b) **TUIDS** [12]: A test bed network was developed at Tezpur University consisting of 5 different networks by configuring 250 hosts across several VLANs, multiple switches, wireless controllers, and routers. Real-time network traffic was generated based on daily activities of students, faculties, system administrators, etc., and attack traffic was generated by launching attacks in test bed network using modern tools such as targa, nmap, brutessh. Network traffic of 7 days was captured using libpcap, and gulp and fast distributed framework

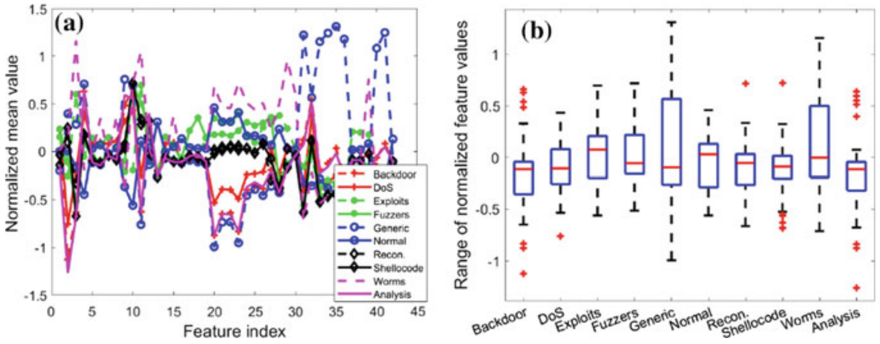

Fig. 1 Feature analysis of UNSW NB15 dataset: **a** normalized average 43 feature values for 10 categories and **b** boxplot showing feature variations of 10 categories

were used to extract 50 features consisting of basic (25), content (5), time (10), and connection (10)-based traffic features. Due to unavailability of training and test files, we could not do feature analysis of this dataset.

(c) There are few other datasets such as **CAIDA**, **LBNL,** and **ISCI,** but they have the problem of anonymity and payload is completely removed because of which there is less utility of these datasets to researchers. **KYOTO** dataset is made using Honeypots consisting of email and DNS traffic only. **IRSC** dataset contains both simulated and real attack data, but it is not publicly available.

(d) In [8], authors have designed a systematic approach to generate dynamic datasets which keeps on updating current traffic pattern and newer types of attacks. Profiles are developed that reflect an abstract representation of characteristics of attacks and their features and aims at reproducibility of experiments to simulate attack traffic.

(e) In [7], a tool-based method was developed to create a controlled and reproducible environment for generation of intrusion datasets. The dataset aims at generating known, similar, and new attacks.

From Table 2, we observe that the recent two datasets UNSW NB15 and TUIDS satisfy many guidelines sets in [8]. Since UNSW-NB15 benchmark dataset is publicly available, we recommend its use for further research in this area.

4 Feature-Level Processing of Network Intrusion Datasets

Typically, the data of network intrusion datasets is huge and high dimensional and all attributes may not be relevant in building the IDS model. Preprocessing of feature matrix **X** is a basic step for preparation of adequate feature vectors, wherein either normalization given by Eq. (1) (left) or standardization by z-score given by Eq. (1) (right) is performed, where $\mathbf{X}_{uw}$ is the uth feature value of $\mathbf{x}_u^{\mathrm{T}}$ corresponding to the wth sample, $N_{\min}$ and $N_{\max}$ are the desirable range of feature values (e.g., [0, 1]), and $\mathbf{x}_u^{\mathrm{T}}$ and s_{xu} are the mean and standard deviation of feature vector $\mathbf{x}_u^{\mathrm{T}}$.

$$\mathbf{X}_{uw}^{'} = \frac{\mathbf{X}_{uw} - \min\left(\mathbf{x}_u^T\right)}{\max\left(\mathbf{x}_u^T\right) - \min\left(\mathbf{x}_u^T\right)} \cdot \left(N_{\max} - N_{\min}\right) + N_{\min}\ ;\ \mathbf{X}_{uw}^{'} = \frac{\mathbf{X}_{uw} - \bar{\mathbf{x}}_u^T}{s_{\mathrm{x}u}} \qquad (1)$$

Next, to achieve the discriminative capability of classifiers, it is desirable to reduce the dimensionality of features either through feature transformation or through feature selection. In the former, techniques such as principal component analysis (PCA) and independent component analysis (ICA) are used due to which the resulting features are projected onto a transformed feature space, wherein only the most important feature dimensions are retained; however, by doing so, the interpretability of feature dimension is lost. In the latter approach of feature selection, redundant and irrelevant features are discarded due to which only the prominent features are retained which are interpretable. A feature x is relevant if its removal from the full feature set F leads to a decrease in classification performance, i.e., $\mathrm{P}(y = y'|F) > \mathrm{P}(y = y'|F\backslash\{x\})$ and $\mathrm{P}(y \neq y'|F) < \mathrm{P}(y \neq y'|F\backslash\{x\})$, where y is the correct label and y' is the predicted label. The task of feature selection is to find the optimal set of features represented by q^* given by Eq. (2), where F is the full feature set, $\Phi(F, q)$ is the set with selected features, q is a binary vector of length F which indicates whether a feature x is selected ($q_x = 1$) or not ($q_x = 0$), and $\mathbf{Q}$ is measure that describes the success of feature selection.

$$q^* = \arg\min_q \mathbf{Q}\left(y, y', \Phi(F, q)\right) \tag{2}$$

The feature selection evaluation strategies can be classified into filter, wrapper, and ensemble methods. For D-dimensional features, the possible number of feature sets is $2^D - 1$ which can be searched in either sequential, exhaustive or random manner. Filters evaluate feature sets using relevance criteria such as correlation, information gain without considering classifier performance. Wrappers evaluate feature sets by measuring the quality (e.g., accuracy) of classifier and thus often achieve better results than filters but have drawbacks of overfitting and time-consuming relative to filters. Ensemble methods are directly integrated into the classifier algorithm (e.g., decision trees) which can help improve the performance, but the drawback is that they cannot be applied to other classification algorithms.

Table 3 lists the state-of-the-art techniques used for feature-level processing of network intrusion datasets. PCA is widely used for extracting relevant information as it derives set of uncorrelated features from the set of correlated features. PCA sorts the principal components by its variance and selects eigenvectors with higher eigenvalues. But it may be possible that these eigenvectors are not discriminative, hence authors in [13] select projections with higher FDR values. Even after the

Table 3 Feature selection/transformation techniques for network intrusion detection

Paper	Method	Approach	Dataset used	#Features
[13]	Feature transformation	Principal component analysis with Fisher Discriminant Ratio (FDR)	NSL KDD	8
[14]	Feature transformation	Kernel principal component analysis	KDD99	–
[15]	Filter	Ensemble of filtered, consistency, correlation-based feature selection techniques	NSL KDD, Kyoto	21, 11
[16]	Filter	Combination of correlation coefficient and information gain	NSL KDD	13
[17]	Filter	Pearson correlation coefficient	NSL KDD	17
[18]	Feature representation	Cluster center and nearest neighbor	KDD99	–
[19]	Filter and wrapper	Combination of different filters and stepwise regression wrapper approach	NSL KDD	16

reduction of features to eight, they found that the accuracy is not reduced. As PCA can only extract linear information, authors in [14] have used kernel PCA which transforms the features in high-dimensional space due to which they found approximately 5% increase in detection rate as compared to using only PCA. In [15], irrelevant features are removed using ensemble of three methods: filtered, consistency, and correlation-based feature selection techniques due to which features are reduced by 48.78% for NSL KDD and 31.25% for Kyoto dataset. In [16], initially correlation-based feature selection (CFS) is applied resulting in 10 features, but since CFS does not guarantee selection of all relevant features, information gain is used to rank the features not selected by CFS and high-ranked features are considered resulting in 13 features. In [17], authors have used multilayer approach; in the first layer, Pearson correlation coefficient is applied between the features, and in the second layer, correlation coefficient is calculated between features selected by the first layer and the class followed by C4.5 used as a classifier, resulting in 1% accuracy improvement. In [18], original dataset is transformed into one-dimensional distance by summing up the distance between a specific data sample and cluster centers and distance between data sample and its nearest neighbor. In [19], multistage feature selection is done by a combination of filters and stepwise regression wrappers to remove the bias by any specific method. The cost of every feature is analyzed and eliminated 13 very costly features. It is observed that the number of prominent features is different for various techniques as the whole dataset is not considered for evaluation. Also, the dimensionality reduction techniques for modern datasets need to be explored.

5 Machine Learning Techniques for NIDS

Supervised classification techniques are widely used for learning and generalizing the behavior of each category, e.g., normal, DoS attack, fuzzer attack. Formally, a classifier is a map f: $\mathbf{X} \rightarrow Y$, where $\mathbf{X}$ is the processed feature matrix of dimensions N (number of samples) by D (feature dimensions), Y is the set of categories of dimension $N \times 1$ indicating the class of each input sample, and f is the multi-class classifier algorithm that assigns each input sample to a class. Several well-known algorithms such as SVM, decision trees (DT), Naïve Bayes (NB), artificial neural networks (ANN) are widely used for modeling IDSs. Recently, extreme learning machines (ELMs) have been used for IDS modeling due to its specific advantage that ELMs are faster to train and can handle multi-class classification inherently. To evaluate the performance of classifier, standard metrics such as accuracy, precision, recall, F-score, area under curve (AUC) of receiver operating characteristics (ROC) are used. In this section, we survey only notable recent works (2013–2017) as earlier works are detailed in previous surveys [2] [4].

In [20], semi-supervised classification using restricted Boltzmann machines (RBMs) is used to combine generative and discriminative approaches of machine learning; the experiments concluded that performance suffers when system is evaluated on a real network traffic instead of traces from training dataset. This indicates that transfer learning methodology should be used to improve and generalize IDS performance. In [21], *k*-best Bayesian network (BN) classifiers were averaged resulting in Bayesian model averaging (BMA) classifier performing significantly better (96% accuracy) compared to NB (92% accuracy) on NSL KDD dataset. In [15], instead of sampling the entire dataset, beta profiling is used to reduce size of dataset and an online sequential version of ELM is used to achieve 96% accuracy on NSL KDD with a significant reduction in testing time (2.6 s). Multiple kernel learning framework is used in [22] by integrating multiple kernel boosting (instead of adaptive boosting) in the kernel version of ELM with its applicability to scale for large datasets and improved accuracy with low false alarms. In [13], PCA is used to de-correlate the features and a probabilistic version of self-organizing map (SOM) is used to provide fuzzy response of the classifier. In [14], kernel PCA is used for feature reduction and SVM as a classifier where RBF function is used in KPCA and SVM is modified by embedding mean value and mean square difference values of attributes to reduce the noise caused by feature difference. In [23], simple rule sets are first determined using classification and regression trees (CARTs) followed by applying discrete wavelet transform (DWT) to transform the features and using SVM classifier to build the IDS model; the DWT features are shown to be more discriminative than standard features of NSL KDD. A modified version of SVM is proposed in [24] which uses ramp loss instead of the hinge loss of conventional SVM to address the problem of class imbalance and tackle the presence of outlier samples in dataset and shows improved performance on both NSL KDD (98.6% accuracy) and UNSW NB15 (93.5% accuracy) datasets.

In [25], a modified version of *k*-means is used to initially reduce the number of samples per class of NSL KDD dataset, followed by applying multi-level ELM and SVM algorithms to classify the input test sample; this strategy is applicable for real-time training since the computational cost of training the classifier is significantly reduced by initial clustering of samples with modified *k*-means clustering. In [7], multi-objective feature selection method is applied followed by new evaluation mechanism that classifies the anomaly into known, similar, and new attacks; this is a new paradigm that enables better performance evaluation of classifiers to detect not only known attacks but also modern zero-day attacks.

Table 4 compares the accuracy of the recent machine learning techniques used for detecting network intrusions. It is evident from the results of [24] that even after applying the same model, the accuracy on UNSW dataset is less (~93%) as compared to NSL KDD dataset (~98%). This implies that the state-of-the-art techniques that perform well for NSL KDD may not work accurately for the modern UNSW dataset; thus, further research is required to model modern attack scenarios. Also, the detection rates of low frequent attacks (R2L and U2R in NSL KDD and worms, shellcode in UNSW) are low as demonstrated by various researchers in their work.

Table 4 Accuracy of recent machine learning techniques for network intrusion detection

Paper	Technique	Description	Dataset	Acc. %
[21]	BMA	Prediction by averaging k-best BN classifier Better predictive power even with small training dataset	NSL KDD	96
[15]	OS-ELM	Fast and can process network packets one at a time or in chunk. Alpha and beta profiling to reduce the effect of imbalanced dataset	NSL KDD, Kyoto	98.66 96.37
[24]	Ramp-KSVCR	KSVCR is used for multi-class classification Ramp loss function is implemented to reduce effect of noise and outliers	NSL KDD, UNSW	**98.68** **93.52**
[25]	Multi-level (SVM, ELM)	Modified K-means to build high-quality training dataset Multi-level model using SVM and ELM	KDD99	95.75
[23]	CART, DWT, SVM	Combination of CART, DWT, and SVM to identify intrusion Visual analysis using interactive PCA to understand intrusion and their relationship	NSL KDD	95.5

6 Open Challenges in NIDS Research

Although many techniques have been developed to classify the packet into normal or attack, still NIDSs face various challenges which need to be addressed so that these systems can be widely deployed commercially. Some of these challenges are:

(a) **Huge volume of data**: IDS should have low computational complexity for training (to be able to learn behavior of new attacks) as well as for testing (for real-time performance). To solve this problem, active learning strategies can be used to select relevant input samples for training the classifier rather than using the full input training dataset.

(b) **High cost of false alarms**: Most IDSs have high false positive rate (FPR) which can be potentially disastrous for the network. If the classifier generates more false positives, the attacker can easily exploit vulnerabilities of the network; in case of false negatives, an alarm is raised even if the packet is normal leading to waste of time and effort for the network administrator. To solve this problem, probabilistic machine learning techniques must be used. With deterministic techniques, an input test sample is always classified into one of the known categories resulting in hard decisions, whereas with probabilistic techniques the decision is soft in terms of probability for each class (e.g., the sample is 80% normal, 15% fuzzer, 5% DoS) due to which the network administrator can take suitable actions.

(c) **Class imbalance**: Since the anomalies are defined as rare instances among network traces, it is obvious that the normal and anomalous class distribution will be skewed. Thus, the classifier should model this phenomenon while building the model via a cost-sensitive variation where the cost reflects the number of samples per class. To solve this problem, weighted extreme learning machine (ELM) can be implemented to improve the performance.

7 Conclusions

The constantly evolving nature of network intrusions demands IDSs that can reliably learn the behavior of incoming traffic pattern. In this work, we have presented the survey of ongoing and recent efforts addressing following aspects of machine learning-based NIDS: (i) recent datasets, (ii) feature-level processing techniques along with feature analysis of UNSW-NB15 dataset, (iii) recent machine learning algorithms including extreme learning machines and multi-level hybrid approaches. Compared to other surveys, this paper provides a comparison of conventional and modern network intrusion datasets, state-of-the-art feature selection, and machine learning techniques and highlights the challenges which can be beneficial for NIDS research community.

For further research in this area, we recommend to consider the use of modern datasets over conventional ones so that performance can be evaluated for the current network scenarios and attacks. To improve the classifier accuracy, appropriate feature selection techniques on recent datasets need to be developed so as to determine prominent features. Unsupervised learning techniques can also be explored and analyzed to solve the problem of clustering various network packets. Active learning strategies and probabilistic machine learning techniques can be implemented to improve the state of the art of NIDSs.

References

1. Mohiuddin, A., Mahmood, A., Hu, J.: A survey of network anomaly detection techniques. J. Netw. Comput. Appl. **60**, 19–31 (2016)
2. Bhuyan, M., Bhattacharyya, D., Kalita, J.: Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. **16**, 303–336 (2014)
3. Weller-Fahy, D., Borghetti, B., Sodemann, A.: A survey of distance and similarity measures used within network intrusion anomaly detection. IEEE Commun. Surv. Tutor. **17**, 70–91 (2015)
4. Garcia-Teodoro, P., Diaz-Verdejo, J., Fernandez, G.: Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. **28**, 18–28 (2009)
5. Hoque, N., Bhuyan, M., Baishya, R., Bhattacharyya, D., Kalita, J.: Network attacks: Taxonomy, tools and systems. J. Netw. Comput. Appl. **40**, 307–324 (2014)

6. Scarfone, K., Mell, P.: Guide to intrusion detection and prevention systems (IDPS). NIST Spec. Publ. **800**, 94 (2007)
7. Viegas, E., Santin, A., Oliveira, L.: Toward a reliable anomaly-based intrusion detection in real-world environments. Comput. Netw. **127**, 200–216 (2017)
8. Shirvai, A., Shirvai, H., Tavallaee, M., Ghorbani, A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. **31**(3), 357–374 (2012)
9. Tavallaee, M., Bagheri, E., Lu, W.: A detailed analysis of the KDD CUP 99 dataset. In: Compuational Intelligence for Security and Defense Applications (2009)
10. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems. In: Military Communications and Information Systems Conference (MilCIS) (2015)
11. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf. Secur. J.: Global Perspect. **25**(1–3), 18–31 (2016)
12. Bhuyan, M., Bhattacharyya, D., Kalita, J.: Towards generating real-life datasets for network intrusion detection. Int. J. Netw. Secur. **17**, 683–701 (2015)
13. De la Hoz, E., De La Hoz, E., Ortiz, A., Ortego, J., Prieto, B.: PCA filtering and probabilistic SOM for network intrusion detection. Neurocomputing **164**, 71–81 (2015)
14. Kuang, F., Xu, W., Zhang, S.: A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl. Soft Comput. **18**, 178–184 (2014)
15. Singh, R., Kumar, H., Singla, R.: An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Expert Syst. Appl. **42**(22), 8609–8624 (2015)
16. Wahba, Y., ElSalamouny, E., ElTaweel, G.: Improving the performance of multi-class intrusion detection systems using feature reduction. IJCSI Int. J. Comput. Sci. Issues **12** (2015)
17. Eid, H., Hassanien, A., Kim, T., Banerjee, S.: Linear correlation-based feature selection for network intrusion detection model. Adv. Secur. Inf. Commun. Netw., 240–248 (2013)
18. Lin, W., Ke, S., Sai, T.: CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowl.-Based Syst. **78**, 13–21 (2015)
19. Iglesias, F., Zseby, T.: Analysis of network traffic features for anomaly detection. Mach. Learn. **101**, 59–84 (2015)
20. Fiore, U., Palmieri, F., Castiglione, A., Santis, A.: Network anomaly detection with the restricted Boltzmann machine. Neurocomputing **122**, 13–23 (2013)
21. Xiao, L., Chen, Y., Chang, C.: Bayesian model averaging of Bayesian network classifiers for intrusion detection. In: Computer Software and Applications Conference Workshops (COMPSACW) (2014)
22. Fossaceca, J., Mazzuchi, T., Sakrani, S.: MARK-ELM: application of a novel multiple kernel learning framework for improving the robustness of network intrusion detection. Expert Syst. Appl. **42**, 4062–4080 (2015)
23. Ji, S., Jeong, B., Choi, S., Jeong, D.: A multi-level intrusion detection method for abnormal network behaviors. J. Netw. Comput. Appl. **62**, 9–17 (2016)
24. Bamakan, S., Wang, H., Shi, Y.: Ramp loss K-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem. Knowl.-Based Syst. **126**, 113–126 (2017)
25. Al-Yaseen, W., Othman, Z., Nazri, M.: Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Syst. Appl. **67**, 396–303 (2017)