Computers & Security 116 (2022) 102675

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

A systematic literature review of methods and datasets for anomaly-based network intrusion detection

Zhen Yang[a], Xiaodong Liu[a], Tong Li[a,*], Di Wu[a], Jinjiang Wang[a], Yunwei Zhao[b], Han Han[b]

[a] Department of Faculty of Information Technology, Beijing University of Technology, Beijing, China
[b] CNCERT/CC, Beijing, China

ARTICLE INFO

Article history:
Received 27 September 2021
Revised 28 November 2021
Accepted 27 February 2022
Available online 1 March 2022

Keywords:
Intrusion detection
Systematic literature review
Machine learning
Datasets

ABSTRACT

As network techniques rapidly evolve, attacks are becoming increasingly sophisticated and threatening. Network intrusion detection has been widely accepted as an effective method to deal with network threats. Many approaches have been proposed, exploring different techniques and targeting different types of traffic. Anomaly-based network intrusion detection is an important research and development direction of intrusion detection. Despite the extensive investigation of anomaly-based network intrusion detection techniques, there lacks a systematic literature review of recent techniques and datasets. We follow the methodology of systematic literature review to survey and study 119 top-cited papers on anomaly-based intrusion detection. Our study rigorously and comprehensively investigates the technical landscape of the field in order to facilitate subsequent research within this field. Specifically, our investigation is conducted from the following perspectives: application domains, data preprocessing and attack-detection techniques, evaluation metrics, coauthor relationships, and datasets. Based on the research results, we identify unsolved research challenges and unstudied research topics from each perspective, respectively. Finally, we present several promising high-impact future research directions.

1. Introduction

Computer and network technologies play an increasingly important role in our daily life, however their benefits have been balanced somewhat by serious network attacks in recent years. The 2020 NTSC (National Technology Security Coalition) security report points out that significant network security issues have been aggravated every month.[1] In 2019, about 620 million account details were leaked by hackers and sold on the dark web. This threatening situation has been exacerbated by the COVID-19 pandemic, because many people have to work from home, resulting in a significant increase in network traffic. According to the 2020 CIRA (Canadian Internet Registration Authority) Cybersecurity Survey, two-thirds of IT workers were required to work from home because of COVID-19.[2]

Intrusion detection system (IDS) is an effective security mechanism that monitors network traffic and prevents malicious requests. Research of intrusion detection is evolving rapidly with the development of machine learning. Traditional machine learning techniques have been widely used in intrusion detection, such as decision tree (DT) (Safavian and Landgrebe, 1991), random forest (RF) (Zhang et al., 2008), and support vector machine (SVM) (Hsu et al., 2003). And, with the development of deep learning, convolutional neural network (CNN) (Vinayakumar et al., 2017), recurrent neural network (RNN) (Yin et al., 2017), and long short-term memory (LSTM) (Roy et al., 2017) are becoming popular in intrusion detection. These techniques are based on different principles, and how to effectively exploit their advantages to address intrusion detection tasks in particular domains remains an open research question. Moreover, due to the high dimensionality and complexity of the data, a common solution is to use data preprocessing techniques which might help to reduce the dimensionality and thus enable the researchers to be able to deal with these high-dimensional spaces. Preprocessing methods can affect detection performance, and should be carefully considered in the design of intrusion-detection methods.

Given the above research issues, a systematic and comprehensive literature review can contribute to the development of the community. Existing IDSs can be divided into two categories based on the detection method: anomaly-based detection and misuse-based detection or signature detection

* Corresponding author.
E-mail address: litong@bjut.edu.cn (T. Li).

[1] https://www.ntsc.org/assets/pdfs/cyber-security-report-2020.pdf

[2] https://www.cira.ca/cybersecurity-report-2020

https://doi.org/10.1016/j.cose.2022.102675
0167-4048/

(Axelsson, 2000; Ghorbani et al., 2009). Anomaly-based network intrusion detection is an important research and development direction of intrusion detection. We follow the methodology of the systematic literature review (SLR) to survey 119 highly cited papers on anomaly-based network intrusion detection. We diversify our analysis from multiple perspectives. First, we analyze research progress and identify potential bottlenecks in specific application scenarios, such as the Internet of Things (IoT) and industrial control networks. Second, we study preprocessing techniques, such as data cleaning, feature selection, and feature transformation, which can provide suggestions for data preparation. Third, we discuss intrusion detection techniques and analyze their principles and related applications by technology category. Fourth, we investigate evaluation methods, including metrics and datasets, which can help us to standardize them. Fifth, to study the current state of the community, we count the contributors and map the collaboration network. Finally, we conduct a systematic survey of cybersecurity datasets, so as to better understand them and evaluate their applicability.

In summary, the contributions of this paper are as follows:

- We are the first to use the SLR methodology to survey and study the 119 most highly cited papers in the field of network security intrusion detection, which were systematically screened from 14,942 candidates.
- We establish a comprehensive technical overview of the intrusion detection field from both coarse- and fine-grained perspectives. We provide a comprehensive overview of 52 cybersecurity datasets and label them according to their attributes.
- The analysis of approaches sheds light on future research directions.

The remainder of this paper is organized as follows. Section 2 summarizes existing literature review work related to intrusion detection systems and datasets, and Section 3 presents a literature review of our intrusion-detection system methodology. Our research results are presented in Section 4. Section 5 concludes this paper and discusses future research directions.

2. Related work

A wealth of literature covers various aspects of intrusion detection. In this section, we present existing related works and compare them with our study.

2.1. Survey of intrusion detection methods

Most of the related research focuses on intrusion detection methods. Bhuyan et al. (2013) briefly describe and compare a large number of network anomaly detection methods and systems. Ahmed et al. (2016) analyzed anomaly detection methods and the complexity of machine learning/data mining (ML/DM) algorithms. Milenkoski et al. (2015) evaluated common practices for intrusion detection systems by analyzing the existing standard evaluation parameters, including workloads and metrics. Buczak and Guven (2015) discussed machine learning and data mining methods for network analysis to support intrusion detection. Hodo et al. (2017) presented a classification of shallow and deep network intrusion detection systems, investigated the performance of machine learning techniques in detecting anomalies, and discussed false and true positive alarm rates. Wang and Jones (2017) reviewed the applications of data mining, machine learning, deep learning, and big data in intrusion detection. Haq et al. (2015) conducted extensive research on the application of machine learning techniques in intrusion detection. Mishra et al. (2018) discuss the application of machine learning methods in intrusion detection and provide attack classification and attack feature mapping for each attack. Nisioti et al. (2018) provide a comprehensive overview of unsupervised and hybrid intrusion detection methods and also present and emphasize the importance of feature engineering techniques in intrusion detection.

These studies classified intrusion detection techniques based on technical principles and detailed their advantages and disadvantages, but provided no research ideas or methods from the point of view of reproducibility. This weakens their stringency and is not conducive to further research. In addition, such papers lack comprehensiveness in the discussion of intrusion detection methods. We have more fully studied and discussed several aspects of intrusion detection such as pre-processing methods, analytical models and evaluation methods.

2.2. Survey of application of different fields in IDS

Zarpelão et al. (2017) surveyed IDS in the IoT. In an overview of IoT devices, they argued that the IoT paradigm has the phases of collection; transmission; and processing, management, and exploitation, and presented a range of technologies that can be used for IoT devices, focusing on wireless technologies. Hande and Muddana (2021) provide an overview of existing security solutions for SDNs and a comparative study of various IDS approaches based on deep learning models and machine learning methods.

These studies discuss the current state of research on intrusion detection under a certain target network. Compared to these works, our research is broader, covering the Internet, the Internet of Things (IoT), Software Defined Networks (SDN), and Industrial Control Networks (ICN). And, we also investigate datasets from different domains for researchers' reference.

2.3. Survey of intrusion detection datasets

Ring et al. (2019) identified 15 features of 34 intrusion detection datasets, categorized in five groups: general information, evaluation, recording environment, data volume, and data nature. Thakkar and Lohiya (2020) investigated different IDS datasets and research advances used to evaluate IDS models, focusing on the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets. The studies mentioned above focus on the characteristics of the dataset and the progress of the study. Compared to these studies, our research also discusses intrusion detection principles and related methods.

Further, we categorized the relevant studies mentioned above. We categorized these studies according to the following criteria and present the results in Table 1.

- Methodology: it indicates whether the study is based on SLR methodology.
- Intrusion detection technique: it indicates whether the study discusses intrusion detection techniques, and furthermore can be specific to preprocessing approaches, analysis models and evaluation methods.
- Multi-field: it indicates whether the study discusses the current state of research on intrusion detection in different network environments.
- Dataset: it indicates whether the study covers the relevant research of the dataset.

As shown in Table 1, in contrast to other studies, our study follows the SLR Methodology with comprehensive coverage of intrusion detection techniques (including preprocessing methods, analytical models, and evaluation methods) and datasets, and explores multi-target networks.

3. Research methodology

Various intrusion detection systems have been proposed. We have developed a research protocol according to the methodology

Table 1
Related studies on network intrusion detection survey.

Related work	Year	SLR-based	Intrusion detection method			Multi-field	Dataset
			Preprocessing	Model	Evaluation		
Bhuyan et al. (2013)	2014	×	√	√	√	×	√
Milenkoski et al. (2015)	2015	×	×	×	√	×	×
Haq et al. (2015)	2015	×	√	√	√	×	√
Buczak and Guven (2015)	2015	×	×	√	√	√	√
Ahmed et al. (2016)	2016	×	×	√	×	×	√
Hodo et al. (2017)	2017	×	×	√	√	×	×
Wang and Jones (2017)	2017	×	×	√	√	×	×
Zarpelão et al. (2017)	2017	×	×	√	×	√	×
Nisioti et al. (2018)	2018	×	×	√	√	×	√
Mishra et al. (2018)	2019	×	√	√	×	×	×
Ring et al. (2019)	2019	×	×	×	×	×	√
Thakkar and Lohiya (2020)	2020	×	×	×	×	×	√
Hande and Muddana (2021)	2021	×	×	√	×	×	×
Our study		√	√	√	√	√	√

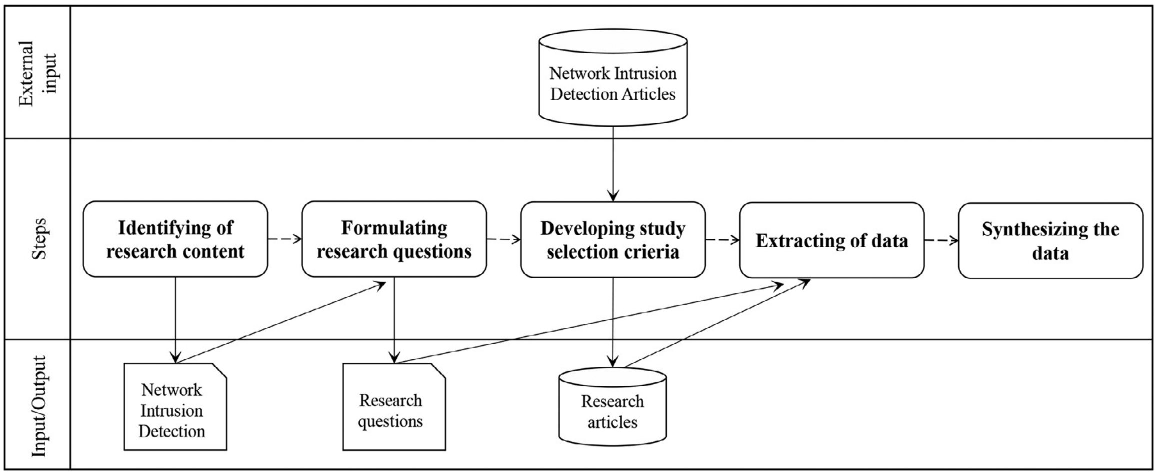
External input
Network Intrusion Detection Articles
Steps
Identifying of research content
Formulating research questions
Developing study selection crieria
Extracting of data
Synthesizing the data
Input/Output
Network Intrusion Detection
Research questions
Research articles

Fig. 1. SLR process.

of the systematic literature review (SLR) (Keele et al., 2007), as shown in Fig. 1. This includes identification of research, research questions, study selection, data extraction, and data synthesis. The method is approached with mixed methods (qualitative and quantitative research methods) to more visually represent the above needs.

3.1. Identification of research content

To obtain a comprehensive set of papers required an unbiased search strategy to find original reviews related to intrusion detection systems. The search process must be as rigorous and sensible as possible, and search terms must be defined. We find that some anomaly-based intrusion detection articles are named with intrusion detection. Therefore, to fully cover anomaly-based intrusion detection articles, we defined the search term as “network intrusion detection”.

Before we started our literature search work, we evaluated three databases, Scopus, Google Scholar and Web of Science. Scopus covers the major publishers of RE (ACM, Springer, IEEE) and is more inclusive than Web of Science, but less inclusive than Google Scholar. However, Google Scholar may include many papers that are not peer-reviewed, such as technical reports. For these reasons, we used Scopus to perform the publication search.

3.2. Research questions

We developed ideas for the analysis of a paper and articulated specific research questions (RQs), as shown in Table 2, including detailed sub-questions, to guide our research. First, we summarize the network environment in which intrusion detection techniques are applied (RQ1), which helps us to analyze the characteristics of the development and application of intrusion detection techniques. Second, we investigate the data preprocessing techniques (RQ2) and intrusion detection datasets (RQ6) commonly used in intrusion detection and make recommendations for the data preparation phase based on the findings. Third, we focus on the intrusion detection techniques proposed in the paper (RQ3), including framework (RQ3(a)), learning method (RQ3(b)), and types of supervision (RQ3(c)). Also, we are very interested in the principles and applications of the model (RQ3(d)). Fourth, evaluation methods are important to measure the capability of intrusion detection techniques, so we would like to learn about the general evaluation metrics in this area (RQ4). Finally, we are also interested in the authors of the papers (RQ5).

3.3. Study selection

The following research principles ensure consistent evaluation and minimize subjectivity.

Table 2
Research questions .

RQ1	Application domains	(a) What domains are covered by intrusion detection techniques?
		(b) How are the studies distributed among the different areas?
		(c) What are the reasons for this distribution?
		(d) How does research in these domains differ from country to country?
RQ2	Data preprocessing methods	(a) What are the common data preprocessing techniques used in network intrusion detection?
		(b) How are the preprocessing technologies implemented and what are their technical features?
		(c) What is the distribution of their applications in intrusion detection?
RQ3	Detection techniques	(a) Which models are applied in intrusion detection techniques?
		(b) How does machine learning and deep learning apply to intrusion detection?
		(c) What is the distribution of supervision types?
		(d) What are the principles and characteristics of different intrusion detection technologies?
RQ4	Evaluation metrics	(a) How is the performance of intrusion detection technologies evaluated?
		(b) In our research articles how are these evaluation methods applied?
RQ5	Authors	(a) Who are the main contributors to the articles?
		(b) What does the network of co-authors look like?
RQ6	Datasets	(a) What are the available public datasets?
		(b) Which datasets are often used in network intrusion detection?
		(c) Why are these datasets widely used?

Table 3
Exclusion criteria and their interpretation.

Criteria	EC/IC	Criteria explanation
Inclusion	IC1	Research efforts are explicitly and specifically dedicated to intrusion detection systems.
	IC2	Research on intrusion detection methods based on machine learning
Exclusion	EC1	Does not meet average of 10 citations per year.
	EC2	A paper shorter than 6 pages contains insufficient research content.
	EC3	Not an anomaly-based intrusion detection paper.
	EC4	Only a review paper.

- *Explicit inclusion and exclusion criteria.* These should be explicitly outlined; see Table 3. We screened papers for quality, length, and type to obtain a collection that could be used effectively for research and analysis.
- *Objective review strategy.* Papers should be reviewed for inclusion or exclusion by at least two reviewers with knowledge in the field. Information determination for datasets should be reviewed for applicability by at least two reviewers. A third reviewer makes a final decision if there is disagreement.

3.4. Data extraction

Papers collected from the database are filtered using our defined criteria, as shown in Fig. 2. After filtering, we had 119 papers related to intrusion detection.

Relevant information on each paper was extracted and tagged for analysis. There are two categories of tags. The first can be obtained from the content of a paper: year of publication, author, number of citations, domain, model, evaluation metrics, and dataset. The second is based on the learning method and supervision type.

It was not necessary to carefully read the full text of a paper. We read the title, abstract, and introduction, which contained most of the information, and examined the text if necessary.

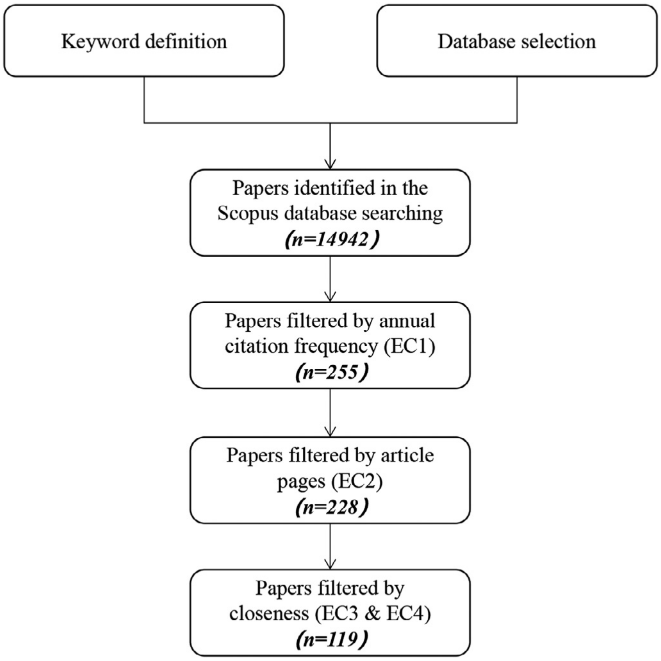
Keyword definition
Database selection
Papers identified in the Scopus database searching (n=14942)
Papers filtered by annual citation frequency (EC1) (n=255)
Papers filtered by article pages (EC2) (n=228)
Papers filtered by closeness (EC3 & EC4) (n=119)

Fig. 2. Paper selection process.

3.5. Data synthesis

For learning methods and types of supervision, we annotated by analyzing the employed models. We classified domains through research of the traditional Internet (Web), IoT, industrial control networks (ICNs), and software defined networks (SDNs). Lacking a specific application scenario, we classified papers as such. The IoT is a network of objects embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems via the Internet. IoT technologies are most often associated with the "smart home" concept. ICNs are networks of digital control systems connected using the Ethernet standard. Industrial networks implement communication protocols between field devices, digital controllers, various software suites, and external systems. SDNs enhance network control through programming. This combination of features can bring the benefits of enhanced configuration, improved performance, and new architecture.

It is also necessary to label datasets. Datasets for network packet analysis in commercial products are not readily available for privacy reasons. Publicly available datasets such as DARPA, KDD, and UNSW-NB15 are widely used as benchmarks. We defined labels, including "Year", "Authenticity", "Count", "Labeled",

Table 4
Number of papers by domain in intrusion detection methods.

Domain	Number
Internet	88
Internet of Things (IoT)	20
Industry Control Network (ICN)	9
Software Defined Network (SDN)	2

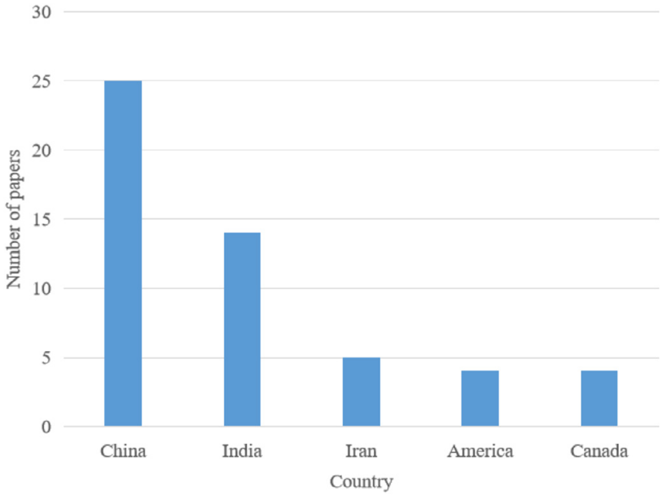

Fig. 3. Five countries with the most published papers.

and "Number of labels" to reflect the availability, novelty, authenticity, and data volume of a dataset.

4. Analysis of intrusion detection literature review

In this section, we present and analyze our findings. We provide an in-depth analysis and present our recommendations in the field of network intrusion detection from various perspectives, including method application areas, data preprocessing methods, detection techniques, evaluation metrics, datasets, and authors.

4.1. Application domain

From the perspective of application domain, 119 articles can be categorized as Internet, IoT, ICN and SDN (**RQ1(a)**). It is worth noting that we classify all articles that do not specify a particular application domain as Internet. We summarize the number of 119 articles classified based on different application domains in Table 4, answering **RQ1(b)**. More than 70% of the papers apply to the Internet, which indicates that the security of traditional networks is an important research topic. This distribution is on one hand due to the relative maturity of intrusion detection research in the traditional network domain, and many papers are devoted to the improvement or refinement of existing methods (**RQ1(c)**). On the other hand, there are more public datasets used in this field, such as KDD99, UNSW-NB15, etc., which are easy for researchers to analyze and experiment. There are fewer research articles in the field of ICN and SDN, with only 9 and 2 articles respectively. Through our research, we found that articles in the ICN domain do not disclose their datasets due to the confidentiality of industrial network data, thus also limiting their development in security research. The lack of datasets is also a key factor limiting research in the SDN field. Researchers often need to build SDN network environments to simulate data before conducting security research.

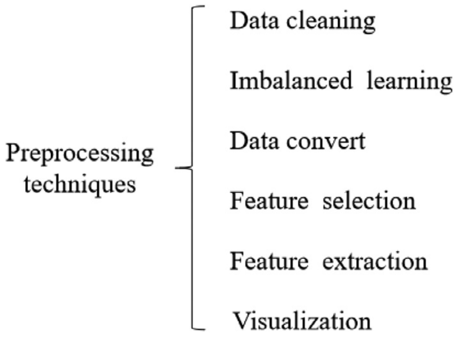

Fig. 4. Data preprocessing techniques commonly used in intrusion detection.

Different countries or regions also show different trends in different application domains. As can be seen in Fig. 3, China and India are in a clear lead in the number of articles published in the field of network intrusion detection (**RQ1(d)**). This indicates that the two major developing countries, China and India, currently have a great devotion and importance to cyber security. With the rapid development of IoT technology, IoT security has also been a hot research topic in recent years, with half of the researched articles published in the past three years. As in the traditional network environment, China is leading the way in the number of articles published in the IoT field, accounting for 1/5 of the total. Based on the rapid development of 5G technology and the empowerment of IoT, IoT will be in a state of continuous development in the future, and the security issues and related research will also become a research hotspot in the field of security.

4.2. Data preprocessing methods

The representation and quality of the data is of primary importance in any data analysis process (Pyle, 1999). Raw data usually contain noisy and unreliable data that can affect training and analysis. Moreover, datasets used in intrusion detection are characterized by high dimensionality, making it more difficult to discover knowledge during training. To build high-performance detectors requires efficient preprocessing. Fig. 4 summarizes the common data preprocessing methods used in intrusion detection, including data cleaning, imbalance learning, data transformation, feature selection, feature extraction, and Visualization (**RQ2(a)**).

The above methods preprocess the data from different perspectives to enable the analysis model to better learn useful information from the data. In order to study the technical characteristics of different methods, we summarize their implementation principles and applications to answer **RQ2(b)**.

a) Data cleaning

Data cleaning corrects corrupt or inaccurate records. Quality criteria may include the following.

- **Validity.** Data might have to be of a certain type, such as Boolean or numeric.
- **Accuracy.** Data must conform to the situation. For example, outliers may exist due to the recording process. Accuracy is difficult to guarantee through data cleaning because real data sources are needed for validation.
- **Completeness.** Some data may have unknown or missing values. Completeness issues are generally resolved through default values, setting zeros, or removal.
- **Uniformity.** Inconsistency occurs when there are conflicts in a dataset. For example, a source IP may differ between two receivers. Fixing this type of problem requires the determination of which datum is most reliable.

Data analysis based on means, standard deviations, or clustering algorithms can reveal errors, whose values can sometimes be set to a mean or other statistical measure.

b) Imbalanced learning

A sample with different proportions of positive and negative cases will lead to a bias in the learning process toward the higher proportion. For example, in the extreme case of a dataset with 95% positive and 5% negative cases, the model will have no practical meaning. Since attacks tend to be sparse, datasets in intrusion detection are often unbalanced. The following methods may improve performance.

- **Sampling methods.** A balanced dataset usually provides better overall classification performance (Estabrooks et al., 2004; Weiss and Provost, 2001), and to obtain the same proportion of positive and negative examples is the most common imbalanced learning method. The simplest method, to undersample the majority class, obviously leads to information loss. Liu et al. (2008) introduced the EasyEnsemble and BalanceCascade algorithms, combining a subset of majority classes with minority classes and performing ensemble learning on the classifiers. NearMiss uses a KNN classifier to select the majority class with the smallest average distance from the minority class (Mani and Zhang, 2003). Oversampling of minority examples is usually accomplished by synthetic sampling. The synthetic minority oversampling technique (SMOTE) synthesizes data based on feature space similarity between minority examples (Chawla et al., 2002). Modified and extended sampling algorithms (Fernández et al., 2018) have been proposed to address the problem of over-generalization in SMOTE (Wang and Japkowicz, 2004). The adaptive integrated sampling method (ADASYN) (He et al., 2008) adaptively creates synthetic data based on the distribution of minority examples and solves the problem of overlapping between classes.
- **Cost-sensitive approach.** This approach considers the costs associated with misclassification and uses a different cost matrix to describe them when misclassifying data (Ting, 2002). It has been shown to be a viable alternative to sampling methods (McCarthy et al., 2005).
- **Additional methods.** Many algorithms obtain good performance from other perspectives. Based on kernel methods as well as active learning, Ertekin et al. proposed an SVM-based active learning method (Ertekin et al., 2007) that restricts the query process in each iteration of active learning to the data pool rather than the entire dataset. The SVM is trained during this process, and the most informative instances are extracted from the hyperplane to form a new training set. One-class learning uses mainly or only single-class samples for recognition, distinguishing it from traditional distinction-based induction, with good results with extremely unbalanced data (especially high feature space dimensions) (Raskutti and Kowalczyk, 2004).

c) Data conversion

Training data must often be transformed and mapped before being fed to the model to accommodate requirements, and to improve detection speed and accuracy. This affects two types of data in IDS datasets.

- **Non-numeric data.** Taking the UNSW-NB15 dataset as an example, features in nominal form include the type of transport protocol, state, service type, and attack type, stored as strings, which most machine learning algorithms do not support. The most straightforward way is to number the values under a feature and map them, but this will cause errors. For example, in the calculation of mean square error,

$$MSE = \frac{1}{M}\sum_{m=1}^{M}\left(y_m - \hat{y}_m\right)^2, \tag{1}$$

the MSE of misclassifying a class marked as 0 as a class marked as 9 will be 81 times that of a class misclassified as 1, which is unreasonable. One-hot encoding is a common way to handle this. The algorithm uses n-bit status registers to encode n states. Only one corresponding register bit is valid when a given state is in effect.
- **Numeric data.** The range of values differs by feature. Deep learning frameworks avoid its impact on model accuracy by introducing bias, but the time spent on model learning may still be affected when the ranges of values of two features are too different. For example, in ML optimization using gradient descent, the eigenvalues of $\{X^T X\}$ (i.e., the scales of features) determine the speed of convergence to a global or local minimum. Therefore, value ranges of features are often unified by data scaling before training, such as by min-max normalization. Every value of a feature is mapped to between 0 and 1,

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{2}$$

However, this method cannot handle outliers. For example, with nine values between 0 and 1 and an outlier equal to 100, the nine smaller values will be mapped to values between 0 and 0.01. This can be avoided by z-score standardization,

$$x' = \frac{x - \mu}{\sigma}, \tag{3}$$

where μ and σ are the mean and standard deviation, respectively, of the feature. This can scale the values to near 0 while preserving the distribution of the features, but the features may not be on exactly the same scale.

d) Feature selection

Feature selection is the selection of a subset of the original dataset as model input. This can avoid dimensional disasters and enhance generalization (Bermingham et al., 2015). To perform feature selection requires that data contain redundant or irrelevant features, so as to avoid excessive information loss. Feature selection can be accomplished in several ways.

- **Manual selection.** Whether to remove a feature is manually determined. See, for example, Zhang et al. (2018).
- **Exhaustive search.** To test every possible subset of features to find the subset with the lowest error rate can be computationally intensive (Guyon and Elisseeff, 2003).
- **Embedded method.** Feature selection is performed during model construction. The Bolasso algorithm (Bach, 2008) reduces many regression coefficients to zero by constructing a linear model and combining the L1 penalty with the L2 penalty of ridge regression. FeaLect (Zare et al., 2013) scores and selects features based on the combinatorial analysis of regression coefficients.
- **Wrapper approach.** A prediction model is trained with each subset and tested on a holdout set. The score of a subset is obtained from the error rate of the model test. This is computationally intensive, and usually used only to find the best subset of features.
- **Filtering method.** Methods such as mutual information (Guyon and Elisseeff, 2003), Pearson correlation coefficients, and significance scores, such as inter- or intra-class distances (Yang and Pedersen, 1997), can be used to score a subset of features, which can rank features but does not produce the best subset.

e) Feature extraction

Unlike feature selection (Sarangi et al., 2020), feature extraction, i.e., the creation of new features to facilitate learning, is considered a key factor in building a model. This can be performed by the following algorithms.

- **Principal Component Analysis (PCA).** One of the most used linear dimensionality reduction methods, PCA changes the basis of data according to principal components, which are essentially eigenvectors of the data covariance matrix. De la Hoz et al. (2015), used PCA for feature extraction, while Xiao et al. (2019) combined PCA with autoencoders to compress high-dimensional features for input to CNNs. Variants include probability PCA (PPCA), which utilizes probability distributions; kernel PCA, which uses kernel functions to map low-dimensional spaces to high-dimensional spaces before using PCA to reduce the dimensions; and independent component analysis (ICA), which requires no hidden variables to obey Gaussian distributions.
- **Linear Discriminant Analysis (LDA).** A classic dime nsionality-reduction method, LDA finds linear combinations of features to describe multiple classes of objects. As a supervised learning algorithm, it searches the low-dimensional space for the vectors that best distinguish classes of data (Martinez and Kak, 2001), projecting data in low dimensions to minimize intra-class distances and maximize inter-class distances. Subba et al. (2015) built an intrusion-detection model using LDA with logistic regression, with significant advantages in computational efficiency.
- **Autoencoder.** This method uses hidden layers for unsupervised learning, mapping high-dimensional features by a nonlinear transformation (Goodfellow et al., 2016) to produce a representation as close as possible to the original input. Regularized autoencoders (sparse, denoised, and shrunken) are commonly used in learning representation (An and Cho, 2015). Zhang et al. (2018) achieved 98.80% accuracy on the UNSW-NB15 dataset with a denoising autoencoder (DAE).

f) Visualization

Data visualization is a graphical representation of data that is used to better help researchers understand characteristics such as data distribution. In intrusion detection, visualization helps us further understand the characteristics of an attack by refining data attributes and characteristics. And, due to the incomprehensibility of machine learning algorithms, we are often unable to analyze the causes of classifier misclassification. Using visualization techniques, we better identify attack behaviors for deeper analysis. Data dimensionality reduction works by extracting a subset of the original features or transforming the original data to a lower dimensional space. In intrusion detection, t-SNE and PCA are often used to implement network traffic visualization.

- **t-distributed Stochastic Neighbor Embedding (t-SNE).** The t-SNE (Van Der Maaten, 2014) technique is a dimensionality reduction technique used to visualize high-dimensional data sets by representing them in a low-dimensional space in two or three dimensions. It is based on the improvement of distributed Stochastic Neighbor Embedding (SNE) (Hinton and Roweis, 2002), which solves the drawback of crowded sample distribution and inconspicuous boundary of SNE after visualization. Visualization can help researchers understand information about data distribution, sample overlap, etc. For example, Fig. 5 shows the distribution of normal samples (green points) and attack samples (gray points) after visualization. Visualization methods have been practically applied to intrusion detection. Hamid and Sugumaran (2020) performed data dimensionality reduction and visualization based on t-SNE and combined with support vector machine for classification in their study. The results showed that the detection rate was improved for almost all attack groups. Yao et al. (2020) proposed a new unsupervised intrusion detection algorithm based on t-SNE and hierarchical neural networks. In this study, the authors used two-dimensional visualization techniques to visually determine the effect of dimensionality reduction.
- **Principal Component Analysis (PCA).** As we talked about in the previous section, PCA is commonly used for dimensionality reduction of high-dimensional data and can be used to extract the main features of the data. It is also often used to visualize data. Fig. 6 shows the distribution of the data after visualization based on PCA. The green points are normal samples and the gray points are attack samples. In studies related to intrusion detection, Ruan et al. (2017) visualized the KDD99 dataset based on PCA and proposed a new sampling method that can visually identify normal classes because it has the compactness and uniqueness of internal classes. In Bulavas et al.'s study Bulavas (2018), the authors proposed an intrusion detection method based on the PCA method for data visualization combined with decision trees. Experiments show that the method has shown good performance in the detection of a variety of attacks.

Table 5
Statistics of feature engineering methods.

Method	Count
Swarm intelligence algorithms	12
Manually defined rules	8
PCA	6
Deep learning	5
Clustering	4
SVM	2
Decision tree	1

Compared with other preprocessing methods, feature selection and feature extraction are the research points of many articles. Among all the papers we investigated, 38 focus on making improvements to feature engineering algorithms (including feature selection, feature extraction), while the others focus on improving classification algorithms. In general, it can be considered that current research on IDS is more focused on improving the performance of classification algorithms. We believe this trend is caused by the fact that the format of each dataset is too different, a problem implying that the generalization of feature-related algorithms is usually worse, further leading to inefficiency in the application of feature engineering algorithms. We summarize the algorithms used in these 38 articles, see Table 5. In the results shown in Table 5, swarm intelligence algorithms can be found to be relatively more popular, which we believe is again related to the dataset (**RQ2(c)**). Given that IDS datasets usually have too many features, researchers usually focus more on performing feature selection. With the difficulty of determining the importance of features, swarm intelligence algorithms with a certain degree of randomness become the preferred choice.

In contrast to feature engineering, researchers have tended to improve classification algorithms based on integration learning and deep learning, which have higher capabilities, in order to obtain more accurate classification results. It is worth mentioning that deep learning itself can also perform feature engineering, which is one of the reasons why deep learning is widely used in the research of IDS. In future work, we also propose to combine feature engineering with visualization, which will help us understand features such as data distribution to further understand the characteristics of attacks.

4.3. Detection technique

We summarize the classification models used in the article in Tables 6 and 7, answering **RQ3(a)**. The most commonly used machine learning algorithm for intrusion detection is SVM, a discrim-

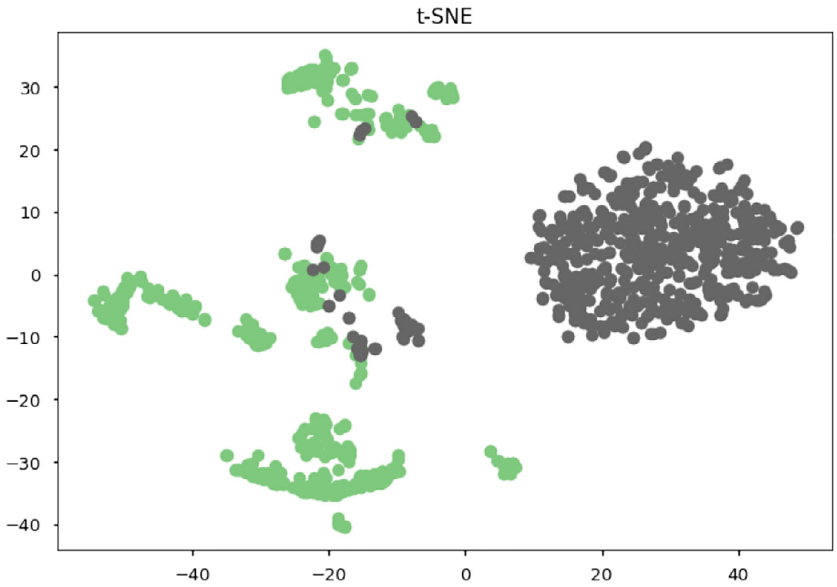

Fig. 5. Visualization plots based on t-SNE.

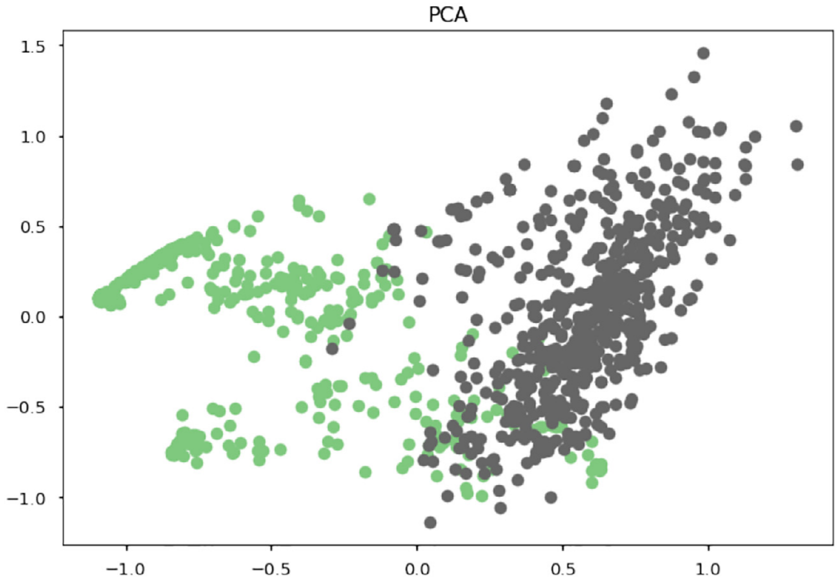

Fig. 6. Visualization plots based on PCA.

Table 6
Most used Machine Learning Algorithms in Proposed Methods.

Type	Method	Count
Supervised learning	Support vector machine	21
	Decision tree	11
	Naive Bayes	8
	K-nearest neighbors	7
	Random forest	5
	AdaBoost	2
	Hidden Markov model	1
Unsupervised learning	K-means	4
	DBSCAN	1

Table 7
Most used Deep Learning Algorithms in Proposed Methods.

Type	Method	Count
Supervised learning	DNN	7
	RNN	7
	CNN	7
	LSTM	4
	DBN	2
	DFFNN	2
	BPNN	1
	ELM	1
Unsupervised learning	Autoencoder	5
	Self-taught learning	3
	RBM	1

inative classifier defined by a split hyperplane that uses a kernel function to map training data to a high-dimensional space for linear classification of intrusions. Data used in intrusion detection usually have high dimensionality, with which SVM has high generalization capability and performs well. Decision trees are widely used due to their high efficiency and interpretability.

Deep learning is evolving rapidly, and is becoming the basis of more intrusion detection methods. To answer **RQ3(b)**, we plotted

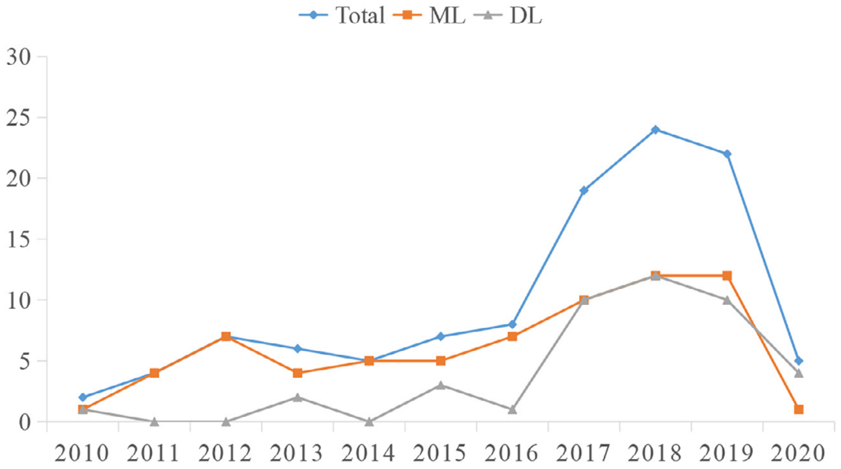

Fig. 7. Changes in numbers of papers published over time.

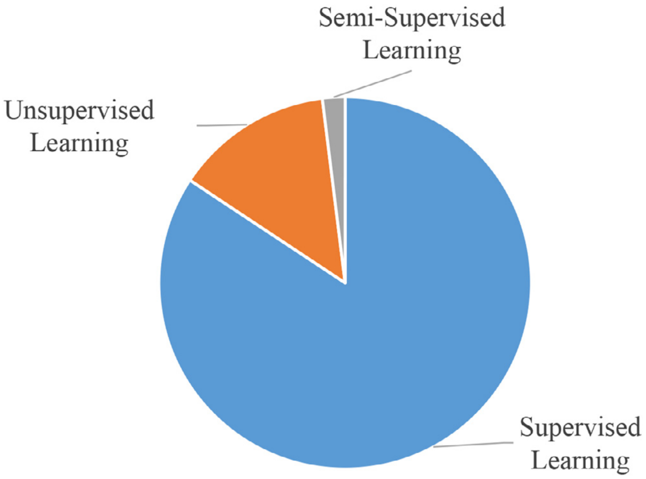

Fig. 8. Number of published papers over time.

the number of ML and DL based intrusion detection articles per year, see Fig. 7. Based on the specificity of our research subject, the number of studies in 20192020 declined compared to previous years. However, the search criteria may have filtered out some quality papers with fewer citations. There is an upward trend in the number of annual papers because, with more frequent network attacks, people are paying more attention to network security. As the Internet carries increasing amounts of information, it has become a profitable target for attackers. In addition, hacking tools and techniques are readily available.

We can see that traditional machine learning methods are still the mainstream technology. These are easier to deploy and implement than deep learning methods, are not limited by computing power, and are more interpretable. However, the trend in the number of machine learning papers is similar to that of deep learning. The rapid development of deep learning has added to research of intrusion detection. Both SVM and DT are supervised learning methods, and they require labels during training. Labeling is time-consuming and tedious for large datasets, and clustering algorithms may be a better choice. K-means and DBSCAN are commonly used clustering algorithms, as shown in Table 6. We classify methods by the supervision type, as shown in Fig. 8, from which it can be seen that supervised learning is most widely used (**RQ3(c)**). This is because many publicly available datasets are already labeled and researchers prefer supervised learning. As mentioned earlier, annotation is time-consuming and labor-intensive. Therefore, unsupervised and semi-supervised learning should also be of interest.

Among the four target networks, SVM and DT are the most widely used machine learning algorithms in the “Internet” and have achieved excellent performance. Unsupervised learning methods are more popular in the “IoT” and “ICN”. For “SDN”, only two studies are included in our work, and they use RNN and RF algorithms, respectively.

Although machine learning and deep learning are increasingly being used for network intrusion detection, the effectiveness of these methods can be significantly reduced in adversarial environments. An adversarial environment is one in which adversaries consciously limit or prevent accurate performance by some means. For example, adversaries design adversarial examples and add them to the training set to trick the model into producing incorrect outputs. Alhajjar et al. (2021) generate adversarial examples using evolutionary computation (particle swarm optimization and genetic algorithms) and deep learning (generative adversarial networks). Their adversarial example generation technique caused high misclassification rates in 11 different machine learning models and voting classifiers. To improve the robustness of intrusion detection algorithms in adversarial environments, researchers have also conducted related research. Caminero et al. (2019) proposed an adversarial environment reinforcement learning algorithm for intrusion detection. They added an adversarial agent strategy to the training to increase the classifier’s false predictions and force it to learn the most difficult cases, ultimately obtaining better results. However, in general, more efforts are needed to study intrusion detection algorithms in adversarial environments.

In order to study the principles and characteristics of different intrusion detection models, we present them in the following and analyze their principles and related applications to answer **RQ3(d)**.

- DT supports decision making through a tree-like model consisting of decisions and their outcomes, is widely used in classification tasks (often called classification trees), and therefore is a common supervised learning classification method in IDS. A trained DT makes multiple selections of a packet’s features to determine its class. An optimal DT holds the most data with a minimum number of levels (Quinlan, 1983). Several algorithms have been proposed to generate optimal trees, such as ID3 (Quinlan, 1986), C4.5 (Quinlan, 2014), and classification and regression tree (CART) (Loh, 2011). There are different metrics to measure DT performance. ID3 uses information gain (entropy) and makes decisions by selecting attributes with the highest information gain, but does not support missing and continuous values in features, which limits its applicability. C4.5 uses the information gain ratio based on ID3, and prunes the tree by replacing branches that do not help as leaf nodes. CART uses Gini impurity (an information-theoretic measure corresponding to Tsallis entropy) as a metric, solving the problem that ID3 not handle the regression task.

DT has an intuitive classification strategy, is interpretable and simple to implement, and often allows for better generalization through post-construction pruning, making it a common model in intrusion detection. Anthi et al. (2019) proposed a three-layer intrusion detection system (IDS) that identifies IoT devices based on MAC addresses, classifies messages as bona fide or malicious, and employs DTs to classify attacks. Abbes et al. (2010) classified records as benign or anomalous by analyzing application protocols, using separate and distinct adaptive DTs for each. The system achieved good results identifying DoS attacks, scanning attacks, and botnets. Muniyandi et al. (2012) proposed an anomaly-detection method that uses k-means to form k clusters of training instances based on Euclidean distance similarity, and C4.5 on each cluster to construct DTs of normal and abnormal instance density regions.

The disadvantage of DT is weak robustness; small changes in training data may result in a completely different DT. Furthermore, information gain is biased toward attributes with more levels (Deng et al., 2011), so larger DTs may require manual pruning.

- SVM (Özgür and Erdem, 2016) constructs an N-dimensional hyperplane to optimally classify data. SVM can be linear or nonlinear. Linear SVM is used for linearly separable data, i.e., datasets that can be divided into two categories by a straight line. Nonlinear SVM is used for nonlinearly separable data. For this, we use a kernel trick that sets data points in a higher dimension where they can be separated using planes or other functions.

SVM can simplify the solution of high-dimensional problems. It is based on small-sample statistical theory, has good generalization ability, and is often used in intrusion detection. Jan et al. (2019) developed a lightweight attack-detection strategy using supervised machine learning-based SVMs to detect attempts to inject unwanted data into IoT networks. It obtains a feature pool from samples, and uses it with a label vector to train the SVM. The method has good classification accuracy and detection times. Teng et al. (2017) proposed an intrusion detection method based on SVM, which constructs four two-stage SVMs based on the structure of DT. SVM1, SVM2, SVM3, and SVM4 detect normal data, DoS/DDoS attacks, probing attacks, and R2L or U2R attacks, respectively. Experiments show that this method outperforms the method of a single SVM in terms of detection rate and recall. De la Hoz et al. (2015) proposed a hybrid statistical technique and Self Organizing Map (SOM) for network anomaly detection and classification. The method uses PCA and the Fisher discriminant ratio (FDR) for feature selection and noise removal, and probabilistic self-organizing mapping (PSOM)-based modeling of the feature space to distinguish normal and malicious traffic.

- Clustering groups objects that are more similar to each other than to objects in other groups. It is generally understood as a task to be solved rather than an algorithm. Since the concept of a cluster (i.e., the similarity between objects) cannot be described precisely, there are widely different clustering algorithms. Clustering can be considered hard or soft, according to matching rules between objects and clusters. Hard clustering strictly assigns objects to classes. The most representative algorithms are k-means clustering and k-nearest neighbor (KNN), which calculate the Euclidean distance between objects to classify clusters. Soft (or fuzzy) clustering calculates the degree (e.g., probability) of each object's belonging to a cluster. Data often cannot be divided into clearly separated clusters, and soft classification is used to obtain more flexible results. Fuzzy clustering means is a widely used soft clustering algorithm that calculates the membership coefficient of each object in each cluster based on the distances between them, which improves the clustering of complex data.

Clustering algorithms can categorized, such as connectivity-based (e.g., hierarchical), centroid-based (k-means, fuzzy c-means), distribution-based (GMM), density-based (DBSCAN), or grid-based (STING). Clustering is generally simple to implement and easy to interpret, but is sensitive to outliers, and initial values of parameters have too much influence on the results.

Peng et al. (2018) proposed a method for intrusion detection systems using small-batch k-means for clustering and PCA to reduce data dimensionality. Experimental results and time complexity analysis showed that the method is effective. Casas et al. (2012) proposed UIDS, an unsupervised network intrusion detection system capable of detecting unknown network attacks without the use of q signature, labeled traffic, or training. UIDS uses an unsupervised outlier detection method based on subspace clustering, and multiple evidence accumulation techniques to identify types of attacks.

- Naive Bayes (NB) is a probabilistic classifier based on Bayes' theorem [44]. All naȯve Bayesian classifiers are based on the principle that the value of a feature is independent of the value of any other feature, i.e.,

$$\hat{y} = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}} p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k), \tag{4}$$

where $\hat{y}$ is the conditional probability that the data belong to each class, k is the number of classes, C_k is the kth class, n is the number of features, $p(C_k)$ is the prior probability of C_k, and $p(x_i \mid C_k)$ is the conditional probability of feature x_i given class C_k. A feature distribution (i.e., an event model) or nonparametric model generated from the training set must be assumed in order to compute a class prior. The multimetric and Bernoulli distributions are usually used for discrete features, and the Gaussian distribution for continuous features. Bayesian classifiers can be trained on both labeled and unlabeled datasets by certain semi-supervised training algorithms [15].

Koc et al. (2012) proposed an approach based on the hidden naȯve Bayes (HNB) model, which can be applied to intrusion detection problems affected by dimensionality, highly correlated features, and high network data stream capacity. HNB is a data mining model that relaxes the conditional independence assumptions of the NB approach. Experimental results show that the HNB model outperforms the traditional NB model in terms of accuracy, error rate, and misclassification cost. To address the potential threat of DDoS attacks in the IoT, Mehmood et al. (2018) proposed an NB algorithm with multi-agent-based IDS (NB-MAIDS) and implemented multi-agents in the whole network.

Although the independence assumption of NB is often violated in practice, it still has relatively high accuracy. In addition, as a linear algorithm, NB has high training efficiency. These qualities have led to its widespread application as a baseline for classification problems.

- Ensemble learning combines multiple classifiers through an algorithm to find a (hopefully) better hypothesis in a mixed multiple hypothesis space. It should be noted that the combination of multiple classifiers does not guarantee better performance than the best individual classifier, but it reduces the risk of a particularly poor selection.

One of the earliest and most intuitive integration-based algorithms, bagging (bootstrap aggregating) (Breiman, 1996) obtains the diversity of classifiers by randomly drawing a subset of the entire training to train classifiers of the same type, and allows each classifier in the set to vote with the same weight to combine individual classifiers. The random forest classifier (Breiman, 2001) is a common machine learning method that combines bagging with

DTs. The boosting method recursively builds an ensemble by training a new classifier to emphasize the training data misclassified by its previous classifier. Based on this algorithm, several well-known machine learning algorithms have been proposed, such as adaptive boosting (AdaBoost) (Freund et al., 1996), gradient boost decision tree (GBDT), and extreme gradient boosting (XGBoost).

Ensemble learning improves the generalizability and accuracy of the final model by ensembling multiple classifiers, and is less likely to be overfitted. Its training and prediction speeds are naturally lower than those of a single classifier, and the interpretability of the model is largely lost in some complex ensembles (Madeh Piryonesi and El-Diraby, 2021). Singh et al. (2014) developed an RF-based DT model for the quasi-real-time peer-to-peer botnet detection problem. Li et al. (2018) proposed an artificial intelligence-based two-stage intrusion detection method that uses software-defined techniques. It uses the swarm partitioning and binary difference variants of the bat algorithm to select typical features, and RFs to classify streams by adaptively changing the weights of samples using a weighted voting mechanism. Hu et al. (2013) proposed an online intrusion detection algorithm that constructs a local parameterized detection model at each node using the online AdaBoost algorithm. A global detection model is constructed in each node using a small number of samples in the nodes, combined with the local parametric model. Experimental results show that the improved online AdaBoost has a higher detection rate and lower false-alarm rate.

- Evolutionary algorithms are global optimization algorithms inspired by biological evolution, usually the trial-and-error problem of populations. Initial candidate solutions are repeatedly updated and iterated, with poorly performing solutions removed at each generation and random variations introduced, consistent with the concept of natural selection and variation.

Most widely used are genetic algorithms, genetic programming, evolutionary algorithms, particle swarm optimization (PSO), and artificial immune systems, which differ mainly in how the iterations are performed. Genetic algorithms and genetic programming calculate a fitness value for each individual in a population, and select individuals with high fitness values for the mating pool with high probability to produce the next generation through the exchange of genetic material and mutations between individuals. The genetic algorithm considers the bit string as an individual, while genetic programming considers the program as the individual. Evolutionary algorithms generally simulate the biological learning process in nature. For example, the artificial bee colony (ABC) algorithm simulates the process of bees searching for food sources. The artificial immune system simulates the immune system function by cloning and mutating antibodies with high affinity to a “virus” (i.e., the sample to be detected) in order to iterate.

Evolutionary computation is characterized by a variety of iterative methods. The iterative approach typically requires the manual definition of multiple parameters and evaluation functions for the problem to be solved. Thus, the algorithm has problem-independent fast search capability and wide applicability, and the population-based principle brings parallelism, which increases the speed of the search for the optimal solution. However, the performance of the evolutionary computation depends strongly on the evaluation function and parameters (which are usually set empirically), which affects the efficiency of the solution. Some algorithms converge too easily to a local optimum, or even an arbitrary point, while others are poor at finding local optimum problems. Although this can be alleviated by replacing the evaluation function and parameters (Taherdangkoo et al., 2013), the “no free lunch” theorem (Wolpert and Macready, 1997) has proved that this problem has no general solution.

Khammassi and Krichen (2017) proposed a GA-LR packing method for feature selection in network intrusion detection, using a genetic algorithm-based packing method as a search strategy and logistic regression as a learning algorithm to select the best feature subset. The method effectively improves the intrusion detection performance. Hajisalem and Babaie (2018) proposed a hybrid classification method based on ABC and artificial fish swarm (AFS) algorithms, using fuzzy C-Means (FCM) clustering and relevance-based feature selection (CFS) to divide the training dataset and remove irrelevant features. Based on the selected features, if-then rules are generated by a CART technique to distinguish normal and abnormal records. The generated rules are used to train the method to the detection model. In simulations on the NSL-KDD and UNSW-NB15 datasets, the method achieved a detection rate of 99% and a false-positive rate of 0.01%.

- The DNN is an artificial neural network (ANN) with multiple layers between the input and output layers (Bengio, 2009). In a narrow sense, it is a fully connected neural network with a structure similar to a multilayer perceptron (MLP). The lower-layer neurons of a fully connected DNN can form connections with all upper-layer neurons. A DNN uses backpropagation to perform a supervised learning task with nonlinear activation functions.

Vinayakumar et al. (2019) built a DNN-based distributed deep learning model for an intrusion detection framework for real-time processing and analysis of very large-scale data. Xu et al. (2018) proposed an IDS consisting of an RNN with gated recurrent units (GRUs), MLP, and softmax module. The DNN can theoretically approximate any function (Cybenko, 1989).

- The CNN is an artificial neural network with a shared-weight structure based on convolutional kernels or filters. Inspired by biological processes (Hubel and Wiesel, 1968), a CNN slides the convolutional kernel along the input features to extract translation-equivariant responses called feature maps.

The CNN and its related architectures have received considerable attention due to their excellent performance at computer vision (He et al., 2016). Starting with LeNet-5 (LeCun et al., 1998), numerous CNN architectures, including AlexNet (Krizhevsky et al., 2012) and ResNet (He et al., 2016), have been proposed. Although CNN architectures are usually applied to CV problems, they have shown good results in IDS as well (Dong et al., 2019; Vinayakumar et al., 2017). Li et al. (2017) proposed an image conversion method for NSL-KDD data, in which CNNs automatically learn the features of graphic NSL-KDD transformations.

Compared to the DNN, the CNN’s extraction of local features reduces the number of weights, as well as the computational complexity, thus improving the training and prediction speed. However, this can lead to problems; some trained CNN models extract the features of wheels in an image and immediately judge the image as a truck.

- The RNN is a class of artificial neural network that can temporally exhibit memory behavior. This dynamic behavior is implemented by connections between nodes to form a directed graph along a time sequence (Dupond, 2019). The internal state of the RNN allows it to process variable-length input sequences.

Depending on whether the constructed graph has a loop, an RNN can be further classified as finite- or infinite-impulse (Miljanovic, 2012). Finite-pulse networks can be unrolled and replaced with strict feedforward neural networks (FNNs), while infinite-pulse recurrent networks cannot. Moreover, there can be additional stored states in an RNN, thus improving it to a network that can be implemented with time delays or feedback loops (e.g., long- and short-term memory networks).

The RNN was proposed to solve the problem that a DNN has difficulty fitting data that changes temporally. Therefore, RNNs have played an important role in areas such as natural language processing and action recognition (Tang et al., 2018). RNNs are increasingly applied to IDS, whose data mostly consist of temporally continuous data streams

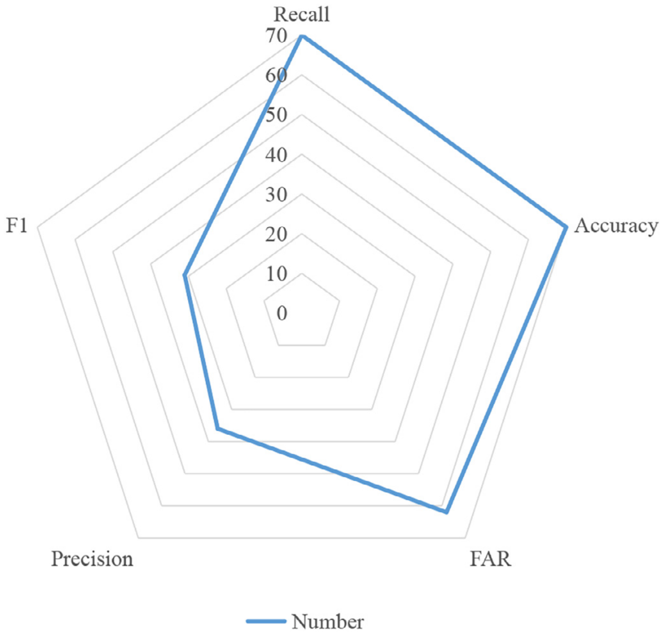

Fig. 9. Evaluation metrics used in papers.

(Hochreiter and Schmidhuber, 1997; Yin et al., 2017). However, since RNNs do not have a special treatment of the activation function, the continuous product of their partial derivatives can easily lead to gradient disappearance or even gradient explosion when the number of layers of the network is high.

- LSTM solves the gradient vanishing problem of the classical RNN by introducing additional storage states (Gers et al., 2000). LSTM effectively controls the degree of gradient vanishing by using a gate function as the activation function to selectively allow a portion of the information to pass through.

Based on the original LSTM architecture, Gers et al. (2000) introduced forgetting gates to enable the LSTM to reset its state, simulating the forgetting process of memory. Because of its excellent performance (Capes et al., 2017; Wu et al., 2016), it is considered the most classical LSTM architecture. Based on this, Cho et al. (2014) proposed a gate recurrent unit GRU consisting of a reset gate and update gate, which maintains the performance of the LSTM as much as possible with fewer parameters.

The LSTM has become one of the most used RNN variants because it solves the gradient vanishing problem of traditional RNNs. Many IDS studies use LSTM networks (Bontemps et al., 2016; Roy et al., 2017) because they are well-suited for classification and prediction based on time-series data, and the forgetting mechanism is a better match for the detection of data streams. However, due to the inherent nature of RNNs, the classical LSTM architecture cannot be trained in parallel (Bai et al., 2018), making LSTM-based models sometimes too costly to run.

4.4. Evaluation metrics

We introduce commonly used evaluation criteria in intrusion detection papers. To answer **RQ4(a)**, Fig. 9 shows evaluation metrics and the number of times they were used.

As shown in Fig. 9, accuracy, precision, recall, F1 value, and false-alarm rate (FAR) are most commonly used (**RQ4(b)**). Recall and accuracy are used in most papers. Recall, also called detection rate or true positive rate (TPR), is the proportion of correctly classified attacks to all attacks. Recall can measure the accuracy of the classifier in identifying attacks. Accuracy is the ratio of the number of correctly predicted samples to the total number of predicted samples, and it measures the overall recognition of the classifier. FAR is a critical metric to evaluate intrusion detection methods. False alarms are a manifestation of false positives, whose large number will increase the load on the system and human resources. After classification, the data can be divided into four categories: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The calculation formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TP + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$FAR = \frac{FP}{TN + FP} \tag{8}$$

$$F{-}measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{9}$$

Detection time is also a common evaluation metric in the field of intrusion detection. There are 74 articles in our research that discuss time performance. Detection time means the time spent to classify a sample with the trained model. Due to the complexity of network traffic, even with methods such as feature selection for dimensionality reduction, IDS research usually faces the problem of dimensional catastrophe, which eventually reflects as high detection time. Some of the numerous existing algorithms for intrusion detection are almost unavailable in engineering implementations, and one important reason is their high detection time. From the application point of view, the main goal of intrusion detection is to achieve an appropriate detection rate with minimal resource consumption, which requires an ideal model structure for IDS as well as parameter settings. A high detection time of a model usually means that its algorithm complexity is too high. Reviewing previous studies, a clear trade-off between the performance and complexity of the model can be found.

Although deep learning based methods usually perform better in terms of detection capability compared to other methods, their detection times are too long, making these methods difficult to use in scenarios such as big data. While computational complexity is the most direct influence on detection time, considering that the computational complexity of some algorithms is difficult to calculate or controversial under different assumptions, most papers only provide the training and testing time of their algorithms on the specified dataset. Since the platforms used to obtain each result and the preprocessing methods for the datasets differ, it is still difficult to judge the superiority of an algorithm in terms of time complexity just from the running time. In summary, we believe that there is still a need for a unified complexity evaluation standard in the current IDS research, rather than just in terms of detection time.

4.5. Authors

We assessed the main contributing authors in intrusion detection by examining the total number of citations of the included publications through Scopus, answering **RQ5(a)**. As can be seen in Fig. 10, The (Ambusaidi et al., 2016; Tan et al., 2014; Yin et al., 2017) contributed most to the field. We found that citations were not very high for all but the top few authors. This indicates that few researchers are cited.

The three most cited articles among the articles we researched are shown in Table 8. The article “A Deep Learning Approach for

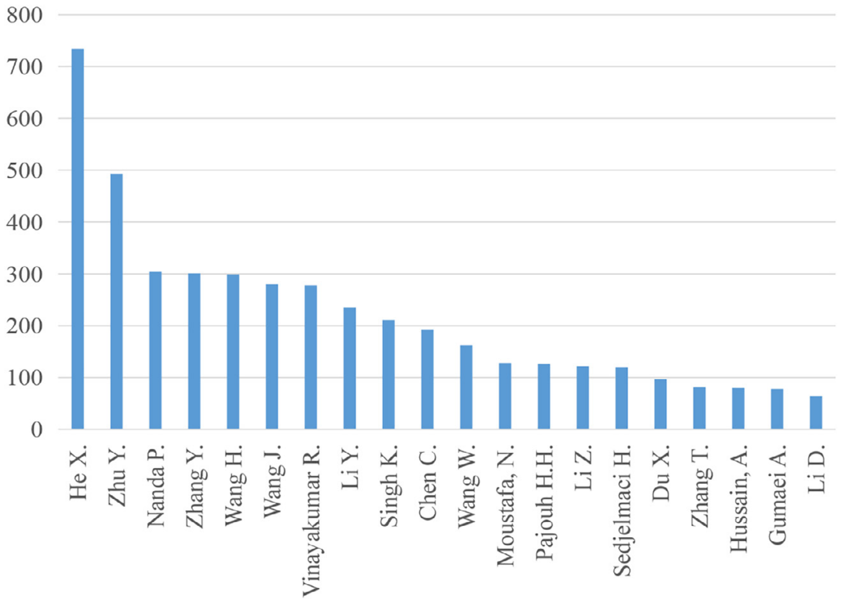

Fig. 10. Top authors by total number of Scopus citations.

Table 8
The three most cited articles.

Paper	Year	Citations	Average citations
A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks	2018	430	143
A Deep Learning Approach to Network Intrusion Detection	2017	313	78
Fuzziness based semi-supervised learning approach for intrusion detection system	2017	286	71

Table 9
Datasets used in papers.

Dataset	Count
KDD99	47
NSL-KDD	35
UNSW-NB15	8
ISCX 2012	6
Kyoto 2006+	2
Botnet	2

Intrusion Detection Using Recurrent Neural Networks” was published in 2018 and has been cited more than 430 times in total, with an average annual citation of 86. In this article, the authors propose a deep learning approach for intrusion detection using Recurrent Neural Networks (RNN-IDS). Moreover, the authors also investigate the performance of the model in binary and multiclass classification, and the effect of the number of neurons and different learning rates on the model performance. In the paper “A Deep Learning Approach to Network Intrusion Detection”, the authors also propose a deep learning-based intrusion detection model. The model is built based on stacked NDAEs and achieves excellent results. From these two highly cited articles, we can see the great impact and potential of deep learning in the field of intrusion detection. In another paper, “Fuzziness based semi-supervised learning approach for intrusion detection system”, the authors propose a fuzzy-based semi-supervised learning approach that uses unlabeled sample-assisted supervised learning algorithm to improve the performance of the classifier. Unlike the previous two papers, this paper aims to reduce the labor consumption in the data labeling process by taking the complexity of data labeling as a pain point.

To answer **RQ5(b)**, we plotted the author network, as shown in Fig. 11. As seen in Fig. 11, the distribution of author networks is dispersed. Note that the sizes of circles for these highlighted authors are scaled according to their numbers of papers, whereas the circle sizes for all other authors are fixed and small, for ease of reading. The figure includes 82 unconnected clusters, with 451 authors. The two largest clusters, as shown in Fig. 12, both contain 32 authors, representing only 14% of all authors. This indicates a low level of collaboration among authors in the community.

4.6. Datasets

To answer **RQ6(a)**, we investigated existing network intrusion detection datasets. As shown in Table 10, we collected a total of 52 datasets through the survey. And, based on the information provided by the dataset publishers and additional searches, we extracted the year of creation, creation method, data volume, annotation status, number of tags and links for each dataset. In terms of time, starting with the DARPA 1998 dataset, new datasets continue to appear in the community. With the year 2009 as the node, research related to network intrusion detection datasets started to increase. From the fourth column of Table 10, it can be seen that more than half of the datasets were obtained through simulation experiments. This reflects the sensitive nature of data in the field of network intrusion detection from the side. However, the accuracy and authenticity of such datasets have been questioned (Mahoney et al., 2003), and the validity of intrusion detection models constructed based on such datasets is poor.

Further, we summarize the frequency of use of the dataset in Table 9 (**RQ6(b)**). As can be seen from the table, KDD99 and NSL-KDD are the two most commonly used datasets, although both of them are simulation experimental data. This is mainly because

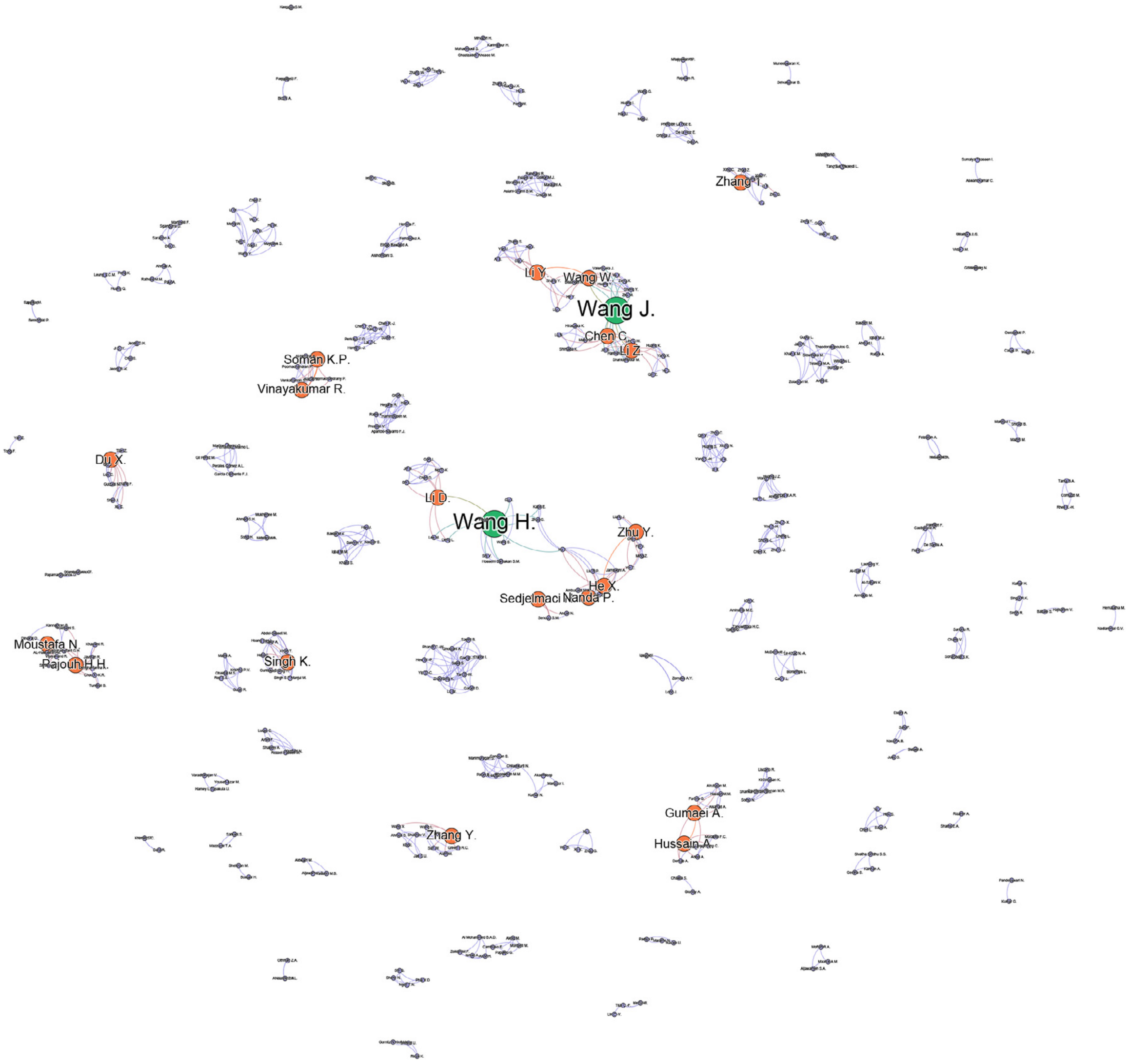

Fig. 11. Full coauthor network.

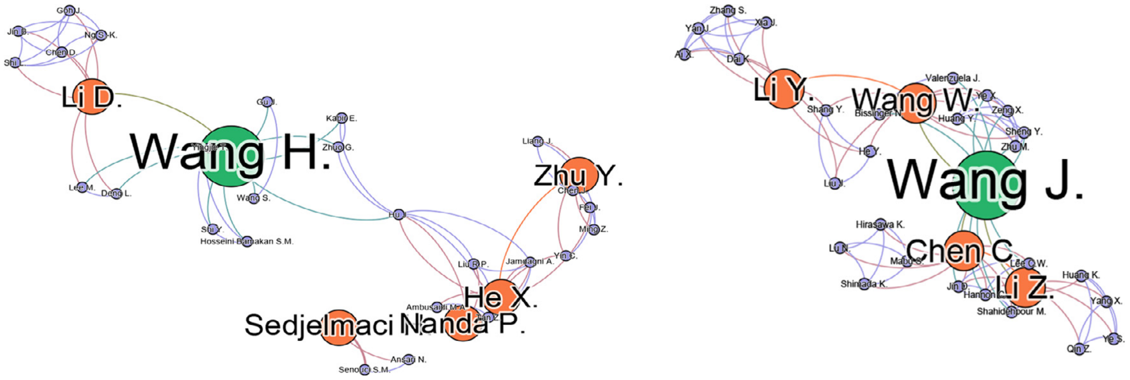

Fig. 12. Author network subgraph.

KDD99 and NSL-KDD are datasets that have been publicly available for a long time. Researchers have published many articles based on these two datasets. When a new intrusion detection technique is proposed, it often needs to be compared with previous techniques, which leads to the constant use of KDD99 and NSL-KDD (**RQ6(c)**). However, the contents of the KDD99 and NSL-KDD datasets are obsolete. In future studies, we recommend that researchers evaluate the performance of intrusion detection methods using newer intrusion detection datasets, such as the CIRA-CIC-DoHBrw 2020 dataset, by referring to the information in our table. In addition, researchers should try to experiment with some real datasets, such as the ISOT CID dataset, to ensure the validity of their approach.

Finally, to facilitate the work of researchers, we provide links to the datasets in the table and present some of the datasets in more detail.

Table 10
Existing network intrusion detection datasets.

No.	Dataset	Year	Authenticity	Count	Labeled	Number of labels	Link
1	1998 DARPA	1998	emulated	7,000,000	yes	4	DARPA (1998,1999)
2	1999 DARPA	1999	emulated	huge	yes	4	DARPA (1998,1999)
3	KDD99	1999	emulated	5,000,000	yes	4	KDD99 (1999)
4	2000 DARPA	2000	emulated	huge	yes	4	DARPA (1998,1999)
5	DEFCON	2000	real	unknown	yes	unknown	DEFCON (2000)
6	Kyoto 2006+Song et al. (2006)	2006	real	unknown	yes	unknown	Kyoto-2006+ (2006)
7	NSL-KDD Tavallaee et al. (2009)	2009	emulated	148,517	yes	4	NSL-KDD (2009)
8	LDID	2009	emulated	huge	no	unknown	unknown
9	ICML-09 Ma et al. (2009)	2009	real	2,400,000	yes	1	ICML-09 (2009)
10	Twente Sperotto et al. (2009)	2009	emulated	unknown	yes	unknown	Twente (2009)
11	CDX	2009	real	5771	yes	2	CDX (2009)
12	ISOT Botnet	2010	real	1,675,424	yes	unknown	ISOT-Botnet (2010)
13	CSIC HTTP 2010	2010	emulated	223,585	yes	1	CSIC-HTTP-2010 (2010)
14	SSENet-2011	2011	real	unknown	yes	3	unknown
15	ISCX-IDS 2012 Shiravi et al. (2012)	2012	real	2,450,324	yes	unknown	ISCX-IDS-2012 (2012)
16	ADFA-LD Creech and Hu (2013)	2013	emulated	5266	yes	6	ADFA-LD (2013)
17	CTU-13	2014	real	huge	yes	7	CTU-13 (2014)
18	Botnet 2014 Beigi et al. (2014)	2014	real	283,770	yes	16	Botnet-2014 (2014)
19	SANTA	2014	real	unknown	yes	6	unknown
20	MAWILab	2014	emulated	unknown	yes	3	MAWILab (2014)
21	SSENet-2014 Bhattacharya and Selvakumar (2014)	2014	real	201,707	yes	3	unknown
22	SSHCure Hofstede et al. (2014)	2014	real	unknown	yes	unknown	SSHCure (2014)
23	UNSW-NB15 Moustafa and Slay (2015)	2015	emulated	2,540,044	yes	9	UNSW-NB15 (2015)
24	ISTS-12	2015	emulated	huge	no	unknown	ISTS-12 (2015)
25	AWID Kolias et al. (2015)	2015	emulated	huge	yes	16	AWID (2015)
26	UCSD Jonker et al. (2017)	2015	emulated	unknown	yes	1	UCSD (2015)
27	IRSC	2015	real	unknown	yes	unknown	unknown
28	NDSec-1 Beer et al. (2017)	2016	emulated	huge	yes	8	NDSec-1 (2016)
29	DDoS 2016 Alkasassbeh et al. (2016)	2016	emulated	734,627	yes	4	DDos-2016 (2016)
30	NGIDS-DS Haider et al. (2017)	2016	emulated	unknown	yes	8	NGIDS-DS (2016)
31	UGR'16 Cermak et al. (2018)	2016	real	unknown	yes	5	UGR'16 (2016)
32	Witty Worm	2016	real	huge	yes	unknown	unknown
33	Unified Host and Network	2016	real	unknown	yes	unknown	Host and Network (2016)
34	CDMC 2016	2016	real	61,730	yes	1	unknown
35	Kharon Kiss et al. (2016)	2016	real	55,733	yes	19	Kharon (2016)
36	CIDDS-001 Ring et al. (2017)	2017	emulated	31,959,267	yes	6	CIDDS (2017)
37	CIDDS-002 Ring et al. (2017)	2017	emulated	16,161,183	yes	5	CIDDS (2017)
38	CAIDA Jonker et al. (2017)	2017	real	huge	no	unknown	CAIDA (2017)
39	CICIDS 2017 Sharafaldin et al. (2018)	2017	emulated	2,830,743	yes	7	CICIDS-2017 (2017)
40	NCCDC	2017	real	unknown	yes	unknown	unknown
41	CICDoS 2017 Jazi et al. (2017)	2017	emulated	32,925	yes	8	CICDDoS-2019 (2019)
42	TRAbID	2017	emulated	huge	yes	2	TRAbID (2017)
43	SUEE 2017	2017	emulated	19,301,217	yes	3	unknown
44	ISOT HTTP Botnet	2017	emulated	huge	yes	9	ISOT HTTP Botnet (2017)
45	PUF	2018	emulated	6,000,000	yes	4	unknown
46	ISOT CID	2018	real	36,938,985	yes	18	ISOT-CID (2018)
47	CICDDoS 2019 Sharafaldin et al. (2019)	2019	emulated	huge	yes	11	CICDDoS-2019 (2019)
48	BoT-IoT Koroniotis et al. (2019)	2019	real	73,360,900	yes	2	BoT-IoT (2019)
49	IoT-23	2020	real	unknown	yes	20	IoT-23 (2020)
50	InSDN	2020	emulated	343,939	yes	7	InSDN (2020)
51	CIRA-CIC-DoHBrw 2020 MontazeriShatoori et al. (2020)	2020	emulated	1,185,286	yes	3	CIRA-CIC-DoHBrw-2020 (2020)
52	OPCUA	2020	emulated	107,634	yes	3	OPCUA, 2020

- DARPA datasets are most popular for intrusion detection, and were created at the MIT Lincoln Laboratory in an emulated network environment. The DARPA 1998 and DARPA 1999 datasets contain seven and five weeks, respectively, of network traffic in packet-based format, including such attacks as DoS, buffer overflow, port scans, and rootkits. Despite (or because of) their wide distribution, the datasets are often criticized for artificial attack injections or large amounts of redundancy.

- KDD99 dataset was created from DARPA network dataset files by Lee and Stolfo (2000). The dataset was constructed through data mining to analyze the features of the DARPA dataset and preprocess the data. The dataset contains seven weeks of network traffic, with approximately 4.9 million vectors. Attacks are classified as: (1) user-to-root (U2R); (2) remote-to-local (R2L); (3) probing; and (4) DoS. Each instance is represented by 41 features in three categories: (1) basic; (2) traffic; and (3) content. Basic features are extracted from TCP/IP connections. Traffic characteristics are grouped into those with the same host characteristics or the same service characteristics. Content characteristics relate to suspicious behavior of the data part. This is the most extensive dataset used to evaluate intrusion detection models.

- NSL-KDD is a dataset suggested to solve some of the inherent problems of the KDD99 dataset. Although, this new version of the KDD dataset still suffers from some of the problems discussed by Tavallaee et al. (2009) and may not be a perfect representative of existing real networks, because of the lack of public data sets for network-based IDSs, we believe it still can be applied as an effective benchmark dataset to help researchers compare different intrusion detection methods. Furthermore, the number of records in the NSL-KDD train and test sets are reasonable. This advantage makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research work will be consistent and comparable.

- UNSW-NB15 was created by the Cyber Range Laboratory of the Australian Cyber Security Center. It is widely used due to its variety of novel attacks. Types of attacks consist of Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. It has a training set with 82,332 records, and a testing set with 175,341 records.

- CICIDS2017 contains benign and common attacks, with both source data (PCAPs) and results of network traffic analysis (CSV files) based on timestamps, source and destination IPs, source and destination ports, protocols, and token flows of attacks. The researchers used the B-Profile system (Sharafaldin, et al. 2016) to analyze the abstract behavior of human interactions and to generate benign background traffic. The dataset includes abstracted behaviors of 25 users based on HTTP, HTTPS, FTP, SSH, and email protocols. Brute force cracking attacks include FTP, SSH, DoS, Heartbleed, web attack, infiltration, botnet, and DDoS.

- CICDoS2017 is a publicly available intrusion detection dataset with application layer DoS attacks from the Canadian Institute for Cybersecurity. The authors executed eight DoS attacks on the application layer. Normal user behavior was generated by combining the resulting traces with attack-free traffic from the ISCX 2012 dataset. The dataset is available in packet-based format and contains 24 h of network traffic.

- CICDDoS2019 contains the latest DDoS attacks, which are similar to real-world data. It includes the results of network traffic analysis using CICFLOWMeter-V3, which contains a token flow based on timestamp source, and destination IPS source and port protocols and attacks.

- Kyoto 2006+ is a publicly available honeypot dataset of real network traffic that includes only a small number and small range of realistic, normal user behavior. The researchers transformed packet-based traffic into a new format called sessions. Each session has 24 attributes, 14 of which are statistical information features inspired by the KDD CUP 99 dataset, and the remaining 10 attributes are typical traffic-based attributes such as IP address (anonymous), port, and duration. The data were collected over three years and include approximately 93 million sessions.

- NDSec-1 contains trace and log files of network attacks synthesized by researchers from network facilities. It is publicly available, and was captured in packet-based format in 2016. It contains additional syslog and Windows eventlog information. Attack compositions include botnet, brute force (against FTP, HTTP, and SSH), DoS (HTTP, SYN, and UDP flooding), exploits, port scans, spoofing, and XSS/SQL injection.

- CTU-13 was captured in 2013 and is available in packet, unidirectional flow, and bidirectional flow formats. Captured in a university network, its 13 scenarios include different botnet attacks. Additional information about infected hosts is provided at the website.[3] Traffic was labeled in three stages: 1) all traffic to and from infected hosts was labeled as a botnet; 2) traffic matching specific filters was labeled as normal; 3) remaining traffic was labeled as background. Consequently, background traffic can be normal or malicious.

- BoT-IoT contains more than 72 million records, including DDoS, DoS, OS, service scan, keylogging, and data exfiltration attacks. The Node-red tool was used to simulate the network behavior of IoT devices. MQTT, a lightweight communication protocol, links machine-to-machine (M2M) communications. The testbed IoT scenarios are weather station, smart fridge, motion activated lights, remotely activated garage door, and smart thermostat.

- IoT-23 consists of 23 network captures (called scenarios) of IoT traffic, including 20 (PCAP files) from infected IoT devices and three of real IoT network traffic. Raspberry Pi malware was executed in each malicious scenario using several protocols and performing different actions. The network traffic capture for benign scenarios was obtained from the network traffic of three real IoT devices: a Philips HUE smart LED lamp, Amazon Echo home intelligent personal assistant, and Somfy smart door lock. Both malicious and benign scenarios were run in a controlled network environment with unrestrained internet connection, like any real IoT device.

- PUF was captured over three days from a campus network and contains exclusively DNS connections, where 38,120 of 298,463 unidirectional flows are malicious. All flows are labeled using logs of an intrusion prevention system. IP addresses were removed for privacy reasons.

- LBNL was created to analyze the network traffic characteristics in an enterprise network. The dataset can be used as background traffic for security research, as it contains almost exclusively normal user behavior. The dataset is not labeled, is anonymized, and contains more than 100 h of network traffic in packet-based format. The dataset can be downloaded at the website.[4]

- The IEEE 300-bus power test system provides the topological and electrical structure of a power grid, to be used to detect false data injection attacks in the smart grid. The system has 411 branches, and an average degree of 2.74. For details about this standard test system, we refer the reader to the work of Hines et al. (2010). The IEEE 300-bus power test system has been used in much work related to cyber-attack classification.

- The ICS cyber attack datasets consist of: (1) power system dataset; (2) gas pipeline dataset; (3) energy management system dataset; (4) new gas pipeline dataset; and (5) gas pipeline and water storage tank dataset. The power system dataset contains 37 scenarios divided into eight natural events, one non-event, and 28 attacks. Attacks are categorized as: (1) relay setting change; (2) remote tripping command injection; and (3) data injection. These datasets can be used for cybersecurity intrusion detection in industrial control systems.

5. Conclusion

We provided a comprehensive overview and analysis of research work on intrusion detection in network security. The survey covered 119 of the most highly cited papers in the field of network security intrusion detection, including preprocessing and intrusion detection techniques, and analyzed the community from multiple perspectives. We analyzed the research progress and bottlenecks in different scenarios. We investigated preprocessing and intrusion detection techniques. We examined evaluation methods, including metrics and datasets, so as to standardize performance assessment. We counted contributors in the community and mapped their collaborative network. Our publication data and category descriptions are publicly available to facilitate repeatability and further research.

Our results show that research on network anomaly detection is unbalanced under different target networks. In the ICN domain, researchers often do not disclose their datasets due to the sensitivity and confidentiality of industrial network data. The lack of available datasets limits cybersecurity research in the ICN domain. The lack of datasets is also a key factor limiting research in the SDN domain. Before conducting security research, researchers often need to build SDN network environments to simulate the data. In terms of the current means of intrusion detection techniques used, supervised learning is still the mainstream direction. However, these studies need to be built on top of the already labeled data. At the time of practical application, the data we obtain is unlabeled. La-

[3] http://mcfp.weebly.com/the-ctu-13-dataset-a-labeled-dataset-with-botnet-normal-and-background-traffic.html.

[4] http://icir.org/enterprise-tracing/download.html.

beling the data is a time-consuming and tedious task. We believe that unsupervised learning and semi-supervised learning are the way forward for network anomaly detection. Similarly, we believe that automated labeling of network data is also a direction worthy of in-depth study. In addition, the adversarial environment has been shown to impact machine learning-based network anomaly detection algorithms. Therefore, anti-perturbation anomaly detection in adversarial environments also needs more research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (No. 61902010), the Major Research Plan of National Natural Science Foundation of China (92167102), the Project of Beijing Municipal Education Commission (No. KM202110005025).

References

Abbes, T., Bouhoula, A., Rusinowitch, M., 2010. Efficient decision tree for protocol analysis in intrusion detection. Int. J. Secur. Netw. 5 (4), 220–235.

ADFA-LD, 2013. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-IDS-Datasets/.

Ahmed, M., Mahmood, A.N., Hu, J., 2016. A survey of network anomaly detection techniques. J. Netw. Comput. Appl. 60, 19–31.

Alhajjar, E., Maxwell, P., Bastian, N., 2021. Adversarial machine learning in network intrusion detection systems. Expert Syst Appl 186, 115782.

Alkasassbeh, M., Al-Naymat, G., Hassanat, A., Almseidin, M., 2016. Detecting distributed denial of service attacks using data mining techniques. Int. J. Adv. Comput. Sci. Appl. 7 (1), 436–445.

Ambusaidi, M.A., He, X., Nanda, P., Tan, Z., 2016. Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans. Comput. 65 (10), 2986–2998.

An, J., Cho, S., 2015. Variational autoencoder based anomaly detection using reconstruction probability. Spec. Lect. IE 2 (1), 1–18.

Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G., Burnap, P., 2019. A supervised intrusion detection system for smart home IoT devices. IEEE Internet Things J. 6 (5), 9042–9053.

AWID, 2015. http://icsdweb.aegean.gr/awid/download.html.

Axelsson, S., 2000. Intrusion Detection Systems: A Survey and Taxonomy. Technical Report.

Bach, F.R., 2008. Bolasso: model consistent Lasso estimation through the bootstrap. In: Proceedings of the 25th international conference on Machine learning, pp. 33–40.

Bai, S., Kolter, J. Z., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Beer, F., Hofer, T., Karimi, D., Bühler, U., 2017. A new attack composition for network security. 10. DFN-Forum Kommunikationstechnologien.

Beigi, E.B., Jazi, H.H., Stakhanova, N., Ghorbani, A.A., 2014. Towards effective feature selection in machine learning-based botnet detection approaches. In: 2014 IEEE Conference on Communications and Network Security, pp. 247–255.

Bengio, Y., 2009. Learning Deep Architectures for AI. Now Publishers Inc.

Bermingham, M.L., Pong-Wong, R., Spiliopoulou, A., Hayward, C., Rudan, I., Campbell, H., Wright, A.F., Wilson, J.F., Agakov, F., Navarro, P., et al., 2015. Application of high-dimensional feature selection: evaluation for genomic prediction in man. Sci. Rep. 5 (1), 1–12.

Bhattacharya, S., Selvakumar, S., 2014. SSENet-2014 dataset: a dataset for detection of multiconnection attacks. In: 2014 3rd International Conference on Eco-friendly Computing and Communication Systems, pp. 121–126.

Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K., 2013. Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. 16 (1), 303–336.

Bontemps, L., McDermott, J., Le-Khac, N.-A., et al., 2016. Collective anomaly detection based on long short-term memory recurrent neural networks. In: International Conference on Future Data and Security Engineering, pp. 141–152.

BoT-IoT, 2019. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php.

Botnet-2014, 2014. https://www.unb.ca/cic/datasets/botnet.html.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Buczak, A.L., Guven, E., 2015. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutor. 18 (2), 1153–1176.

Bulavas, V., 2018. Investigation of network intrusion detection using data visualization methods, 1–6.

CAIDA, 2017. https://www.impactcybertrust.org/dataset_view?idDataset=834.

Caminero, G., Lopez-Martin, M., Carro, B., 2019. Adversarial environment reinforcement learning algorithm for intrusion detection. Comput. Netw. 159, 96–109.

Capes, T., Coles, P., Conkie, A., Golipour, L., Hadjitarkhani, A., Hu, Q., Huddleston, N., Hunt, M., Li, J., Neeracher, M., et al., 2017. Siri on-device deep learning-guided unit selection text-to-speech system. In: INTERSPEECH, pp. 4011–4015.

Casas, P., Mazel, J., Owezarski, P., 2012. Unsupervised network intrusion detection systems: detecting the unknown without knowledge. Comput. Commun. 35 (7), 772–783.

CDX, 2009. https://www.usma.edu/centers-and-research/cyber-research-center/data-sets.

Cermak, M., Jirsik, T., Velan, P., Komarkova, J., Spacek, S., Drasar, M., Plesnik, T., 2018. Towards provable network traffic measurement and analysis via semi-labeled trace datasets. In: 2018 Network Traffic Measurement and Analysis Conference (TMA), pp. 1–8.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

CICDDoS-2019, 2019. https://www.unb.ca/cic/datasets/ddos-2019.html.

CICIDS-2017, 2017. https://www.unb.ca/cic/datasets/ids-2017.html.

CIDDS, 2017. http://www.hs-coburg.de/cidds.

CIRA-CIC-DoHBrw-2020, 2020. https://www.unb.ca/cic/datasets/dohbrw-2020.html.

Creech, G., Hu, J., 2013. Generation of a new IDS test dataset: time to retire the KDD collection. In: 2013 IEEE Wireless Communications and Networking Conference (WCNC), pp. 4487–4492.

CSIC-HTTP-2010, 2010. https://petescully.co.uk/research/csic-2010-http-dataset-in-csv-format-for-weka-analysis/.

CTU-13, 2014. http://mcfp.weebly.com/.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Math. Control Signals Syst. 2 (4), 303–314.

DARPA, 1998,1999. http://www.tp-ontrol.hu/index.php/TP_Toolbox.

DDos-2016, 2016. www.researchgate.net/publication/292967044_Dataset-_Detecting_Distributed_Denial_of_Service_Attacks_Using_Data_Mining_Techniques.

DEFCON, 2000. https://defcon.org/html/links/dc-ctf.html.

Deng, H., Runger, G., Tuv, E., 2011. Bias of importance measures for multi-valued attributes and solutions. In: International Conference on Artificial Neural Networks, pp. 293–300.

Dong, Y., Wang, R., He, J., 2019. Real-time network intrusion detection system based on deep learning. In: 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), pp. 1–4.

Dupond, S., 2019. A thorough review on the current advance of neural network structures. Annu. Rev. Control 14, 200–230.

Ertekin, S., Huang, J., Bottou, L., Giles, L., 2007. Learning on the border: active learning in imbalanced data classification. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 127–136.

Estabrooks, A., Jo, T., Japkowicz, N., 2004. A multiple resampling method for learning from imbalanced data sets. Comput. Intell. 20 (1), 18–36.

Fernández, A., Garcia, S., Herrera, F., Chawla, N.V., 2018. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J. Artif. Intell. Res. 61, 863–905.

Freund, Y., Schapire, R.E., et al., 1996. Experiments with a new boosting algorithm. In: International Conference on Machine Learning, vol. 96, pp. 148–156.

Gers, F.A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: continual prediction with LSTM. Neural Comput. 12 (10), 2451–2471.

Ghorbani, A.A., Lu, W., Tavallaee, M., 2009. Network Intrusion Detection and Prevention: Concepts and Techniques, vol. 47. Springer Science & Business Media.

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep Learning, vol. 1. MIT Press Cambridge.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3 (Mar), 1157–1182.

Haider, W., Hu, J., Slay, J., Turnbull, B.P., Xie, Y., 2017. Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. J. Netw. Comput. Appl. 87, 185–192.

Hajisalem, V., Babaie, S., 2018. A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. Comput. Netw. 136, 37–50.

Hamid, Y., Sugumaran, M., 2020. A t-SNE based non linear dimension reduction for network intrusion detection. Int. J. Inf. Technol. 12 (1), 125–134.

Hande, Y., Muddana, A., 2021. A survey on intrusion detection system for software defined networks (SDN). In: Research Anthology on Artificial Intelligence Applications in Security. IGI Global, pp. 467–489.

Haq, N.F., Onik, A.R., Hridoy, M.A.K., Rafni, M., Shah, F.M., Farid, D.M., 2015. Application of machine learning approaches in intrusion detection system: a survey. IJARAI-Int. J. Adv. Res. Artif. Intell. 4 (3), 9–18.

He, H., Bai, Y., Garcia, E.A., Li, S., 2008. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Hines, P., Blumsack, S., Sanchez, E.C., Barrows, C., 2010. The topological and electrical structure of power grids. In: 2010 43rd Hawaii International Conference on System Sciences, pp. 1–10.

Hinton, G., Roweis, S.T., 2002. Stochastic neighbor embedding. In: NIPS, vol. 15. Citeseer, pp. 833–840.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., Atkinson, R., 2017. Shallow and deep networks intrusion detection system: a taxonomy and survey. arXiv preprint arXiv:1701.02145.

Hofstede, R., Hendriks, L., Sperotto, A., Pras, A., 2014. SSH compromise detection using NetFlow/IPFIX. ACM SIGCOMM Comput. Commun. Rev. 44 (5), 20–26.

Host, U., Network, 2016. https://csr.lanl.gov/data/cyber1/.

De la Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., Prieto, B., 2015. PCA filtering and probabilistic SOM for network intrusion detection. Neurocomputing 164, 71–81.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al., 2003. A practical guide to support vector classification.

Hu, W., Gao, J., Wang, Y., Wu, O., Maybank, S., 2013. Online adaboost-based parameterized methods for dynamic distributed network intrusion detection. IEEE Trans. Cybern. 44 (1), 66–82.

Hubel, D.H., Wiesel, T.N., 1968. Receptive fields and functional architecture of monkey striate cortex. J. Physiol. 195 (1), 215–243.

ICML-09, 2009. http://www.sysnet.ucsd.edu/projects/url/.

InSDN, 2020. http://aseados.ucd.ie/?p=177.

IoT-23, 2020. https://mcfp.felk.cvut.cz/publicDatasets/IoT-23-Dataset/iot_23_datasets_small.tar.gz.

ISCX-IDS-2012, 2012. https://www.unb.ca/cic/datasets/ids.html.

ISOT-Botnet, 2010. https://www.uvic.ca/engineering/ece/isot/datasets/botnet-ransomware/index.php.

ISOT-CID, 2018. https://www.uvic.ca/engineering/ece/isot/datasets/cloud-security/index.php.

ISTS-12, 2015. http://ists.sparsa.org/.

ISOT, 2017. https://www.uvic.ca/engineering/ece/isot/datasets/botnet-ransomware/index.php.

Jan, S.U., Ahmed, S., Shakhov, V., Koo, I., 2019. Toward a lightweight intrusion detection system for the internet of things. IEEE Access 7, 42450–42471.

Jazi, H.H., Gonzalez, H., Stakhanova, N., Ghorbani, A.A., 2017. Detecting http-based application layer dos attacks on web servers in the presence of sampling. Comput. Netw. 121, 25–36.

Jonker, M., King, A., Krupp, J., Rossow, C., Sperotto, A., Dainotti, A., 2017. Millions of targets under attack: a macroscopic characterization of the dos ecosystem. In: Proceedings of the 2017 Internet Measurement Conference, pp. 100–113.

KDD99, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Keele, S., et al., 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report. Citeseer.

Khammassi, C., Krichen, S., 2017. A GA-LR wrapper approach for feature selection in network intrusion detection. Comput. Secur. 70, 255–277.

Kharon, 2016. http://kharon.gforge.inria.fr/dataset/index.html.

Kiss, N., Lalande, J.-F., Leslous, M., Tong, V.V.T., 2016. Kharon dataset: android malware under a microscope. In: The {LASER} Workshop: Learning from Authoritative Security Experiment Results ({LASER} 2016), pp. 1–12.

Koc, L., Mazzuchi, T.A., Sarkani, S., 2012. A network intrusion detection system based on a Hidden Naïve bayes multiclass classifier. Expert Syst. Appl. 39 (18), 13492–13500.

Kolias, C., Kambourakis, G., Stavrou, A., Gritzalis, S., 2015. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Commun. Surv. Tutor. 18 (1), 184–208.

Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B., 2019. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-IoT dataset. Future Gener. Comput. Syst. 100, 779–796.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1097–1105.

Kyoto-2006+, 2006. http://www.takakura.com/Kyoto_data/.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

Lee, W., Stolfo, S.J., 2000. A framework for constructing features and models for intrusion detection systems. ACM Trans. Inf. Syst. Secur.(TiSSEC) 3 (4), 227–261.

Li, J., Zhao, Z., Li, R., Zhang, H., 2018. Ai-based two-stage intrusion detection for software defined IoT networks. IEEE Internet Things J. 6 (2), 2093–2102.

Li, Z., Qin, Z., Huang, K., Yang, X., Ye, S., 2017. Intrusion detection using convolutional neural networks for representation learning. In: International Conference on Neural Information Processing, pp. 858–866.

Liu, X.-Y., Wu, J., Zhou, Z.-H., 2008. Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Man Cybern. Part B 39 (2), 539–550.

Loh, W.-Y., 2011. Classification and regression trees. Wiley Interdiscip. Rev. Data Min.Knowl. Discov. 1 (1), 14–23.

Ma, J., Saul, L.K., Savage, S., Voelker, G.M., 2009. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1245–1254.

Madeh Piryonesi, S., El-Diraby, T.E., 2021. Using machine learning to examine impact of type of performance indicator on flexible pavement deterioration modeling. J. Infrastruct. Syst. 27 (2), 04021005.

Mahoney, Matthew, V., Philip, K., Chan, 2003. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In: International Workshop on Recent Advances in Intrusion Detection. Springer, Berlin, Heidelberg, pp. 220–237.

Mani, I., Zhang, I., 2003. kNN approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of Workshop on Learning from Imbalanced Datasets, vol. 126.

Martinez, A.M., Kak, A.C., 2001. PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell. 23 (2), 228–233.

MAWILab, 2014. http://www.fukuda-lab.org/mawilab/documentation.html.

McCarthy, K., Zabar, B., Weiss, G., 2005. Does cost-sensitive learning beat sampling for classifying rare classes? In: Proceedings of the 1st International Workshop on Utility-Based Data Mining, pp. 69–77.

Mehmood, A., Mukherjee, M., Ahmed, S.H., Song, H., Malik, K.M., 2018. NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing iot against DDoS attacks. J. Supercomput. 74 (10), 5156–5170.

Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A., Payne, B.D., 2015. Evaluating computer intrusion detection systems: asurvey of common practices. ACM Comput. Surv. (CSUR) 48 (1), 1–41.

Miljanovic, M., 2012. Comparative analysis of recurrent and finite impulse response neural networks in time series prediction. Indian J. Comput. Sci. Eng. 3 (1), 180–191.

Mishra, P., Varadharajan, V., Tupakula, U., Pilli, E.S., 2018. A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Commun. Surv. Tutor. 21 (1), 686–728.

MontazeriShatoori, M., Davidson, L., Kaur, G., Lashkari, A.H., 2020. Detection of DoH tunnels using time-series classification of encrypted traffic. In: 2020 IEEE Intl. Conf. on Dependable, Autonomic and Secure Computing, Intl. Conf. on Pervasive Intelligence and Computing, Intl. Conf. on Cloud and Big Data Computing, Intl. Conf. on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp. 63–70.

Moustafa, N., Slay, J., 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6.

Muniyandi, A.P., Rajeswari, R., Rajaram, R., 2012. Network anomaly detection by cascading k-means clustering and C4. 5 decision tree algorithm. Procedia Eng. 30, 174–182.

NDSec-1, 2016. https://www2.hs-fulda.de/NDSec/NDSec-1/Files/.

NGIDS-DS, 2016. research.unsw.edu.au/people/professor-jiankun-hu.

Nisioti, A., Mylonas, A., Yoo, P.D., Katos, V., 2018. From intrusion detection to attacker attribution: acomprehensive survey of unsupervised methods. IEEE Commun. Surv. Tutor. 20 (4), 3369–3388.

NSL-KDD, 2009. https://www.unb.ca/cic/datasets/nsl.html.

OPCUA, 2020. https://digi2-feup.github.io/OPCUADataset/.

Özgür, A., Erdem, H., 2016. A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ Preprints 4, e1954v1.

Peng, K., Leung, V.C., Huang, Q., 2018. Clustering approach based on mini batch Kmeans for intrusion detection system over big data. IEEE Access 6, 11897–11906.

Pyle, D., 1999. Data Preparation for Data Mining. Morgan Kaufmann.

Quinlan, J.R., 1983. Learning efficient classification procedures and their application to chess end games. Mach. Learn. 463–482.

Quinlan, J.R., 1986. Induction of decision trees. Mach. Learn. 1 (1), 81–106.

Quinlan, J.R., 2014. C4. 5: Programs for Machine Learning. Elsevier.

Raskutti, B., Kowalczyk, A., 2004. Extreme re-balancing for SVMs: a case study. ACM Sigkdd Explor. Newsl. 6 (1), 60–69.

Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A., 2017. Flow-based benchmark data sets for intrusion detection. In: Proceedings of the 16th European Conference on Cyber Warfare and Security. ACPI, pp. 361–369.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A., 2019. A survey of network-based intrusion detection data sets. Comput. Secur. 86, 147–167.

Roy, S.S., Mallik, A., Gulati, R., Obaidat, M.S., Krishna, P.V., 2017. A deep learning based artificial neural network approach for intrusion detection. In: International Conference on Mathematics and Computing, pp. 44–53.

Ruan, Z., Miao, Y., Pan, L., Patterson, N., Zhang, J., 2017. Visualization of big data security: a case study on the KDD99 cup data set. Digit. Commun. Netw. 3 (4), 250–259.

Safavian, S.R., Landgrebe, D., 1991. A survey of decision tree classifier methodology. IEEE Trans. Syst. Man Cybern. 21 (3), 660–674.

Sarangi, S., Sahidullah, M., Saha, G., 2020. Optimization of data-driven filterbank for automatic speaker verification. Digit. Signal Process. 104, 102795.

Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSp, pp. 108–116.

Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A., 2019. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST), pp. 1–8.

Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A., 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. 31 (3), 357–374.

Singh, K., Guntuku, S.C., Thakur, A., Hota, C., 2014. Big data analytics framework for peer-to-peer botnet detection using random forests. Inf. Sci. 278, 488–497.

Song, J., Takakura, H., Okabe, Y., 2006. Description of kyoto university benchmark data. Available at link: http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf [Accessed on 15 March 2016].

Sperotto, A., Sadre, R., Van Vliet, F., Pras, A., 2009. A labeled data set for flow-based intrusion detection. In: International Workshop on IP Operations and Management, pp. 39–50.

SSHCure, 2014. www.simpleweb.org/wiki/index.php.

Subba, B., Biswas, S., Karmakar, S., 2015. Intrusion detection systems using linear discriminant analysis and logistic regression. In: 2015 Annual IEEE India Conference (INDICON), pp. 1–6.

Taherdangkoo, M., Paziresh, M., Yazdi, M., Bagheri, M.H., 2013. An efficient algorithm for function optimization: modified stem cells algorithm. Cent. Eur. J. Eng. 3 (1), 36–50.

Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R.P., Hu, J., 2014. Detection of denial-of-service attacks based on computer vision techniques. IEEE Trans. Comput. 64 (9), 2519–2533.

Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M., 2018. Deep recurrent neural network for intrusion detection in SDN-based networks. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 202–206.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6.

Teng, S., Wu, N., Zhu, H., Teng, L., Zhang, W., 2017. SVM-DT-based adaptive and collaborative intrusion detection. IEEE/CAA J. Autom. Sin. 5 (1), 108–118.

Thakkar, A., Lohiya, R., 2020. A review of the advancement in intrusion detection datasets. Procedia Comput. Sci. 167, 636–645.

Ting, K.M., 2002. An instance-weighting method to induce cost-sensitive trees. IEEE Trans. Knowl. Data Eng. 14 (3), 659–665.

TRAbID, 2017. https://secplab.ppgia.pucpr.br/?q=trabid.

Twente, 2009. www.simpleweb.org/wiki/index.php.

UCSD, 2015. https://www.impactcybertrust.org/dataset_view?idDataset=915.

UGR’16, 2016. https://nesg.ugr.es/nesg-ugr16/index.php.

UNSW-NB15, 2015. https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys?path=2FUNSW-NB1520-20CSV20Files.

Van Der Maaten, L., 2014. Accelerating t-SNE using tree-based algorithms. J. Mach. Learn. Res. 15 (1), 3221–3245.

Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. IEEE Access 7, 41525–41550.

Vinayakumar, R., Soman, K., Poornachandran, P., 2017. Applying convolutional neural network for network intrusion detection. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1222–1228.

Wang, B.X., Japkowicz, N., 2004. Imbalanced data set learning with synthetic samples. In: Proc. IRIS Machine Learning Workshop, vol. 19.

Wang, L., Jones, R., 2017. Big data analytics for network intrusion detection: asurvey. Int. J. Netw.Commun. 7 (1), 24–31.

Weiss, G.M., Provost, F., 2001. The effect of class distribution on classifier learning: an empirical study.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1 (1), 67–82.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al., 2016. Google’s neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Xiao, Y., Xing, C., Zhang, T., Zhao, Z., 2019. An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access 7, 42210–42219.

Xu, C., Shen, J., Du, X., Zhang, F., 2018. An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access 6, 48697–48707.

Yang, Y., Pedersen, J.O., 1997. A comparative study on feature selection in text categorization. Icml 97 (412–420), 35.

Yao, H., Li, C., Sun, P., 2020. Using parametric t-distributed stochastic neighbor embedding combined with hierarchical neural network for network intrusion detection. Int. J. Netw. Secur. 22 (2), 265–274.

Yin, C., Zhu, Y., Fei, J., He, X., 2017. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5, 21954–21961.

Zare, H., Haffari, G., Gupta, A., Brinkman, R.R., 2013. Scoring relevancy of features based on combinatorial analysis of Lasso with application to lymphoma diagnosis. BMC Genomics 14 (1), 1–9.

Zarpelão, B.B., Miani, R.S., Kawakani, C.T., de Alvarenga, S.C., 2017. A survey of intrusion detection in internet of things. J. Netw. Comput. Appl. 84, 25–37.

Zhang, H., Wu, C.Q., Gao, S., Wang, Z., Xu, Y., Liu, Y., 2018. An effective deep learning based scheme for network intrusion detection. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 682–687.

Zhang, J., Zulkernine, M., Haque, A., 2008. Random-forests-based network intrusion detection systems. IEEE Trans. Syst. Man Cybern.Part C 38 (5), 649–659.

Zhen Yang is currently a full professor of computer science and engineering at Beijing University of Technology. He received the PhD degree in signal processing from the Beijing University of Posts and Telecommunications. His research interests include data mining, machine learning, trusted computing, and content security. He has published more than 30 papers in highly ranked journals and top conference proceedings. He is a senior Member of the Chinese Institute of Electronics and a member of the IEEE.

Xiaodong Liu is currently studying for a master’s degree in the School of Computer Science and Technology at Beijing University of Technology. Research Fields: Network Security, Intrusion Detection, Machine Learning.

Tong Li holds a lecturer position in the Faculty of Information Technology at the Beijing University of Technology, China. He received his PhD degree in Computer Science from the University of Trento in 2016. He has been an author or co-author of more than 70 papers in peer-reviewed journals, conferences, or workshops in the areas of requirements engineering, security engineering, and conceptual modeling. He is currently focusing on analyzing security requirements for social engineering attacks. He is now hosting a National Natural Science Foundation of China, a subtask of a National Key Research and Development Program of China, and a Beijing Education Science Planning Funding. He is an expert of ISO/IEC JTC 1/ SC 27/ WG 4 and works as a co-editor of ISO/IEC 24392.

Di Wu is currently pursuing the PhD degree in college of computer science and technology at Beijing University of Technology, Beijing, China. Her research interests include many-objective optimization algorithm and knowledge graph embedding.

Jinjiang Wang is a current undergraduate student majoring in information security at Beijing University of Technology, Beijing, China. His research interests include machine learning-based network intrusion detection algorithm, and reinforcement learning.

Yunwei Zhao received her PhD from Tsinghua University in 2015 and worked as a postdoctoral researcher in Nanyang Technological University afterwards. She joined CNCERT/CC in 2017. Her research interest is data analytics, network security, data interdependence, behavior modeling, and social media analytics. Her publications appear in top-tier venues including IJCAI, IJCNN, WI-IAT, etc.

Han Han is an engineer of CNCERT/CC. He specializes in software engineering, AI, and cybersecurity. His research has bridged the gap between the theory and practical usage of AI-assisted software systems for better quality assurance and security.