DEEP LEARNING (WITH APPLICATIONS IN CYBERSECURITY AND ANALYTICS)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΜΣ Κυβερνοασφαλεια και Επιστημη Δεδομενων
MSc Cybersecurity and Data Science
DEPT OF INFORMATICS
UNIVERSITY OF PIRAEUS

MACHINE LEARNING/DEEP LEARNING (ML/DL) TECHNIQUES IN INTRUSION DETECTION SYSTEMS (IDS)

ASSOC.PROF. PANAYIOTIS KOTZANIKOLAOU

ML/DL METHODS FOR INTRUSION DETECTION SYSTEMS (IDS)

Intrusion Detection Systems monitor the security state of H/W and S/W systems running in the network.

Challenges

- Accuracy: Minimize False Negatives
- Efficiency: Reduce False Positives
- Detect unknown and emerging threats.

Machine learning methods

Automatically discover the essential differences between normal data and abnormal data with high accuracy.

Generalizability: Detect unknown threats.

Deep Learning methods

A branch of ML with much higher performance

TAXONOMY OF IDS

Intrusion: An attempt for unauthorized access to data, systems or services, aiming to cause disclosure, modification or unavailability of data, systems or services.

Main IDS functions:

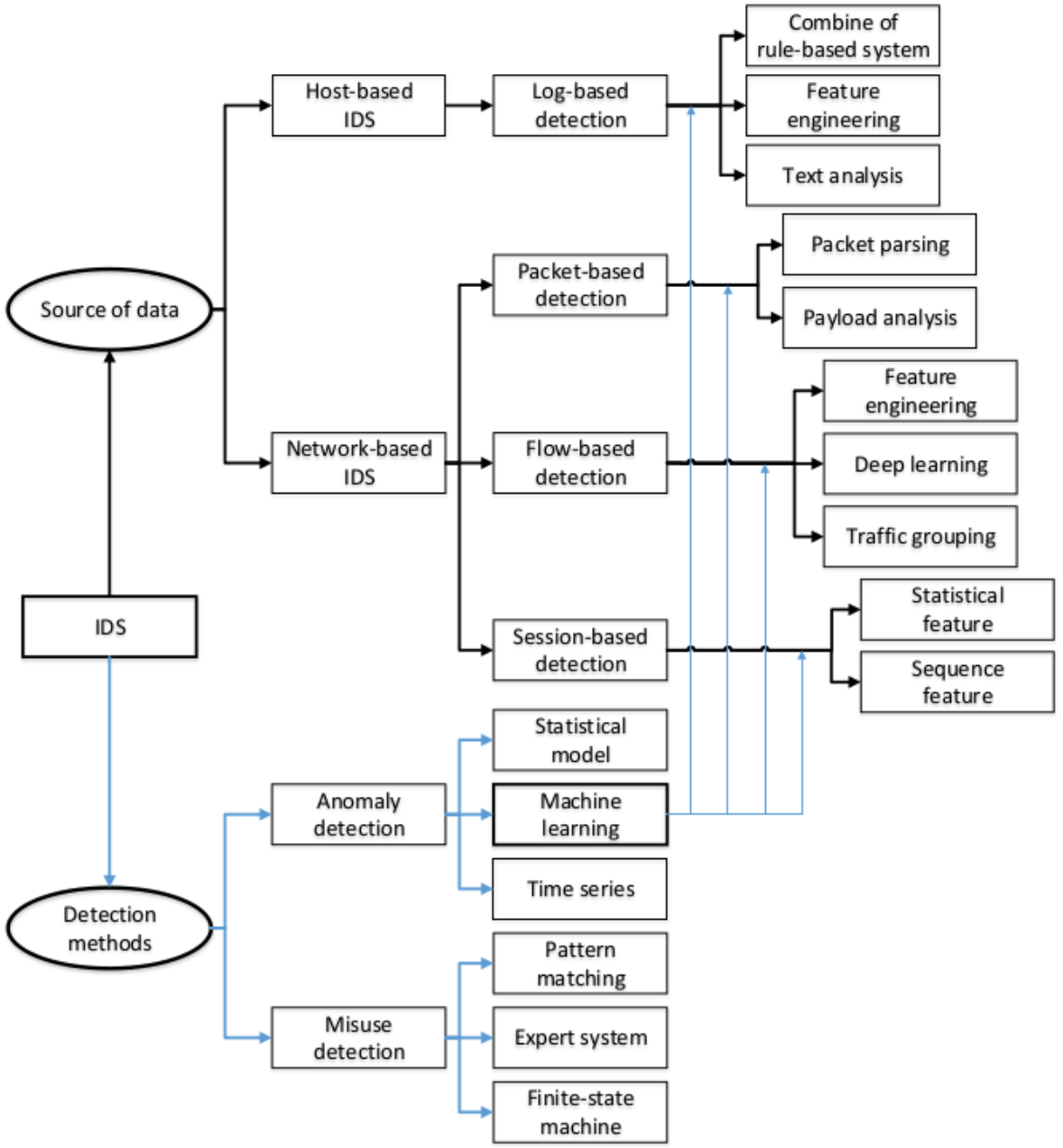
Monitor hosts

Monitor networks

Analyze system behavior

Generate alerts and/or respond to possible intrusion attempts (Intrusion Prevention Systems)

TAXONOMY OF IDS



DIFFERENCES BETWEEN MISUSE (SIGNATURE-BASED) DETECTION AND ANOMALY DETECTION

	Misuse Detection	**Anomaly Detection**
Detection performance	Low false alarm rate; High missed alarm rate	Low missed alarm rate; High false alarm rate
Detection efficiency	High, decrease with scale of signature database	Dependent on model complexity
Dependence on domain knowledge	Almost all detections depend on domain knowledge	Low, only the feature design depends on domain knowledge
Interpretation	Design based on domain knowledge, strong interpretative ability	Outputs only detection results, weak interpretative ability
Unknown attack detection	Only detects known attacks	Detects known and unknown attacks

DIFFERENCES BETWEEN HOST-BASED AND NETWORK-BASED IDS

	Host-Based IDS	**Network-Based IDS**
Source of data	Logs of operating system or application programs	Network traffic
Deployment	Every host; Dependent on operating systems; Difficult to deploy	Key network nodes; Easy to deploy
Detection efficiency	Low, must process numerous logs	High, can detect attacks in real time
Intrusion traceability	Trace the process of intrusion according to system call paths	Trace position and time of intrusion according to IP addresses and timestamps
Limitation	Cannot analyze network behaviors	Monitor only the traffic passing through a specific network segment

DEFINITIONS

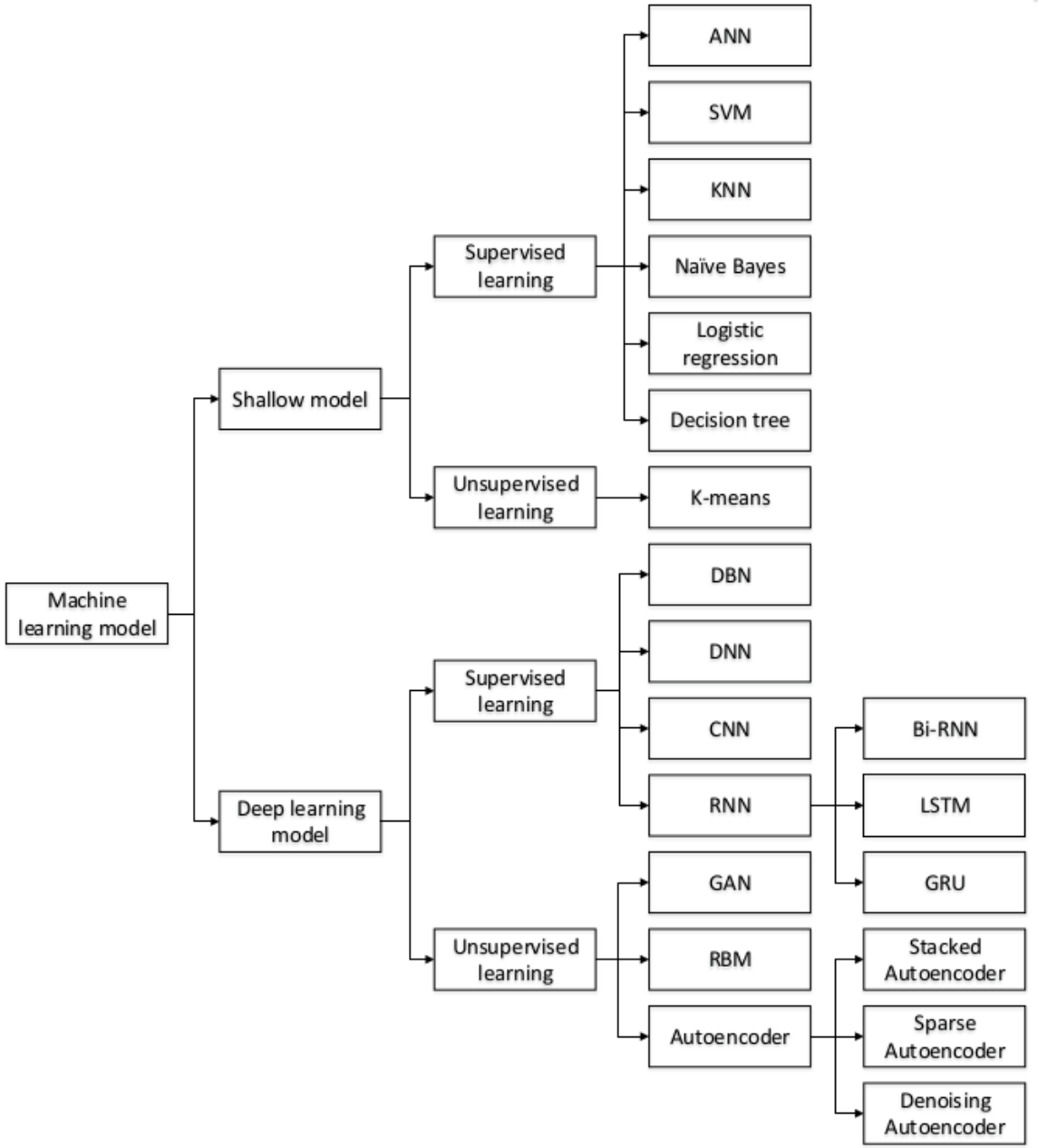
ML vs DL

- Traditional or shallow ML models, (e.g. support vector machine (SVM) and k-nearest neighbor (KNN)): they contain none or only one hidden layer.
- Deep learning: deep structure, which contains multiple hidden layers.

Supervised vs Unsupervised learning

- **Supervised learning** relies on useful information in **labeled data**.
  - Classification is the most common task in supervised learning (and is also most frequently used in IDS); however, labeling data manually is expensive and time consuming.
  - Consequently, the lack of sufficient labeled data forms the main bottleneck to supervised learning. .
- **Unsupervised learning** extracts valuable feature information from **unlabeled data**,
  - Much easier to obtain training data.
  - However, the detection performance of unsupervised learning methods is usually inferior to those of supervised learning methods

1. TAXONOMY OF ML/DL IN IDS



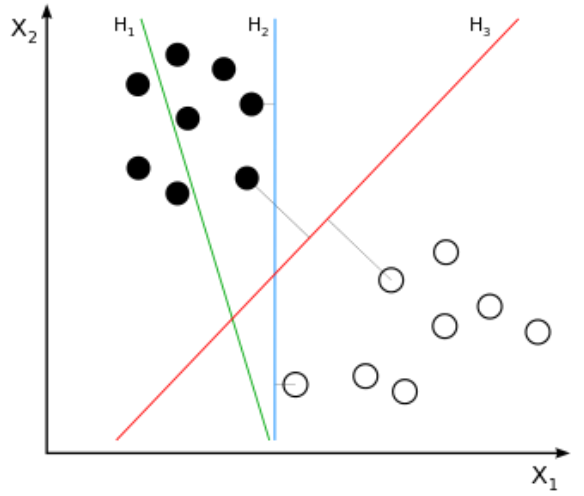
1.1. Shallow (Traditional ML) Models

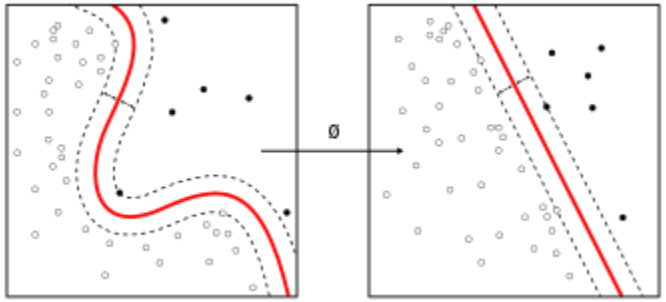
ARTIFICIAL NEURAL NETWORK (ANN)

- Mimics the way human brains work. It contains an **input layer**, **several hidden layers**, and an **output layer**.
- The units in adjacent layers are fully connected.
- An ANN contains a huge number of units and can theoretically approximate arbitrary functions; hence, it has strong fitting ability, especially for nonlinear functions.
- Due to the complex model structure, training ANNs is time-consuming. It is noteworthy that ANN models are trained by the back propagation algorithm that cannot be used to train deep networks.
- Thus, an ANN belongs to shallow models and differs from the deep learning models

SUPPORT VECTOR MACHINE (SVM)

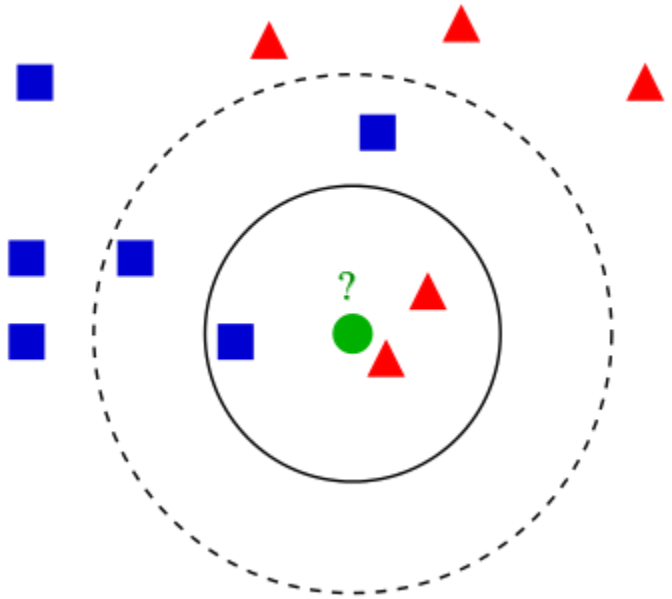
- The strategy in SVMs is to **find a max-margin separation hyperplane** in the n-dimension feature space.
- SVMs can achieve gratifying results even with small-scale training sets because the separation hyperplane is determined only by a small number of support vectors.
- However, SVMs are sensitive to noise near the hyperplane. SVMs are able to solve linear problems well. For nonlinear data, kernel functions are usually used.
- A kernel function maps the original space into a new space so that the original nonlinear data can be separated.
- Kernel tricks are widespread among both SVMs and other machine learning algorithms.





K-NEAREST NEIGHBOR (KNN)

- The core idea of KNN is based on the manifold hypothesis. If most of a sample's neighbors belong to the same class, the sample has a high probability of belonging to the class.
- Thus, the **classification result is only related to the top-k nearest neighbors.** The parameter k greatly influences the performance of KNN models. The smaller k is, the more complex the model is **and the higher the risk of overfitting.**
- Conversely, the larger k is, the simpler the model is and the weaker the fitting ability.



NAÏVE BAYES

- The Naïve Bayes algorithm is based on the **conditional probability** and the **hypothesis of attribute independence**.
- For every sample, the Naïve Bayes classifier calculates the conditional probabilities for different classes.
- The sample is classified into the maximum probability class. The conditional probability formula is calculated as:

$$P(X = x|Y = c_k) = \prod_{i=1}^{n} P(X^{(i)} = x^{(i)}|Y = c_k)$$

- When the attribute independence hypothesis is satisfied, the Naïve Bayes algorithm reaches the optimal result.
- Unfortunately, that **hypothesis is difficult to satisfy in reality**; hence, the Naïve Bayes algorithm **does not perform well on attribute-related data.**

LOGISTIC REGRESSION (LR)

- The LR is a type of logarithm linear model.
- It computes the probabilities of different classes through parametric logistic distribution, calculated as:

$$P(Y = k|x) = \frac{e^{w_k * x}}{1 + \sum_{k}^{K-1} e^{w_k * x}}$$

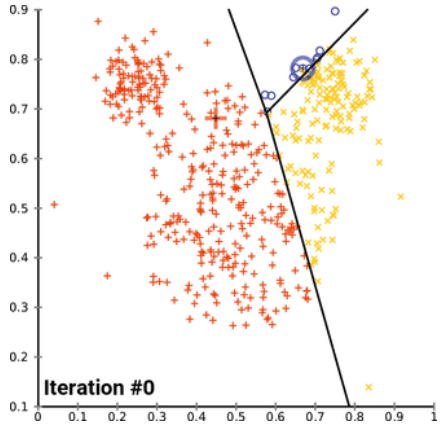
- where k = 1,2...K − 1. The sample x is classified into the maximum probability class.
- An LR model is **easy to construct, and model training is efficient.**
- However, **LR cannot deal well with nonlinear data**, which limits its application.

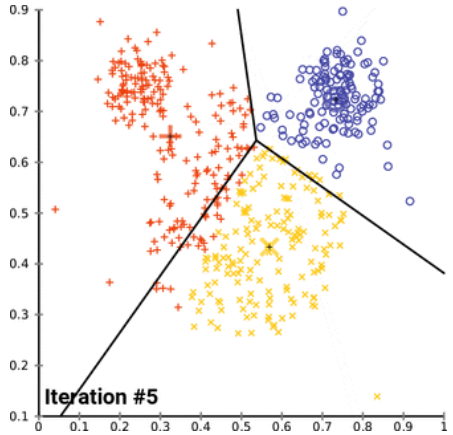
DECISION TREE

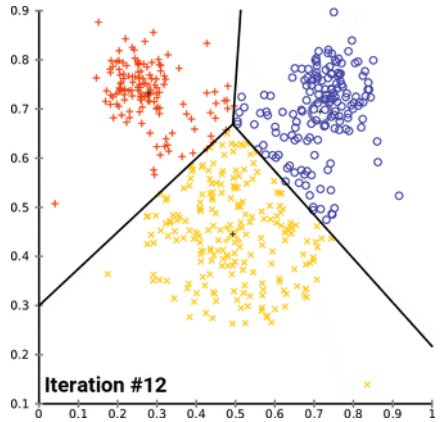
- The decision tree algorithm classifies data using a series of rules.
- The model is tree like, which makes it interpretable.
- The decision tree algorithm can automatically exclude irrelevant and redundant features.
- The learning process includes **feature selection**, **tree generation**, and **tree pruning**.
- When training a decision tree model, the algorithm selects the most suitable features individually and generates child nodes from the root node.
- The decision tree is a basic classifier.
- Some advanced algorithms, such as the random forest and the extreme gradient boosting (XGBoost), consist of multiple decision trees.

CLUSTERING

- Clustering is based on similarity theory, i.e., grouping highly similar data into the same clusters and grouping less-similar data into different clusters.
- Different from classification, **clustering is a type of unsupervised learning**. No prior knowledge or labeled data is needed for clustering algorithms; therefore, the data set requirements are relatively low.
- However, when using clustering algorithms to detect attacks, it is necessary to refer external information.
- **K-means** is a typical clustering algorithm, where **K is the number of clusters and the means is the mean of attributes.**
- The K-means algorithm **uses distance as a similarity measure criterion.** The shorter the distance between two data objects is, the more likely they are to be placed in the same cluster.
- The K-means algorithm adapts well to linear data.
- In addition, the K-means algorithm is sensitive to the initialization condition and the parameter K.
- Consequently, many repeated experiments must be run to set the proper parameter value.







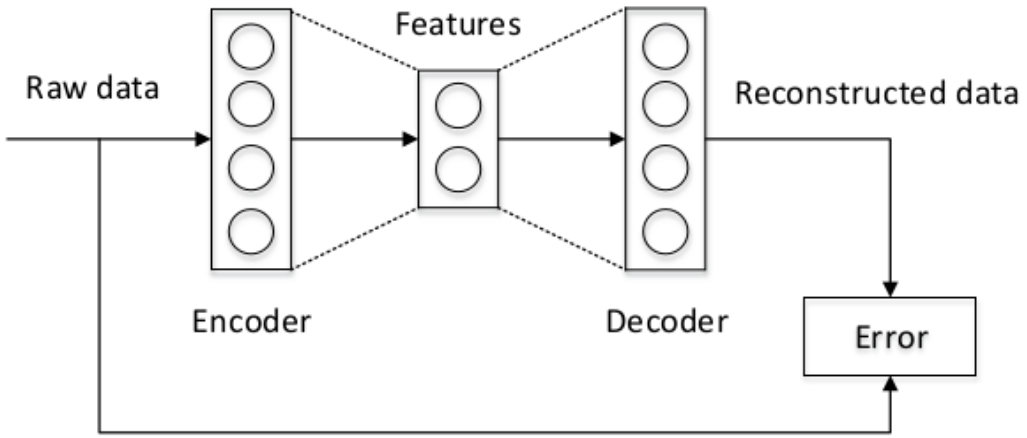
SUMMARY OF ML MODELS USED IN IDS

Algorithms	Advantages	Disadvantages	Improvement Measures
ANN	Able to deal with nonlinear data; Strong fitting ability	Apt to overfitting; Prone to become stuck in a local optimum; Model training is time consuming	Adopted improved optimizers, activation functions, and loss functions
SVM	Learn useful information from small train set; Strong generation capability	Do not perform well on big data or multiple classification tasks; Sensitive to kernel function parameters	Optimized parameters by particle swarm optimization (PSO)[8]
KNN	Apply to massive data; Suitable to nonlinear data; Train quickly; Robust to noise	Low accuracy on the minority class; Long test times; Sensitive to the parameter *K*	Reduced comparison times by trigonometric inequality; Optimized parameters by particle swarm optimization (PSO) [9]; Balanced datasets using the synthetic minority oversampling technique (SMOTE) [10]
Naïve Bayes	Robust to noise; Able to learn incrementally	Do not perform well on attribute-related data	Imported latent variables to relax the independent assumption [11]
LR	Simple, can be trained rapidly; Automatically scale features	Do not perform well on nonlinear data; Apt to overfitting	Imported regularization to avoid overfitting [12]
Decision tree	Automatically select features; Strong interpretation	Classification result trends to majority class; Ignore the correlation of data	Balanced datasets with SMOTE; Introduced latent variables
K-means	Simple, can be trained rapidly; Strong scalability; Can fit to big data	Do not perform well on nonconvex data; Sensitive to initialization; Sensitive to the parameter *K*	Improved initialization method [13]

1.2. Deep Learning Models

AUTOENCODER

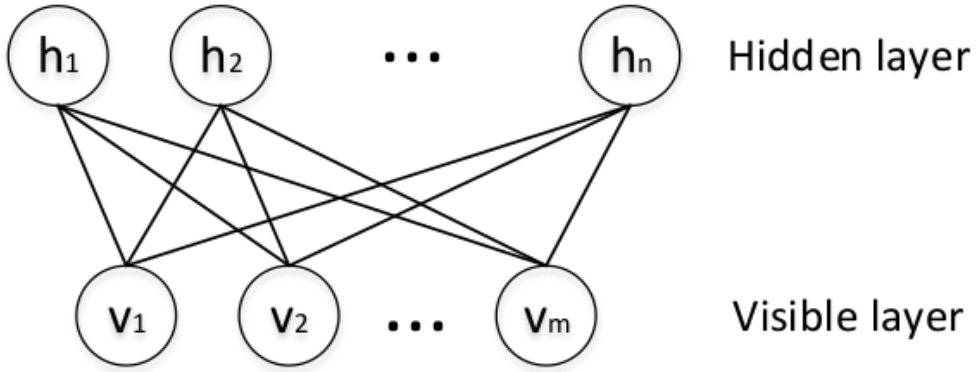
- An autoencoder contains two symmetrical components, an encoder and a decoder.



- The **encoder extracts features from raw data**, and the **decoder reconstructs the data from the extracted features.**
- During training, the divergence between the input of the encoder and the output of the decoder is gradually reduced.
- When the decoder succeeds in reconstructing the data via the extracted features, it means that the features extracted by the encoder represent the essence of the data.
- It is important to note that this entire process **requires no supervised information.**
- Famous autoencoder variants: denoising autoencoders and sparse autoencoders .

RESTRICTED BOLTZMANN MACHINE (RBM)

- An RBM is a randomized neural network in which units obey the Boltzmann distribution.
- It is composed of a visible layer and a hidden layer



Boltzmann distribution is a probability distribution that gives the probability that a system will be in a certain state as a function of that state's **energy** and the **temperature** of the system.

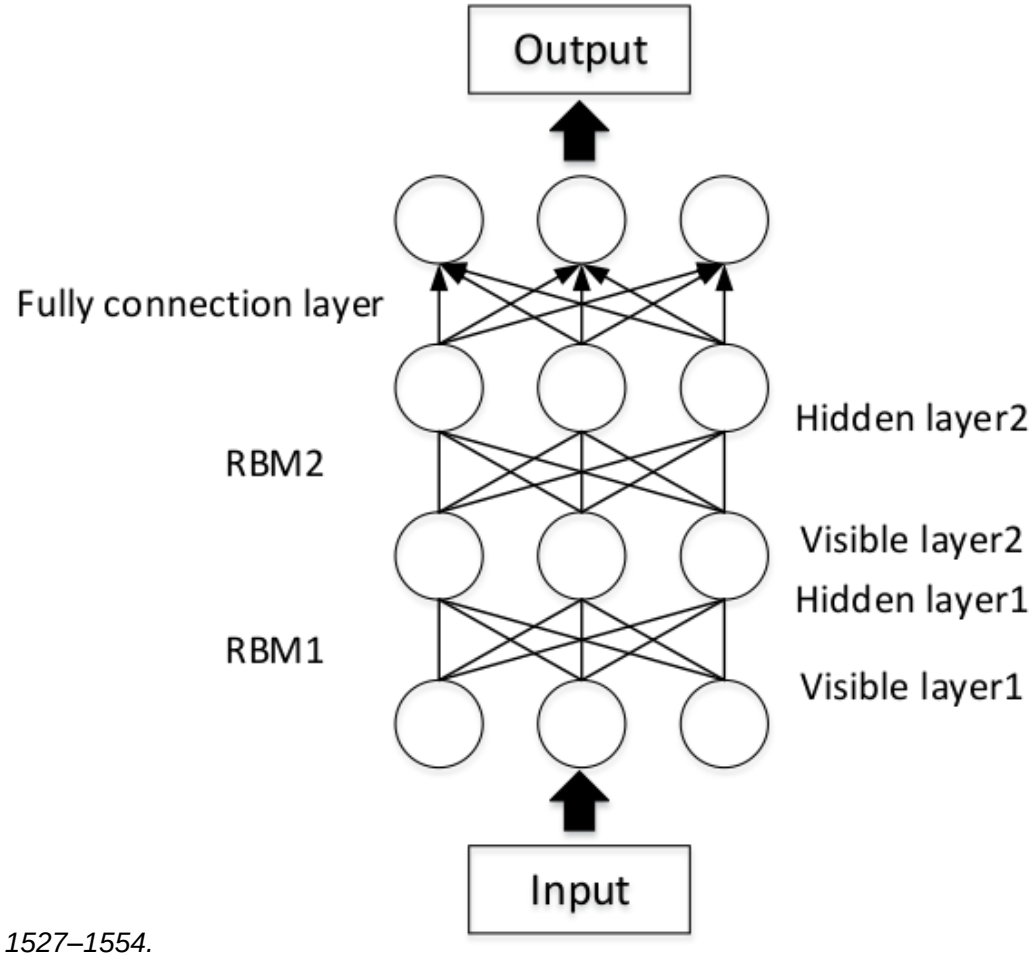
The distribution is expressed in the form:

$$p_i \propto e^{-\varepsilon_i/(kT)}$$

- The units in the same layer are not connected; however, the units in different layers are fully connected.
- RBMs do not distinguish between the forward and backward directions; thus, the weights in both directions are the same.
- RBMs are unsupervised learning models trained by the contrastive divergence algorithm [17], and they are usually applied for **feature extraction** or **denoising**.

DEEP BRIEF NETWORK (DBN)

- A **DBN consists of several RBM layers** and **a softmax classification layer.**
- Training a DBN involves two stages: **unsupervised pretraining** and **supervised fine-tuning** [18,19].
- First, each RBM is trained using greedy layer-wise pretraining.
- Then, the weight of the softmax layer are learned by labeled data.
- In attack detection, DBNs are used for both feature extraction and classification [20–22].



[18] Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. Neural Comput. 2006, 18, 1527–1554.

[19] Boureau, Y.l.; Cun, Y.L.; Ranzato, M.A. Sparse feature learning for deep belief networks. In Proceedings of the 21st Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–10 December 2008; pp. 1185–1192.

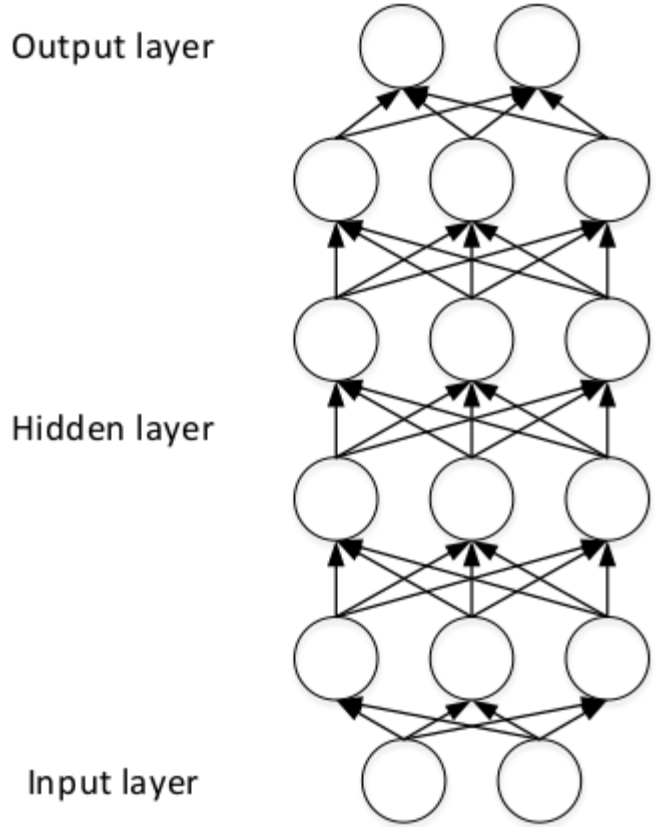
[20] Zhao, G.; Zhang, C.; Zheng, L. Intrusion detection using deep belief network and probabilistic neural network. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, July 2017; Volume 1, pp. 639–642.

[21] Alrawashdeh, K.; Purdy, C. Toward an online anomaly intrusion detection system based on deep learning.In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016; pp. 195–200.

[22] Yang, Y.; Zheng, K.; Wu, C.; Niu, X.; Yang, Y. Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. Appl. Sci. 2019, 9, 238.

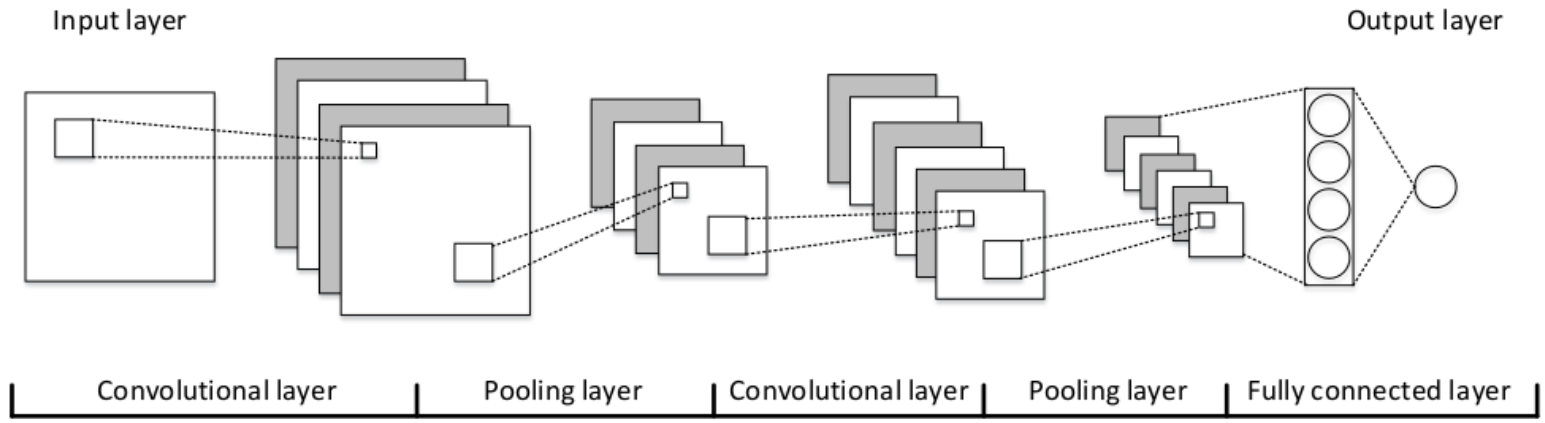
DEEP NEURAL NETWORK (DNN)

- A layer-wise pretraining and fine-tuning strategy makes it possible to construct DNNs with multiple layers.
- When training a DNN, **the parameters are learned first using unlabeled data**, which is an **unsupervised feature learning stage;**
- Then, the network is **tuned** through the labeled data, which is a **supervised learning** stage.
- The astonishing achievements of DNNs are mainly due to the unsupervised feature learning stage.



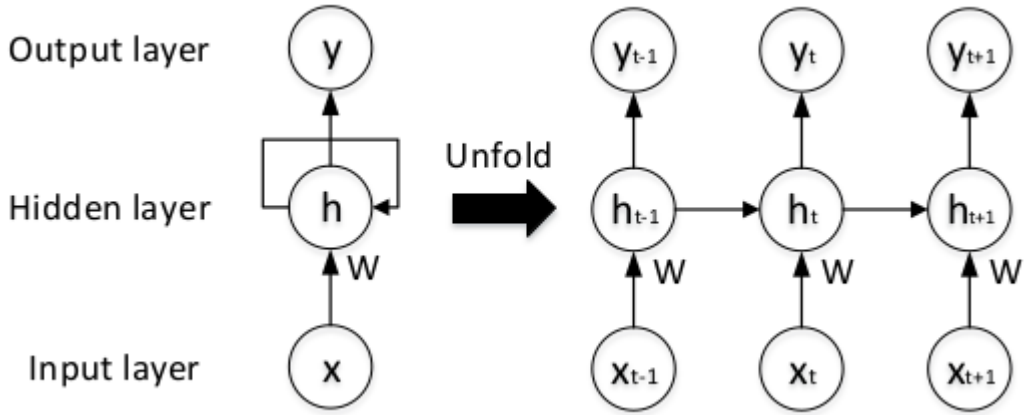
CONVOLUTIONAL NEURAL NETWORK (CNN)

- CNNs are designed to **mimic the human visual system**
- CNNs have made great achievements in the computer vision field
- A CNN is stacked with alternate convolutional and pooling layers
- The **convolutional layers are used to extract features**, and the **pooling layers are used to enhance the feature generalizability**
- CNNs work on 2-dimensional (2D) data, **so the input data must be translated into matrices for attack detection**



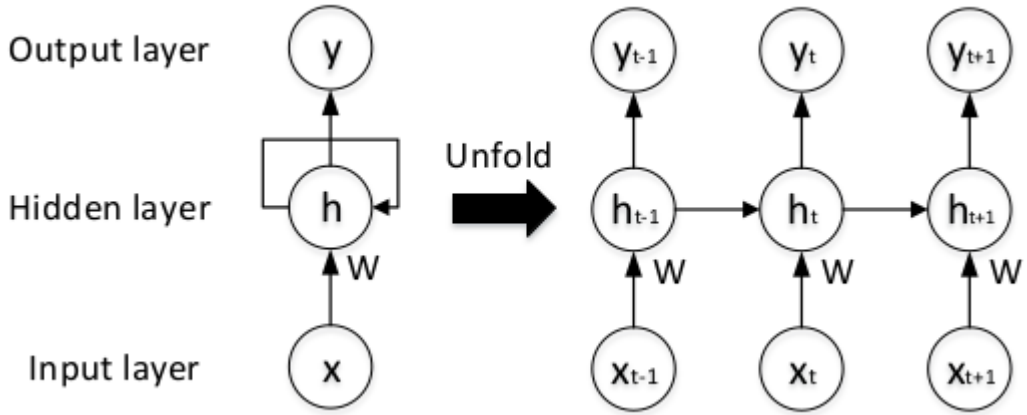
RECURRENT NEURAL NETWORK (RNN)

- RNNs are networks designed for sequential data and are **widely used in natural language processing (NLP)**
- The characteristics of sequential data are contextual; analyzing isolated data from the sequence makes no sense.
- To obtain contextual information, each unit in an RNN receives not only the current state but also ***previous states***
- All wight values (W) are the same. This characteristic causes RNNs to often suffer from vanishing or exploding gradients.
- In reality, standard RNNs deal with only limited-length sequences.
- To solve the long-term dependence problem, many RNN variants have been proposed, such as long short-term memory (LSTM) [29], gated recurrent unit (GRU) [30], and bi-RNN [31]



...RECURRENT NEURAL NETWORK (RNN)

- The **LSTM model** [29]: each LSTM unit contains three gates: a **forget gate**, an **input gate**, and an **output gate**.
- The forget gate eliminates outdated memory, the input gate receives new data, and the output gate combines short-term memory with long-term memory to generate the current memory state.
- The **GRU model** [30]: The GRU model **merges the forget gate and the input gate into a single update gate**, which is simpler than the LSTM.



[29] Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780.

[30] Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv 2014, arXiv:1412.3555.

GENERATIVE ADVERSARIAL NETWORK (GAN)

- A GAN model includes two subnetworks, i.e., a **generator** and a **discriminator**
  - The generator aims to generate synthetic data similar to the real data
  - The discriminator intends to distinguish synthetic data from real data.
- Thus, the generator and the discriminator improve each other.
- GANs are currently a hot research topic used to augment data in attack detection, which partly ease the problem of IDS dataset shortages.
- Meanwhile, GANs belong to adversarial learning approaches which can raise the detection accuracy of models by adding adversarial samples to the training set.

SUMMARY OF DL MODELS USED IN IDS

Algorithms	Suitable Data Types	Supervised or Unsupervised	Functions
Autoencoder	Raw data; Feature vectors	Unsupervised	Feature extraction; Feature reduction; Denoising
RBM	Feature vectors	Unsupervised	Feature extraction; Feature reduction; Denoising
DBN	Feature vectors	Supervised	Feature extraction; Classification
DNN	Feature vectors	Supervised	Feature extraction; Classification
CNN	Raw data; Feature vectors; Matrices	Supervised	Feature extraction; Classification
RNN	Raw data; Feature vectors; Sequence data	Supervised	Feature extraction; Classification
GAN	Raw data; Feature vectors	Unsupervised	Data augmentation; Adversarial training

1.3. Shallow Models Compared To Deep Models

SHALLOW VS DEEP MODELS

Main differences:

1) **Running time.** (includes both training and test time). Due to their high complexity, DL models are much longer than those of shallow models.

2) **Number of parameters.** There are two types of parameters:

- The ***learnable parameters*** are calculated during the training phase.
- the ***hyperparameters*** are set manually before training begins.
- Both types in DL models far outnumber those in shallow models; consequently, training and optimizing deep models takes longer.

3) **Feature representation.**

- The input to traditional ML models is a feature vector, and feature engineering is an essential step.
- In contrast, **DL models are able to learn feature representations from raw data** and are not reliant on feature engineering.
- The DL methods can execute in an end-to-end manner, giving them an outstanding advantage over traditional ML methods.

…SHALLOW VS DEEP MODELS

4) **Learning capacity.**

- The **structures of DL models are complex** and they contain huge numbers of parameters (generally millions or more).
- Therefore, the deep learning models have stronger fitting ability than do shallow models.
- **DL models also face a higher risk of overfitting**, require a much larger volume of data for training. However, the effect of deep learning models is better.

5) **Interpretability.**

- **DL models are black boxes** [32–35]; **the results are almost uninterpretable**, which is a critical point in deep learning.
- Traditional ML algorithms, such as the decision tree and naïve Bayes, have strong interpretability.

[32] Ribeiro, M.T.; Singh, S.; Guestrin, C. Why should i trust you?: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.

[33] Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In Proceedings of the Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 4765–4774.

[34] Li, J.; Monroe, W.; Jurafsky, D. Understanding neural networks through representation erasure. ArXiv 2016, arXiv:1612.08220.

[35] Fong, R.C.; Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3429–3437.

2. EVALUATION METRICS

METRICS FOR EVALUATING ML/DL METHODS

Multiple metrics are often used simultaneously in IDS research.

1) **Accuracy** is defined as the ratio of correctly classified samples to total samples.

- Accuracy is a *suitable metric when the dataset is balanced*.
- In *real network environments however, normal samples are much more than are abnormal samples;* thus, accuracy may not be a suitable metric.

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

2) **Precision (P)** is defined as the ratio of true positive samples to predicted positive samples;

- It represents the *confidence of attack detection*.

$$P = \frac{TP}{TP + FP}$$

TP=True Positives. TN=True Negatives. FP=False Positives. FN=False Negatives

…METRICS FOR EVALUATING ML/DL METHODS

Multiple metrics are often used simultaneously in IDS research.

3) Recall (R) is defined as the ratio of true positive samples to total positive samples and *is also called the detection rate*.

- The detection rate reflects the model's ability to recognize attacks, which is an important metric in IDS.

$$R = \frac{TP}{TP + FN}$$

(4) F-measure (F) is defined as the harmonic average of the precision and the recall.

$$F = \frac{2 * P * R}{P + R}$$

…METRICS FOR EVALUATING ML/DL METHODS

Multiple metrics are often used simultaneously in IDS research.

5) The false negative rate (FNR) is defined as the ratio of false negative samples to total positive samples.

- In attack detection, the FNR is also called the *missed alarm rate*.

$$FNR = \frac{FN}{TP + FN}$$

6)The false positive rate (FPR) is defined as the ratio of false positive samples to predicted positive samples.

- In attack detection, the FPR is also called the *false alarm rate.*

$$FPR = \frac{FP}{TP + FP}$$

For IDS , the most frequently used metrics are accuracy, recall (or detection rate), FNR (or missed alarm rate), and FPR (or false alarm rate).

3. BENCHMARK DATASETS IN IDS

BENCHMARK DATASETS FOR IDS

- Understanding data is the basis of machine learning methodology. As always, the performance of ML/DL ***depends upon the quality of the input data***.
- For IDSs, the adopted data should be
  - easy to acquire and
  - reflect the behaviors of the hosts or networks.
- The common source data types for IDSs are:
  - packets,
  - flow,
  - sessions, and
  - logs.
- Building a dataset is complex and time-consuming.
- After a benchmark dataset is constructed, it can be reused repeatedly by many researchers.

KNOWN IDS DATASETS – DARPA1998

(1) **DARPA1998.** http://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset

- Built by the Lincoln laboratory of MIT, it is a widely used benchmark dataset in IDS.
- To compile it, the researchers collected Internet traffic over nine weeks;
  - the first seven weeks form the training set,
  - and the last two weeks form the test set.
- The dataset contains both **raw packets** and **labels**.
- There are five types of labels:
  - Normal,
  - Denial of Service (DOS),
  - Probe,
  - User to Root (**U2R**) and
  - Remote to Local (**R2L**).
- Because raw packets cannot be directly applied to traditional machine learning models, the KDD99 dataset was constructed to overcome this drawback.

KNOWN IDS DATASETS – KDD99

(2) **KDD99.** http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

- The most widespread IDS benchmark dataset at present.
- Its compilers extracted **41-dimensional features** from data in DARPA1998.
- The labels in KDD99 are the same as the DARPA1998.
- There are four types of features in KDD99, i.e.,
  - Basic features,
  - Content features,
  - Host-based statistical features, and
  - Time-based statistical features.
- Unfortunately, the KDD99 dataset includes **many defects.**
  - The data are **severely unbalanced**, making the classification results **biased toward the majority classes**.
  - There are many **duplicate records** and redundant records exist.
  - Many researchers have to filter the dataset carefully before they can use it. As a result, the **experimental results** from different studies are **not always comparable**.
  - KDD data are **too old to represent the current network environment.**

KNOWN IDS DATASETS – NSL-KDD

(3) **NSL-KDD.** https://www.unb.ca/cic/datasets/nsl.html

- The records in the NSL-KDD were carefully selected based on the KDD99.
- Records of **different classes are balanced** in the NSL-KDD, which avoids the classification bias problem.
- The NSL-KDD also **removed duplicate and redundant records**; therefore, it contains only a moderate number of records.
- Therefore, the experiments can be implemented on the whole dataset, and the **results from different papers are consistent and comparable.**
- The NSL-KDD **alleviates the problems of data bias and data redundancy to some degree**.
- However, the NSL-KDD **does not include new data**; thus, **minority class samples are still lacking**, and its samples are **still out-of-date.**

KNOWN IDS DATASETS – UNSW-NB15

(4) **UNSW-NB15** [39] Compiled by the University of South Wales. https://research.unsw.edu.au/projects/unsw-nb15-dataset

- Researchers configured three virtual servers to capture network traffic and extracted 49-dimensional features using the *Bro* IDS tool (now known as *Zeek*).
- The dataset includes more types of attacks than does the KDD99 dataset, and its features are more plentiful.
- The data categories include normal data and nine types of attacks.
- The features include:
  - flow features,
  - basic features,
  - content features,
  - time features,
  - additional features, and
  - labeled features.
- The UNSW-NB15 is representative of new IDS datasets, and has been used in some recent studies.
- Although the influence of UNSW-NB15 is currently inferior to that of KDD99, it is necessary to construct new datasets for developing new IDS based on machine learning.

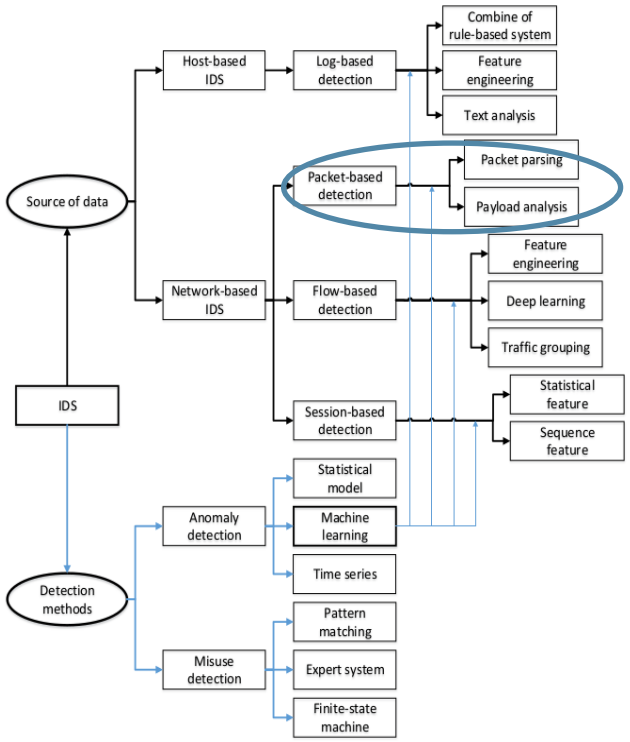
[39] Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications And Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6

4. APPLYING ML/DL IN IDS – STATE OF THE ART RESEARCH

APPLYING ML/DL IN IDS DESIGN

- ML/DL are data-driven methods
- The **different types of data reflect different attack behaviors**, which include **host behaviors** and **network behaviors**:
  - *Host behaviors* are reflected by **system logs**
  - *Network behaviors* are reflected by **network traffic**.
- **Each attack type has a unique pattern**.
- *Selecting appropriate data sources is required to detect different attacks according to the attack characteristics.*
  Examples:
  - A key feature of a **DOS attack** is to send many packets within a very short period of time; therefore **flow data** is suitable for detecting a DOS attack.
  - A **covert channel** involves data-leaking activity between two **specific IP addresses**, which is more suited to detection from session data.

4.1. Packet-Based Attack Detection



APPLYING ML/DL IN IDS DESIGN: PACKET-BASED ATTACK DETECTION

- Packets, are the basic units of network communication.
- Packets are binary data, meaning that they are incomprehensible unless they are first parsed.
- A packet consists of a header and application data.
  - The **headers** are structured fields that specify IP addresses, ports and other fields specific to various protocols.
  - The **application data** portion contains the payload from the application layer protocols.
- There are three advantages to using packets as IDS data sources:
  - (1) Packets contain **communication contents**; thus, they can effectively be used to detect U2R and R2L attacks.
  - (2) Packets contain **IPs and timestamps**; thus, they can locate the attack sources.
  - (3) Packets can be **processed instantly without caching**; suitable for real-time detection.
- Individual packets **do not reflect the full communication state,** nor **the contextual information** of each packet, so it is difficult to detect some attacks, such as DDOS.
- Detection methods based on packets mainly include: **packet parsing** and **payload analysis** .

PACKET-BASED ATTACK DETECTION: (1) PACKET PARSING-BASED DETECTION

- Each network protocol (e.g. HTTP, DNS) has its own headers. Packet parsing-based detection methods primarily focus on the **protocol header fields.**
- Usual practice:
  - (a) **Extract the header fields** using parsing tools (e.g. Wireshark)
  - (b) ***Treat the values of the most important fields as feature vectors*.**
- Packet parsing-based detection methods apply to **shallow models**.
- The header fields provide basic packet information from which features can be extracted, using **classification algorithms** to detect attacks.
- In packet parsing-based detection, **unsupervised learning** is a common way to solve the high false alarm rate problem.

PACKET-BASED ATTACK DETECTION: … (1) PACKET PARSING-BASED DETECTION

- **Example of classification algorithms:** Mayhew et al. [40] proposed an **SVM**- and **K-means**-based packet detection method.
  - They captured packets from a real enterprise network and parsed them with Bro.
  - First, they grouped the packets according to protocol type. Then, they **clustered the data with the K-means++ algorithm** for the different protocol datasets.
  - Thus, the original dataset was grouped into many clusters, where the data from any given cluster were homologous.
  - Next, they extracted features from the packets and **trained SVM models on each cluster.**
  - Their precision scores for HTTP, TCP, Wiki, Twitter, and E-mail protocols reached 99.6%, 92.9%, 99%, 96%, and 93%, respectively.
- **Example of unsupervised learning:** Hu et al. [41] proposed a **fuzzy C-means** based packet detection method.
  - The fuzzy C mean algorithm **introduces fuzzy logic into the standard K-means algorithm** such that samples belong to a cluster with a membership degree rather than as a Boolean value such as 0 or 1.
  - They used Snort to process the DARPA 2000 dataset, extracting Snort alerts, source IPs, destination IPs, source ports, destination ports, and timestamps.
  - They used this information to **form feature vectors and distinguished false alerts from true alerts by clustering the packets.**
  - To reduce the influence of initialization, they ran the clustering algorithms ten times.
  - Results: the fuzzy C-means algorithm reduced the false alarm rate by 16.58% and the missed alarm rate by 19.23%.

PACKET-BASED ATTACK DETECTION: (2)

PAYLOAD ANALYSIS-BASED DETECTION

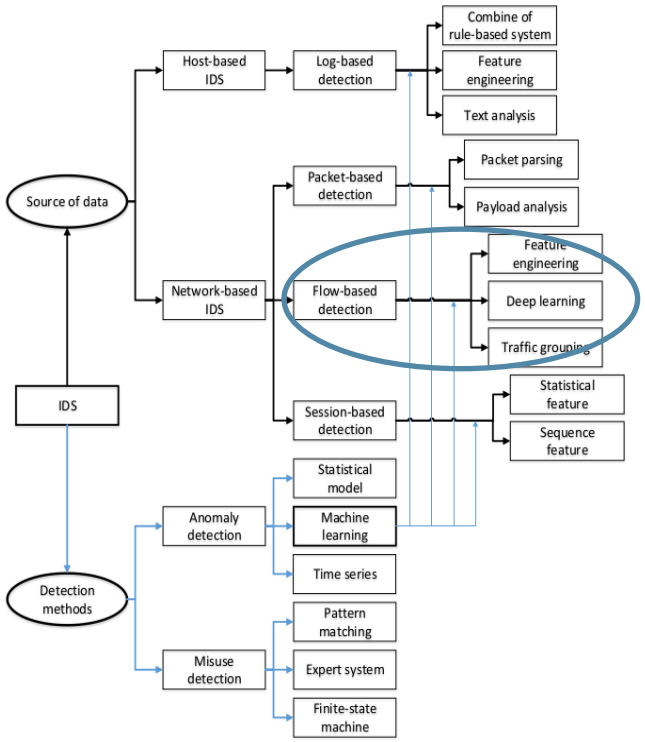
- Payload analysis-based detection places **emphasis on the application data.**
- **Suitable for multiple protocols** because they do not need to parse the packet headers.
- As a type of unstructured data, **payloads can be processed directly by deep learning models.**
- Advantages:
  - Shallow models depend on manual features and private information in packets, leading to high labor costs and privacy leakage problems. Deep learning methods **learn features from raw data without manual intervention.**
- Disadvantages:
  - Cannot be applied in encrypted payloads.

PACKET-BASED ATTACK DETECTION: ...(2) PAYLOAD ANALYSIS-BASED DETECTION

- Min et al. [43] utilized **a text-based CNN** to detect attacks from payloads.
  - They conducted experiments on the ISCX 2012 dataset and detected attacks with both statistical and content features.
    - The statistical features came from **packet headers** and included protocols, IPs, and ports.
    - The content features came from the **payloads**.
  - First, payloads from different packets were concatenated. The concatenated payloads were encoded.
  - Then, the content features were extracted with a CNN.
  - Finally, they trained a random forest model to detect attacks.
  - The final model reached an accuracy of 99.13%.
- Zeng et al. [44] proposed a payload detection method with **multiple deep learning models**.
  - They adopted three deep learning models (a **CNN**, an **RNN**, and a **stacked autoencoder**) to extract features from different points of view.
  - Among these, the CNN extracted local features, the RNN extracted time series features, and the stacked autoencoder extracted text features.
  - The accuracy of this combined approach reached 99.22% on the ISCX 2012 dataset.

PACKET-BASED ATTACK DETECTION: ...(2) PAYLOAD ANALYSIS-BASED DETECTION

- **Unsupervised learning example.** Yu et al. [45] utilized a **convolutional autoencoder** to extract payload features.
  - They conducted experiments on the CTU-UNB dataset, which includes the raw packets of 8 attack types.
  - To take full advantage of convolutions, they first converted the packets into images.
  - Then, they trained a convolutional autoencoder model to extract features.
  - Finally, they classified packets using the learned features.
  - The precision, recall and F-measure on the test set reached 98.44%, 98.40%, and 98.41% respectively.
- **Adversarial learning example.** Rigaki et al. [46] used a **GAN** to improve the malware detection effect.
  - To evade detection, malware applications try to generate packets similar to normal packets.
  - Example: the malware FLU as an example generates command & control (C & C) packets that are very similar to packets generated by Facebook.
  - They configured a virtual network with various hosts, servers, and an IPS.
  - They started up the malware FLU and trained a GAN model. The GAN guided the malware to produce packets similar to Facebook.
  - As the training epochs increased, the packets blocked by the IPS decreased and packet that passed inspection increased.
  - The result was that the malicious packets generated by the GAN were more similar to normal packets.
  - Then, by analyzing the generated packets, the robustness of the IPS was improved.



4.2. Flow-Based Attack Detection

APPLYING ML/DL IN IDS DESIGN: FLOW-BASED ATTACK DETECTION

- Flow data contains packets grouped in a period (KDD99 and the NSL-KDD are flow datasets).
- Flow-based attack detection mainly includes **feature engineering** and **deep learning methods**.
- Benefits of flow-based detection:
  - Flow represents the whole network environment, which can detect most attacks, especially DoS and Probe.
  - Without packet parsing or session restructuring, flow preprocessing is simple.
- Disadvantages:
  - Flow ignores the content of packets; thus, the detection of U2R and R2L is unsatisfactory.
  - When extracting flow features, packets must be cached packets;
  - In addition, the strong heterogeneity of flow may cause poor detection effects

FLOW-BASED ATTACK DETECTION: (1)

FEATURE ENINEERING-BASED DETECTION

- Traditional ML models cannot directly address flow data; feature engineering is an essential step before these models can be applied.
- They adopt a **"feature vectors + shallow models"** mode. The feature vectors are suitable for most ML algorithms.
- Each dimension of the feature vectors has **clear interpretable semantics.**
- The common features include:
  - the average packet length,
  - the variance in packet length,
  - the ratio of TCP to UDP,
  - the proportion of TCP flags,
  - etc.
- Advantages:
  - simple to implement,
  - highly efficient,
  - can meet real-time requirements.
- The existing feature engineering-based IDSs often have high detection accuracy but suffer from a high false alarm rate. One solution is to combine many weak classifiers to obtain a strong classifier.

FLOW-BASED ATTACK DETECTION: ...(1) FEATURE ENINEERING-BASED DETECTION

- **Combining weak classifiers:** Goeschel et al. [47] proposed a hybrid method that combines SVM, decision tree, and Naïve Bayes algorithms.
  - First train an **SVM model** to divide the data into **normal** or a**bnormal samples**.
  - For the abnormal samples, utilize a **decision tree** model to **determine specific attack types.**
  - Since decision trees can only identify known attacks, finally apply a **Naïve Bayes** classifier to **discover unknown attacks**.
  - This hybrid method achieved an accuracy of 99.62% and a false alarm rate of 1.57% on the KDD99 dataset.
- **Accelerating detection speed:** Kuttranont et al. [48] proposed a KNN-based detection method and accelerated calculation via parallel computing running on a graphics processing unit (GPU).
  - A modified neighbor-selecting rule of the KNN algorithm.
  - The standard KNN selects the top K nearest samples as neighbors, while the improved algorithm **selects a fixed percentage (such as 50%) of the neighboring samples as neighbors.**
  - The method considers the unevenness of data distribution and performs well on sparse data.
  - Experiments were conducted using the KDD99 dataset, achieving an accuracy of 99.30%.
  - GPU acceleration results to 30 times faster calculation than that without the GPU.

FLOW-BASED ATTACK DETECTION: (2)

DEEP LEARNING-BASED DETECTION

- Feature engineering depends on domain knowledge; the quality of features often becomes a bottleneck of detection.
- DL methods can directly **process raw data**, allowing them to learn features.
- DL-based detection methods **learn features automatically** and **perform classification at the same time**.
- They are gradually becoming the mainstream in IDS.
- **CNN-based detection example**: Potluri et al. [49].
  - They conducted experiments on the NSL-KDD and the UNSW-NB 15 datasets. The data type in these datasets is a feature vector.
  - Because CNNs are good at processing 2-dimensional (2D) data, **they first converted the feature vectors into images.** Nominal features were one-hot coded, and the feature dimensions increased from 41 to 464. Then, each 8-byte chunk was transformed into one pixel. Blank pixels were padded with 0.
  - The end result was that the **feature vectors were transformed into images of 8*8 pixels.**
  - Finally, they constructed a 3-layer CNN to classify the attacks.
  - They compared their model with other deep networks (ResNet 50 and GoogLeNet), and the proposed CNN performed best, reaching accuracies of 91.14% on the NSL-KDD and 94.9% on the UNSW-NB 15.

FLOW-BASED ATTACK DETECTION: ...(2) DEEP LEARNING-BASED DETECTION

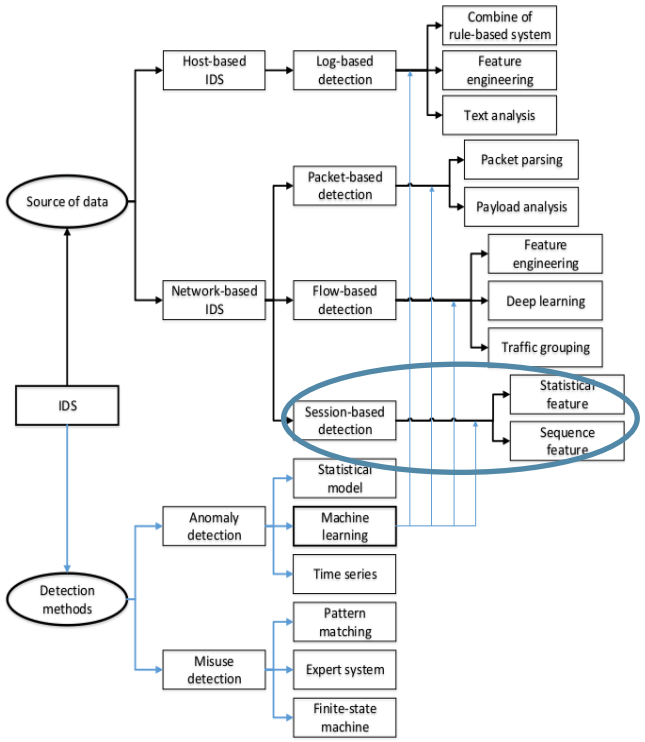
- **Unsupervised DL for feature extraction + shallow model for classification.** Zhang et al. [50].
  - Extract features with a sparse autoencoder and detected attacks with an XGBoost model.
  - They used data from the NSL-KDD dataset.
  - To solve the imbalanced nature of the dataset, they first sampled the dataset using SMOTE.
  - The SMOTE algorithm **oversamples the minority classes** and **divides the majority classes into many subclasses** so that every class is balanced.
  - The sparse autoencoder introduces a sparsity constraint into the original autoencoder, **enhancing its ability to detect unknown samples.**
  - Their model achieved accuracies on the Normal, DOS, Probe, R2L, and U2R classes of 99.96%, 99.17%, 99.50%, 97.13%, and 89.00%, respectively.
- **Adversarial learning (GAN) approach:** DL models do not perform well on small or unbalanced datasets. Adversarial learning approaches can improve the detection accuracy on small datasets. Zhang et al. [51].
  - The **KDD99 dataset is both unbalanced and lacks new data**, which **leads to poor generalizability** of ML models.
  - To address these problems, they utilized a **GAN to expand the dataset** (data augmentation)..
  - The **GAN model generated data similar to the flow data** of KDD99. Adding this generated data to the training set allows attack variants to be detected.
  - They selected 8 types of attacks and compared the accuracies achieved on the original dataset compared to the expanded dataset.
  - The experimental results showed that adversarial learning improved 7 accuracies in 8 attack types.

FLOW-BASED ATTACK DETECTION: (3)

TRAFFIC GROUPING-BASED DETECTION

- Flow includes all traffic within a period, and **many types of traffic may act as white noise** in attack detection.
- Solution: **group traffic to decrease heterogeneity**. The grouping methods include protocol-based and data-based methods.
- The traffic features of various protocols have significant differences; thus, grouping traffic by protocol is a valid step toward improving accuracy.
- Protocol-based grouping method. Teng et al. [52]:
  - They proposed an **SVM detection** method based on protocol grouping using the data of the KDD99 dataset, which involves various protocols.
  - **They divided the dataset based on protocol type**, by considering **the TCP, UDP, and ICMP** protocols.
  - According to the characteristics of these protocols, they **selected features for each sub-dataset**.
  - Finally, they trained the SVM models on the 3 subdatasets, obtaining an average accuracy of 89.02%.
- Data-based grouping method. Ma et al. [53]:
  - A clustering method based on a **DNN** and **spectral clustering-based** detection method.
  - They first divided the original dataset into **6 subsets**, in which each subset was **highly homogeneous.**
  - Then, they trained DNN models on every subset.
  - The accuracy of their approach on the KDD99 and the NSL-KDD datasets reached 92.1%.

4.3. Session-Based Attack Detection



APPLYING ML/DL IN IDS DESIGN: SESSION-BASED ATTACK DETECTION

- A session is the interaction process between two terminal applications and can represent high-level semantics.
- A session is usually divided on the basis of a 5-tuple **(client IP, client port, server IP, server port, and protocol)**.
- There are two advantages of detection using sessions.
  - (1) Sessions are suitable for **detecting an attack between specific IP addresses**, such as tunnel and Trojan attacks.
  - (2) Sessions contain detailed communications between the attacker and the victim, which can help **localize attack sources.**
- However, session duration can vary dramatically. As a result, a session analysis sometimes needs to cache many packets, which may increase lag.
- The session-based detection methods primarily include **statistics-based features** and **sequence-based features.**

SESSION-BASED ATTACK DETECTION: (1)

STATISTIC-BASED FEATURE DETECTION

- Session statistical information includes
  - the fields in packet headers,
  - the number of packets,
  - the proportion of packets coming from different directions,
  - etc.
- Main use: **compose feature vectors suitable for shallow models.**
- The sessions have **high layer semantics**; thus, they are **easily described by rules**.
- Decision tree or rule-based models may be appropriate methods.
- Unfortunately, the methods based on statistical features **ignore the sequence information**, and they have **difficulties detecting intrusions related to communication content.**
- Because statistical information includes the basic features of sessions, supervised learning methods can utilize such information to differentiate between normal sessions and abnormal sessions.
- The existing session-based detection methods often face problems of **low accuracy** and have **high runtime costs.**

SESSION-BASED ATTACK DETECTION: ...(1) STATISTIC-BASED FEATURE DETECTION

- **Hierarchical decision tree.** Ahmim et al. [54]:
  - To reduce the detection time, they analyzed the frequency of different types of attacks and designed the detection system to recognize specific attacks.
  - They used data from the CICIDS 2017 dataset that included 79-dimensional features and 15 classes.
  - The proposed detection system had a two-layer structure.
  - The first layer consisted of two independent classifiers (i.e., a decision tree and a rule-based model), which processed part of the features.
  - The second layer was a random forest classifier, which processed all the features from the dataset as well as the output of the first layer.
  - They compared multiple machine learning models on 15 classes; their proposed methods performed best on 8 of the 15 classes. Moreover, the proposed method had low time consumption, reflecting its practicability.
- **An unsupervised method to detect attacks in smart grids.** Alseiari et al. [55]:
  - A session-based detection using supervised learning models **depends on expert knowledge, which is difficult to expand** to new scenarios.
  - Solution: Due to the lack of smart grid datasets, they **constructed a dataset through simulation** experiments.
  - First, they captured and cached packets to **construct sessions.**
  - Then, they **extracted 23-dimensional features** from the sessions.
  - Next, they utilized **mini batch K-means** to divide the data into many clusters.
  - Finally, they **labeled the clusters.**
  - No expert knowledge was required for any part of this process.
  - Two assumptions: (1) normal samples were the majority. (2) the distances among the normal clusters were relatively short.
  - When the size of a cluster was less than 25% of the full sample amount or a cluster centroid was far away from all other the other cluster centroids, that cluster was judged as abnormal.
  - The proposed methods were able to detect intrusion behaviors in smart grids effectively and locate the attack sources while holding the false alarm rate less to than 5%.

SESSION-BASED ATTACK DETECTION: (2)

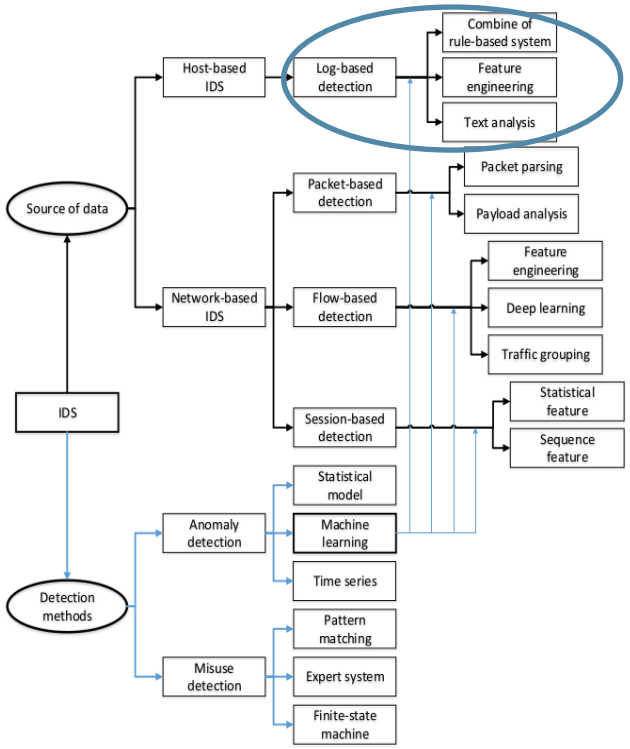
SEQUENCE FEATURE-BASED DETECTION

- Different from flow, the packets in sessions have a strict order relationship. The sequence features mainly contain the packet length sequence and the time interval sequence. Analyzing the sequence can obtain detailed session interaction information.
- Most machine learning algorithms cannot deal with sequences, and related methods are relatively rare. At present, most sequence feature-based detection adopts the RNN algorithm.
- Encoding raw data is a common preprocessing step for RNN methods. The bag of words (BoW) model is a frequently used text processing technology.
- Yuan et al. [56] proposed a DDOS detection method based on the LSTM using UNB ISCX 2012 dataset.
  - They first extracted 20-dimensional features from the packets and encoded them with BoW.
  - Then, they concatenated the packets in sequence, resulting in matrices with a size of m*n, where m was the number of packets in a session and n was the dimension of a packet, and both m and n were variable.
  - Finally, they trained a CNN to extract local features and an LSTM to classify the sessions.
  - They provided comprehensive experimental results, reaching accuracy, precision, recall, and F-measure scores of 97.606%, 97.832%, 97.378%, and 97.601%, respectively.

SESSION-BASED ATTACK DETECTION: ...(2) SEQUENCE FEATURE-BASED DETECTION

- One of the drawbacks of the BoW is that it is unable to represent the similarity between words. Word embedding approaches overcome that problem.
- Radford et al. [57] proposed a session detection method based on a bi-LSTM.
  - Because LSTMs had made great strides in NLP, they expressed the sessions as a specific language. They conducted experiments on the ISCX IDS dataset.
  - First, they grouped packets on the basis of IP addresses to obtain sessions.
  - Then, they encoded the sessions with the word embedding.
  - Finally, they trained an LSTM model to predict abnormal sessions.
  - To utilize the contextual information, they adopted a bi-LSTM model to learn the sequence features in two directions. In addition to text processing technology, the character-level CNN is a novel encoding method.
- Wang et al. [58] proposed a hierarchical deep learning detection method in which a session contains not only packet contents but also the packet time sequence.
  - Then, they designed a hierarchical deep learning method using a CNN to learn the low-level spatial features and an LSTM to learn the high-level time features, where the time features are based on the spatial features.
  - They conducted experiments on the DARPA 1998 and the ISCX 2012 datasets.
  - They first applied the CNN to extract spatial features from packets. Next, they concatenated the spatial features in sequence and extracted time features using the LSTM model. The resulting model achieved accuracies between 99.92% and 99.96%, and detection rates between 95.76% and 98.99%.

4.4. Log-Based Attack Detection



APPLYING ML/DL IN IDS DESIGN: LOG-BASED ATTACK DETECTION

- Logs are the activity records of operating systems or application programs; they include system calls, alert logs, and access records.
- Logs have **definite semantics.**
- Benefits to using logs as a data source in IDS.
  - (1) Logs include detailed content information suitable for detecting SQL injection, U2R, and R2L attacks.
  - (2) Logs often carry information about users and timestamps that can be used to trace attackers and reveal attack times.
  - (3) Logs record the complete intrusion process; thus, the result is interpretable.
- Disadvantages:
  - (1) Log analysis depends on cyber security knowledge.
  - (2) The log formats of different application programs do not have identical formats, resulting in low scalability.
- The log-based attack detection primarily includes **hybrid methods** involving
  - rules and machine learning,
  - log feature extraction-based methods,
  - and text analysis-based methods.

LOG-BASED ATTACK DETECTION: (1)

RULE AND ML-BASED HYBRID METHODS

- Hybrid methods combine rule-based detection and machine learning, which together achieve better performances than do single detection systems.
- Many rule-based detection systems (e.g., Snort) **generate masses of alerts;** however, most of the alerts involve only operations that do not match the rules; therefore, these are often not real intrusion behaviors.
- The hybrid methods take the log output of the rule-based systems as inputs; then, **ML models are used to filter out the meaningless alerts.**
- Many IDSs suffer from high false alarm rates, which cause real attacks to be embedded among many meaningless alerts. Ranking alerts via machine learning models forms a possible solution.

LOG-BASED ATTACK DETECTION: ...(1)

RULE AND ML-BASED HYBRID METHODS

- To reduce the false alarm rate, Meng et al. [59] proposed a KNN method to filter alarms.
  - They conducted experiments in a real network environment and generated alerts using Snort.
  - Then, they trained a KNN model to rank the alerts.
  - There were 5 threat levels in total in their experiment, and the results showed that the KNN model reduced the number of alerts by 89%.
- Some IDSs perform a function similar to human interaction, in which alerts are ranked by machine learning to reduce analyst workloads.
- McElwee et al. [60] proposed an alert filtering method based on a DNN.
  - They first collected the log generated by McAfee.
  - Then, they trained a DNN model to find important security events in the logs.
  - Next, the extracted important events were analyzed by security experts.
  - Then, the analysis results were used as training data to enhance the DNN model, forming an interaction and promotion cycle.
  - The proposed hybrid system can reduce analyst workloads and accelerate security analyses.

LOG-BASED ATTACK DETECTION: (2) LOG FEATURE EXTRACTION-BASED METHODS

- This method involves extracting log features according to **domain knowledge** and discovering abnormal behaviors using the extracted features, which is suitable for most machine learning algorithms.
- Using a sliding window to extract features is a common approach. The sliding window makes use of the contextual information contained in logs. In addition, the sliding window is a streaming method that has the benefit of low delay.
- Intrusion behaviors may leave traces of system calls, and analyzing these system calls with classification algorithms can detect intrusions.
- Tran et al. [61] proposed a CNN method to analyze system calls.
  - Every underlying operation that involves the operating system will use system calls; thus, analyzing the system call path can reproduce the complete intrusion process.
  - They conducted experiments on the NGIDS-DS and the ADFA-LD datasets, which include a series of system calls.
  - First, they extracted features with a sliding window. Then, they applied a CNN model to perform classification. The CNN was good at finding local relationships and detecting abnormal behaviors from system calls.

LOG-BASED ATTACK DETECTION: (2) LOG FEATURE EXTRACTION-BASED METHODS

- Model interpretation is another important research direction, which has attracted extensive attention.
- Tuor et al. [62] proposed an interpretable deep learning detection method using data from the CERT Insider Threat dataset, which consists of system logs.
  - They first extracted 414-dimensional features using a sliding window.
  - Then, they adopted a DNN and an RNN to classify logs.
  - The DNN detected attacks based on the log contents, and the RNN detected attacks based on the log sequences.
  - The proposed methods reduced the analysis workload by 93.5% and reached a detection rate of 90%.
  - Furthermore, they decomposed the abnormal scores into the contributions of each behavior, which was a helpful analysis.
- Interpretable models are more convincing than are uninterpretable models.
- Some logs lack labeled information; consequently, supervised learning is inappropriate.
- Unsupervised learning methods are usually used with unlabeled logs.
- Bohara et al. [63] proposed an unsupervised learning detection method in the enterprise environment.
  - They conducted experiments on the VAST 2011 Mini Challenge 2 dataset and extracted features from the host and network logs.
  - Due to the different influences of each feature, they selected features using the Pearson correlation coefficient.
  - Then, they clustered the logs with the K-means and DBSCAN algorithms.
  - By measuring the salient cluster features, the clusters were associated with abnormal behaviors.
  - Finally, they analyzed the abnormal clusters manually to determine the specific attack types.

LOG-BASED ATTACK DETECTION: (3) TEXT ANALYSIS-BASED DETECTION

- The text analysis-based detection regards logs as plain text. The methods utilize mature text processing techniques such as the n-gram to analyze logs.
- Compared with log feature extraction-based methods, this method understands log content at the semantic level and therefore has stronger interpretability.
- In log-based detection, extracting text features from logs and then performing classification is the usual approach.
- When analyzing texts, a small number of keywords have large impacts on the whole text. Thus, the keywords in the field of cyber security aid in improving the detection effect.
- Uwagbole et al. [64] proposed an SQL-injection detection method for the Internet of Things (IoT).
  - They collected and labeled logs from a real environment. The logs provide the contextual information of the SQL injection attack.
  - First, they extracted 479,000 high-frequency words from the logs and then added 862 keywords that appear in SQL queries to compose a dictionary.
  - Then, they removed duplicate and missing records from the log and balanced the data with SMOTE.
  - Next, they extracted features using the n-gram algorithm and selected features using Chi-square tests.
  - Finally, they trained an SVM model to perform classification, achieving accuracy, precision, recall, and F-measure scores of 98.6%, 97.4%, 99.7% and 98.5%, respectively.

LOG-BASED ATTACK DETECTION: ...(3) TEXT ANALYSIS-BASED DETECTION

- In an actual network environment, normal samples are in the majority, and abnormal samples are rare.
- One-class classification, a type of unsupervised learning method, uses only normal samples for training, which solves the problem of a lack of abnormal samples.
- Vartouni et al. [65] proposed a web attack detection method based on the isolate forest model.
  - They used the data of the CSIC 2010 dataset.
  - First, they extracted 2572-dimensional features from HTTP logs with the n-gram.
  - Then, they utilized an autoencoder to remove irrelevant features.
  - Finally, they trained an isolation forest model to discover abnormal webs, which reached an accuracy of 88.32%.

SUMMARY AND COMPARISON

Methods	Papers	Data Sources	Machine Learning Algorithms	Datasets
Packet parsing	Mayhew et al. [40]	Packet	SVM and K-means	Private dataset
	Hu et al. [41]	Packet	Fuzzy C-means	DARPA 2000
Payload analysis	Min et al. [43]	Packet	CNN	ISCX 2012
	Zeng et al. [44]	Packet	CNN, LSTM, and autoencoder	ISCX 2012
	Yu et al. [45]	Packet	Autoencoder	CTU-UNB
	Rigak et al. [46]	Packet	GAN	Private dataset
Statistic feature for flow	Goeschel et al. [47]	Flow	SVM, decision tree, and Naïve Bayes	KDD99
	Kuttranont et al. [48]	Flow	KNN	KDD99
	Peng et al. [13]	Flow	K-means	KDD99
Deep learning for flow	Potluri et al. [49]	Flow	CNN	NSL-KDD and UNSW-NB15
	Zhang et al. [50]	Flow	Autoencoder and XGBoost	NSL-KDD
	Zhang et al. [51]	Flow	GAN	KDD99
Traffic grouping	Teng et al. [52]	Flow	SVM	KDD99
	Ma et al. [53]	Flow	DNN	KDD99 and NSL-KDD
Statistic feature for session	Ahmim et al. [54]	Session	Decision tree	CICIDS 2017
	Alseiari et al. [55]	Session	K-means	Private dataset
Sequence feature for session	Yuan et al. [56]	Session	CNN and LSTM	ISCX 2012
	Radford et al. [57]	Session	LSTM	ISCX IDS
	Wang et al. [58]	Session	CNN	DARPA 1998 and ISCX 2012
Rule-based	Meng et al. [59]	Log	KNN	Private dataset
	McElwee et al. [60]	Log	DNN	Private dataset
Log feature extraction with sliding window	Tran et al. [61]	Log	CNN	NGIDS-DS and ADFA-LD
	Tuor et al. [62]	Log	DNN and RNN	CERT Insider Threat
	Bohara et al. [63]	Log	K-means and DBSCAN	VAST 2011 Mini Challenge 2
Text analysis	Uwagbole et al. [64]	Log	SVM	Private dataset
	Vartouni et al. [65]	Log	Isolate forest	CSIC 2010 dataset

5. CONCLUSIONS AND OPEN CHALLENGES

TAKE-AWAY KEY NOTES

(1) Lack of available datasets.

(1) Widespread datasetS (e.g. KDD99) are too old to reflect new attacks.

(2) Ideally, datasets should include most of the common attacks and correspond to current network environments.

(3) Constructing new datasets depends on expert knowledge, and the required cost is high.

(4) Datasets should be representative, balanced and have less redundancy and less noise.

(5) Systematic datasets construction and incremental learning may be solutions to this problem.

(2) Inferior detection accuracy in actual environments.

(1) ML methods often do not perform well on completely unfamiliar data.

(2) Most the existing studies were conducted using labeled datasets.

(3) If datasets do not cover all real-world samples, performance in actual environments is not guaranteed—even if the models achieve high accuracy on test sets.

(3) Low efficiency.

(1) Most studies emphasize the detection results and usually employ complicated models and extensive data preprocessing methods, leading to low efficiency.

(2) However, IDS need to detect attacks in real time. A trade-off between effect and efficiency is required.

(3) Parallel computing [66,67] approaches using GPUs [48,68,69] are common solutions.

FUTURE RESEARCH CHALLENGES

(1) **Utilizing domain knowledge.** Combining domain knowledge with machine learning can improve the detection effect, especially when the goal is to recognize specific types of attacks in specific application scenarios

- Combining ML/DL methods with rule-based systems, such as Snort [70–73], can result in IDSs with low false alarm rates and low missed alarm rates.
- For specific attacks, such as DOS [74–79], botnets [80], and phishing [81], proper features must be extracted according to the attack characteristics that can be abstracted using domain knowledge.
- For specific application scenarios, such as cloud computing [82,83], IoT [84–86], and smart grids [87,88], domain knowledge can be used to provide the environmental characteristics that are helpful in data collection and data preprocessing.

(2) **Improving ML/DL algorithms.** Improvements in ML/DL algorithms are the main means to enhance the detection effect.

- Compared with shallow models, DL methods learn features directly from raw data, and their fitting ability is stronger.
- DL models with deep structures can be used for classification, feature extraction, feature reduction, data denoising, and data augmentation tasks.
- Unsupervised learning methods require no labeled data; thus they can be used even when a dataset shortage exists. The usual approach involves dividing data using an unsupervised learning model, manually labeling the clusters, and then training a classification model with supervised learning [89–92].

(3) **Developing practical models.** Practical IDSs not only need to have high detection accuracy but also high runtime efficiency and interpretability.

- As real-time detection is essential, improving the efficiency of ML is a future research challenge.
- Reducing the time required for data collection and storage is also of concern.
- Interpretability is important for practical IDSs. Increasing the interpretability of DL models in IDS is an open challenge [93].

REFERENCES

Presentation based on:

[I] Liu, Hongyu, and Bo Lang. "Machine learning and deep learning methods for intrusion detection systems: A survey." applied sciences 9.20 (2019): 4396.

(see also the bibliography section of [I] for papers referenced in this presentation)

Additional material:

[II] Popoola, S. I., Adebisi, B., Ande, R., Hammoudeh, M., Anoh, K., & Atayero, A. A. (2021). smote-drnn: A deep learning algorithm for botnet detection in the internet-of-things networks. *Sensors*, *21*(9), 2985.

[III] Hijazi, A., El Safadi, A., & Flaus, J. M. (2018, December). A Deep Learning Approach for Intrusion Detection System in Industry Network. In *BDCSIntell* (pp. 55-62).

[IV] Short description of Python ML libraries: https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/

[V] Python code example for pcap file analysis: https://www.kaggle.com/code/rohitiitpatna/starter-ip-network-traffic-flows-40d90008-c

[VI] Python implementation in research: https://link.springer.com/article/10.1007/s42979-022-01031-1#Sec10