# Artificial Intelligence based Intrusion Detection System – A Detailed Survey

**Vishwas Sharma[1], Dharmesh Shah[2], Sachin Sharma[3], Sunil Gautam[4]**

PhD Scholar, Sankalchand Patel University, Visnagar, India[1]
Provost, Indrashil University, Mehsana, India[2]
Associate Professor, Department of Computer Engineering, Indrashil University, Mehsana, India[3]
Assistant Professor, Department of Computer Engineering, Nirma University, Ahmedabad, India[4]

vishwas.ece@gmail.com[1],djshah99@gmail.com[2],sharma.f@gmail.com[3], sunil.gautam@nirmauni.ac.in[4]

**Abstract***:* The Internet and communications have rapidly expanded, leading to a significant rise in data generation and heterogeneity. Intrusion detection systems play a crucial role in ensuring the security and integrity of computer systems. These systems have been developed by researchers, academicians, and practitioners to effectively detect and mitigate network attacks. Intrusion detection systems are designed to analyze network traffic and compare it with a baseline of normal behavior, allowing them to identify any deviations or inconsistencies that may indicate an intrusion. Furthermore, the cooperative and distributed architecture of intrusion detection systems enables them to effectively detect attacks and protect the network from unauthorized access. Additionally, to enhance the performance of intrusion detection systems, techniques such as resampling the dataset and utilizing classifier ensemble are used to improve the classification accuracy. Moreover, intrusion detection systems have been integrated with intrusion response systems to ensure a timely and effective response to detected attacks. AI-based Intrusion Detection Systems have emerged as a crucial tool in ensuring network security and combating cyber threats. These systems utilize artificial intelligence algorithms to analyze network traffic, identify patterns of malicious activity, and detect potential cyber-attacks. They have proven to be highly effective in improving the detection accuracy, reducing false alarms, and even detecting previously unknown types of attacks. In summary, the development of accurate and efficient intrusion detection systems is crucial for ensuring network security. In today's rapidly changing world, the significance of accurate intrusion detection systems cannot be overstated.

**Keywords**: machine learning; deep learning; feature selection; feature extraction; deep neural network;

## I. INTRODUCTION

Intrusion Detection Systems have become a crucial component in ensuring the security of computer networks. Their ability to monitor network activity and detect any suspicious or malicious behavior is essential in protecting sensitive data and preventing unauthorized access. AI-based Intrusion Detection Systems utilize machine learning algorithms to analyze network traffic patterns and identify anomalous activities. These systems can automatically learn and adapt to new threats, making them more effective in detecting and responding to cyber-attacks. By leveraging artificial intelligence and machine learning techniques, AI-based intrusion detection systems are able to analyze massive amounts of data, identify patterns, and detect anomalies that may indicate a potential intrusion or attack. Additionally, AI-based intrusion detection systems can enhance network security by continuously monitoring and analyzing network traffic in real-time. This allows them to quickly detect and respond to any suspicious activity, minimizing the potential damage caused by cyber-attacks. By leveraging machine learning techniques, AI-based IDSs can continuously learn and adapt to evolving threats, making them more resilient and capable of addressing the ever-changing.

In this paper, we aim to provide a comprehensive survey on different types of intrusion detection systems and the techniques proposed by leading researchers in the field. By conducting an in-depth investigation of IDS literature, we will categorize the anomaly-based IDS according to methods such as frequency-based, machine learning-based, statistical-based, and hybrid-based. This comprehensive survey will not only provide a holistic view of IDS implementation in the CAN bus network system but also help accelerate further research in this field.

### *A. Related Work*

The authors in [64], proposed an efficient DoS attack traffic detection method in this an Algorithm for detecting network attacks using a mix of Multilayer Perceptron with Random Forest. The effectiveness of the method and its usability are confirmed by the results of the evaluation of the method using the real network traffic CICIDS2017 dataset and UNSW-NB15 dataset. The study demonstrates that the model is capable of accurately classifying and detecting DoS assaults, with a 99.83 percent accuracy rate and a
93.51 percent false alarm rate.
The authors in [65], proposed a novel technique that use the self-adaptive density-based spatial clustering of apps with noise algorithm to identify low-rate DoS assaults. This method offers a flexible cluster discovery in multi-density datasets. The study showed that it enhances detection precision, decreases the rate of false negatives, and is applicable to large-scale complex network systems.
The authors in [66], proposed two novel techniques called GMM and UBM, which are normally used in other scientific or engineering areas, the use of a real traffic dataset to detect DoS/DDos cyber-attacks (CICIDS2017). Three experimental scenarios were implemented, including UBM, GMM, and a Random Forest alternative. The outcomes open up new opportunities for implementing these reducing techniques in new contexts and investigating novel solutions to pressing issues like DoS/DDoS cyber- attack detection, which is closely related to many services on the current Internet.
The authors in [69], Information Gain Ratio (IGR), Correlation (CR), and ReliefF are three filter-based feature reduction techniques that have been combined to produce a suggested feature reduction approach (ReF). The system initially calculates feature subsets based on the average weight for each classifier before implementing the Subset Combination Strategy (SCS).For the CICIDS 2017 DoS dataset, the suggested feature reduction strategy yields 24 reduced features. On the CICIDS 2017 dataset, the suggested technique outperforms the most advanced systems currently available. On the KDD Cup 99 dataset, the suggested technique has also been put to the test and contrasted with the most advanced systems currently available.

### *B. Contributions*

In this survey it is aimed at academics and developers who seek to create learning and understanding in AI based intrusion detection system utilizing the new ML/DL methodologies. The relevance of dimensionality reduction techniques for traditional ML models and DL algorithms are discussed in this study along with how these ideas enhance IDS. We can't say that we've examined every paper that uses ML/DL, but we've discussed the key strategies that have been discussed in this literature. We give a thorough overview of the cutting-edge machine learning and DL techniques used in IDS; Discuss advantages of FE and selection using traditional ML techniques; we investigate several Techniques and instruments for DL/ML deployment on IDS, ; Draw attention to the issues and difficulties with DL/ML.

### *C. Paper Organization*

This paper's structure includes the following: under Section II, The main ML based algorithms used in the IDS context are listed with their benefits and drawbacks. Next, we compare and summaries the ML/DL methods that were looked at in the table. The good efficiency of these algorithms that significantly improve the effectiveness of the model is examined in Section II (C) in relation to how these methods might be applied. It discusses dimensionality reduction strategies, including advantages and disadvantages. The major goal of Section III is thorough assessment of DL. Section IV summarizes this paper's conclusions in its final paragraph.

## II. Machine Learning

ML is a subfield in AI [18–20]. The popularity of ML algorithms in the 1990s was largely due to the rise of data. Currently, it is a developing field that draws interest from both academics and business. Its efficacy has been verified in a variety of application scenarios, including network traffic management, autonomous driving, computer vision, and healthcare. As an illustration, historical traffic information is employed to enhance traffic classification and lessen congestion. To put it another way, ML models typically go through two stages: (1) Developing a model involves using a training set; (2) The unknown data is then predicted or categorized using the model. As a result, machine learning (ML) might be compared to "programming by example".

*A. Supervised learning*

The majority of people employ this kind of learning methodology. When a predetermined class needs to be allocated

to an object based on a number of observable attributes specific to that object, it operates [23]. As a result, supervised learning seeks model parameters that, given a loss function L(y, y), best predict the data. Regression and classification are two instances of supervised learning; classification is one of the most used techniques since the outputs take on discrete values. In regression, the data are transformed into a real-valued variable by a learning function. The three methods of categorization are binary, multi-class, and multi-labeled [24]. For instance, binary classification can only categorize traffic as "attack" or "regular," leaving only two viable groups.;

*B. Unsupervised learning*

Unsupervised learning may be a viable option given the growing number and complexity of data. Without any prior knowledge of the desired properties, it can determine a relationship between the instances. Unsupervised learning specifically looks at the similarities between the cases to group them into various clusters. When compared to instances in different clusters, instances within the same cluster are more comparable [26];

*C. Semi-supervised learning*

Semi-supervised learning, as its name suggests, combines both supervised and unsupervised learning to benefit from each method. In contrast to supervise learning (all data are labelled), it tries to employ both labelled and unlabeled data to train the model (data all unlabeled). Semi-supervised learning seeks to reduce these issues by using a small number of labelled examples with a big amount of unlabeled data. It is difficult, time-consuming, and labor-intensive to label data (such as network traffic), particularly for attack detection [27]. When there is a lack of huge amounts of labelled data, this approach is appropriate. Because of this, the scientific community has shown an increasing interest in semi-supervised learning in recent years, particularly for traffic classification [28];

*D. Reinforcement learning (RL)*

Biological learning systems served as the fundamental inspiration for reinforcement learning. Just using the input, a computer programme receives from its surroundings after executing an action; RL differs from supervised and unsupervised learning. RL does not try to find patterns or learn from a training set of labelled data, but it does attempt to find patterns in real- world data [29]. In addition to supervised, unsupervised, and semi-supervised learning, it is recognized as a fourth ML approach for this reason. Also, the environment's availability of the training data defines RL. It is a method that enables an agent to develop new behaviors through interaction with the outside world. This learning strategy is built on three crucial components: observations, rewards, and action. As a result, the software agent performs actions and makes observations within an environment, receiving incentives as a result.

**Table 1.** A comparison of ML methods

| **Machine Learning** | **Merits** | **Demerits** |
|---|---|---|
| Supervised learning | Affordable, quick, and scalable | Needs data labelling and training, and performs badly with severely skewed data. |
| Unsupervised learning | Simply needs data samples, has the ability to spot undiscovered patterns, and produces labelling data. | Unable to provide accurate information |
| Semi Supervised learning | Uses both labelled and unlabelled data to learn | When we select the incorrect rate of unlabelled data, it could result in worse performance. |
| Reinforcement learning | Efficient when interacting with the environment is the sole way to gather information about it and can be used to solve difficult problems. | Convergence is slow, because it requires a lot of data and work. |

A summary of the DL architecture and models is offered in Section II (A) and the provided algorithms in Section III and used in the IDS context are the subject of our attention.

## III. Deep Learning

It is a subfield of machine learning (ML) that originated from neural networks (NNs) and allows algorithms to categories or predict data from big datasets without the use of explicit coding. In a number of areas, DL algorithms can perform more accurately than humans. While DL models may automatically extract knowledge in a hierarchical fashion from raw data by stacking layers, feature engineering tasks, such as feature selection, are necessary for traditional ML-based models. [35, 23].

The terms deep feature learning, deep structured learning, and hierarchical learning have also been used to describe DL in the literature [37]. Deep learning (DL), which outperforms conventional ML algorithms, is able to learn incredibly complicated patterns. Many DL methods are shown in Table 2.

Deep learning is superior to shallow neural networks in learning high-level features. With only one hidden layer, shallow ANN has a very small number of hidden layers compared to DL, which has many more layers (deep). A group of neurons, or learning units, make up each hidden layer. The DL model may learn complex patterns without being bound by the restrictions of linear functions thanks to its non-linearity. Network training time is impacted by the activation function selected [40]. ReLu networks really perform better in terms of convergence than sigmoid and tanh networks. [40]

**Table 2.** An overview of the various DL models applied in SDN

| **Approach** | **ML Model** | **Description** | **Merits** | **Demerits** |
|---|---|---|---|---|
| MLP | Supervised, unsupervised | MLP is a simple, three-layer ANN. The input layer, features extraction layer and output layer. These layers contain a number of neurons. | Simple to use. | Low performance, sluggish convergence, and memory-intensive. |
| AE | Unsupervised | The encoder, code, and decoder blocks make up the first three components of an autoencoder. The encoder transforms the input information into a code, an abstraction. The decoder uses the code to attempt to recreate the input features. | Features may be extracted from large, unlabeled datasets using this method, which eliminates the need for manual engineering. | The quality of the features is influenced by the model and its hyper parameters, although it is difficult to estimate size of the code layer. |
| CNN | Supervised, unsupervised | Convolution and pooling (subsampling) layers are used in the CNN class of deep learning, which is then followed by fully linked layers. In order to uncover meaningful patterns and reduce the dimensionality of the features, pooling and convolution layers are used. Next, categorization is done using layers that are fully connected. Applications involving image recognition make extensive use of it. | Weight sharing, feature extraction that is pertinent, and strong competitiveness. | To acquire the optimal features, a lot of hyper parameter adjustment, a large training dataset, and high computing cost are required. |
| LSTM | Supervised, | A development of recurrent neural networks, LSTM was created as a short-term memory solution (RNNs). It features built-in "gates" that may determine which data in a sequence should be kept and which data should be discarded. | Excellent for sequential information and effective with lengthy sequences. | High computational expense and great model complexity |

| | | | | |
|---|---|---|---|---|
| GRU | Supervised, | In 2014, GRU was proposed. Though it has fewer parameters, it is comparable to LSTM. Like LSTM, it performs well with sequential data. GRU is less sophisticated than LSTM. | More effective in terms of computation than LSTM. | Accuracy-wise less effective than LSTM. |
| DRL | Reinforcement | DRL makes use of both DL and RL to tackle more complex issues. In other words, DL makes it possible for RL to scale to previously unsolvable decision-making issues. | capable of scaling | In terms of training, slowly |
| DBN | Unsupervised, supervised | Many constrained Boltzmann machines stack DBN. The model parameters are initialized using the greedy learning technique, and the label is then used to fine-tune the entire model. | Unsupervised learning eliminates the need to label data for training or initialize the network correctly, which can help prevent local maxima and extract robust features. | cost of computation is high |

### *A. Conventional Machine Learning Models*

There are still a number of difficulties even if DL-based models excel in complex applications. Its hyperparameters, for instance, increase exponentially in number as the model depth increases. Finding an appropriate DL architecture and figuring out the best hyper parameters are also challenging challenges. Also, because DL requires a big volume of data to identify patterns in the data, it struggles to perform well when the volume of data is minimal. Conventional ML models (such as decision trees) can produce superior outcomes in this situation with a small amount of hyperparameter fine-tuning. Moreover, DL is more computationally expensive than traditional ML models, making it difficult to utilize on devices with a basic Processor.

The ensemble process mimics human inclination to consult with multiple people before reaching a choice. [ 44]. the most widely used ensemble models are bagging, boosting, and stacking generalization (or just stacking). There is no perfect ensemble strategy, just like there isn't one for single models, because different ensemble strategies work better in various situations. Table 3 lists several traditional ML techniques that are employed with SDN.

Yet, the quantity and caliber of features affect how well the traditional ML models perform. In this context, it takes a long time to extract a lot of information from the incoming flow, which can lower the system's QoS [45].

**Table 3.** Comparison of traditional ML methods applied in SDN.

| **Approach** | **ML Model** | **Description** | **Merits** |
|---|---|---|---|
| Decision tree | An attribute is represented by each internal node and a class label is represented by each leaf (terminal) of the tree-like structure in a DT. The root node is the node that sits at the very top of the tree. Information Gain and the Gini Index are examples of tree splitting techniques [46]. | Easy to comprehend and analyse, minimum data preparation is needed, numerous categories of data , and high dimension data are simply processed | Produces a complex tree with numerical data and needs a lot of space. |
| Random forest | One of the most well-known supervised machine learning approaches for regression and classification,. Decision | effective at preventing over-fitting | Large training dataset is necessary, making it unsuitable for RT |

| | | | |
|---|---|---|---|
| | trees are used to build random forests, as their name would imply. It employs the bagging technique, which improves performance. | | applications. |
| Support vector machine | SVM is an effective classifier that may be applied to both regression and classification issues. It is, however, mostly employed as a categorization tool. | Scalable; can manage complicated data | Although selecting the proper kernel function requires a lot of calculation, there is no theorem for doing so. |
| K-nearest neighbour | The KNN supervised model's guiding principle is "Tell me who your friends are, and I'll tell you who you are" [48]. | It is simple to use, performs well with straightforward issues, and is effective even for non-expert users. | It takes a long time to test the model, requires a lot of storage space, and isn't suitable for real-time classification |
| K-means | K-means is a well-known unsupervised model. Based on a similarity metric, the data may be grouped into K clusters, and observations within a cluster are quite comparable to those within other clusters. | quick, easy, and less complicated | Requires several clusters in advance and is unable to manage outliers. |
| Boosting algorithms | Boosting's main objective is to improve any model's performance, even that of struggling students (i.e., XGBoost, AdaBoost, etc. | high precision, strong resistance to underfitting | It is costly to compute and difficult to determine the ideal parameters |

### *B. Feature Reduction*

The features on which standard ML algorithms are applied typically determine their effectiveness. In other words, the features utilized to characterize each item to the ML models have an impact on their performance in addition to the parameters [50]. Generally, the dataset could include irrelevant and duplicated characteristics, increasing the cost of processing and storage in a datasets context and lowering model performance. While redundant features are those that are covered by other features, relevant features are those that offer information that is helpful. [51,52]. Several machine learning (ML) algorithms, for instance, logistic regression has a time complexity of O ($mn^2 + n^3$), where n is the number of features and m is the number of instances [53]. One of the most significant problems in machine learning is what is known as the "curse of dimensionality," which refers to these difficulties. Bellman first mentioned the "curse of dimensionality" in 1961 [18]. Each model is also only as excellent as its available features. Feature reduction tasks are challenging because they must simultaneously solve two challenges: increase learning capacity and decrease computing cost and delay by lowering the amount of features.

The amount of information that is useful for the learning process is retained while the original material is reduced. So, data reduction is crucial in order to develop accurate ML models for a dataset with a big number of features in a fair period of time. Each technique can be used individually or in combination to improve performance, including the precision of the said material [55]. To enhance classification performance, Author [56] presented techniques for dimensionality reduction that combine feature extraction with feature selection. The effectiveness of three-dimensionality reduction strategies was then evaluated using three datasets. These combinations enhance the classifier's precision, recall, and F-measure, according to their findings. In addition, there is a feature building method that is distinct from feature reduction.

**Table 4.** Comparing various methods of dimension reduction [16,57].

| Approach | Merits | Demerits |
|---|---|---|
| Filter | Affordable, quick, and scalable | Disregards how you interact with the classifier |
| Wrapper | Interaction with the classifier and | Risk of over-fitting, slow, and pricey for |

| | competitive classification accuracy | large feature spaces |
|---|---|---|
| Feature extraction | Dimension reduction without information loss | Regarding the initial characteristics, nothing is known. |

### *C. Feature Selection*

The feature selection methodology, which evolved into a fundamental element of traditional machine learning algorithms, reduces the data size (such SVM, DT, etc.). the target variable's "optimally" described by a subset of attributes. A method for enhancing classification performance, regression performance, and cluster efficiency by choosing the most advantageous traits from a starting set of features. Since that feature selection for classification requires a lot more work than other ML tasks, we will concentrate on it in this section (e.g., clustering). The least amount of dimensions that significantly increase accuracy make up the best subgroup. In order to execute a more focused and efficient analysis, it assists us in determining the crucial features and weeding out the unnecessary ones from the dataset. There are four main steps in the feature selection process, which are as follows:

1. An examination of prospective feature subsets in accordance with a search strategy is done using the search technique known as subset generation;
2. The objective of subset assessment is to quantify an attributes capacity to discriminate between the specified variables;
3. Limiting criteria specify the point at which the feature selection process should be stopped;
4. In order to verify the reliability of the chosen features, result validation is conducted.

Many assessment measures have been offered in various research publications to gauge the caliber of the proposed subset of attributes. Evaluation criteria can be classified as independent or dependent depending on how much they rely on ML algorithms [16]. We have a wide range of measures, including accuracy, for dependent criteria.. Below is a description of the two primary categories that can be used to categories feature selection: filters and wrappers.

*Filter Method:* Before utilizing any classification approach, it is best to first select the optimal feature set using some independent criterion. The computational effectiveness of the filter methods allows for the selection of characteristics from high-dimensional data sets. Depending on the kind of data, it can alternatively be characterized as a binary or continuous feature selection technique. Each feature is assigned a weight using feature-weighting algorithms, which then rank the features according to their weights. These methods may be computationally cheap and not to account the relationships between the characteristics, which could result in the selection of redundant features. Subset search algorithms have gained popularity because of this. Subset search methods examine possible feature subsets using evaluation metrics that reflect the usefulness of each subset, such as correlation ;

*Wrapper Method*: employs the performance of the learning algorithm as the evaluation criterion. It performs a search operation in the universe of potential features to calculate a single learning algorithm's expected accuracy. Different search methods are available, including forwarding direction (which starts with an empty set and adds the most relevant features one at a time) and backward direction (which starts with a full set and deletes less relevant features one at a time).

*Feature Extraction*

The mapping function F is used in feature extraction to turn the original variables into new features while keeping the majority of the important data. The original attributes can be combined in a linear or non-linear way to produce this transformation. For instance, we extract a new feature set called f1′ from n features called f1, f2, f3,..., fn.

Where m < n and fi′ = F ( f1, f2, f3, . . . , fn). Frequently, without significantly reducing the amount of information from the original attribute space, the feature extraction technique can reduce the feature space. A negative aspect of it, is that it frequently loses track of how the original characteristics contribute [52]. As a result, it is nearly impossible to analyses the new characteristics because feature extraction tools cannot derive a physical meaning for the changed data. We explore three of the most popular techniques in this poll.

PCA is the earliest method of multivariate analysis. It is a non-parametric, unsupervised method for reducing the dimensionality of data. PCA does have several restrictions, which are described below:

- It posits a linear relationship between the variables;
- This depends on the data scaling;

AE is an unsupervised learning model that aims to recreate the hidden layer's input. The AE attempts to reduce the reconstruction error throughout the process by enhancing the learned feature's quality. It is widely used to discover distinguishing characteristics of original data.

*Federated Learning*

A decentralized learning strategy called federated learning (FL) is employed to protect data privacy and reduce message

transmission between the server and clients. The worldwide model is subsequently broadcast to several clients. Using their own data for a predetermined amount of epochs, each client trains the global model based on this model. In order to do the global aggregate, they then submit the revised model back to the FL server. For a certain number of rounds, these processes must repeat until the global model is reached in order to achieve appropriate performance. FL makes it possible for the finished model to make money from different clients without sharing data as a result. Also, by sending the model parameters rather than the customers' data, it might reduce the communication overhead. Given the increased traffic data. It maintains the flow of traffic in the same area. FL has consequently begun to draw researchers' attention.

## IV. Conclusions

An AI-based survey on intrusion detection systems was carried out, with particular attention paid to the function of deep learning in intrusion detection, the significance of intrusion detection datasets, and the efficacy and efficiency of the sug gested methods. The survey's conclusions demonstrate that even with today's cutting-edge intrusion detection technologies, there is still potential for improvement. To improve AI-based intrusion detection systems' accuracy and functionality, more study and development are required. Additionally, the relevant experimental solutions put forth as deep learning-based IDSs are compared and discussed in this survey. The survey determines the optimal technological approach for detecting intrusions in dynamic settings and varied network technologies by assessing the experimental tests.

We begin with a brief introduction to ML and DL in intrusion detection system, highlighting the background context and motivating factors for these technologies. It demonstrated how ML and DL methods can contribute in intrusion detection. Also, we have outlined the fundamental ideas and sophisticated principles of a number of ML and DL models, and their advantages and disadvantages. We have also discussed the advantages of feature selection and extraction strategies for dimensionality reduction when used with traditional ML models. Additionally, it is demonstrated how ML/DL models may assist with intrusion detection in a number of ways, including classification of attacks, prediction, and security. Yet, there are a number of problems and difficulties that might arise for intrusion detection. Thus, we talked about the issues that require research both in academia and industry. This survey intends to analyses key components linked to the AI based Intrusion detection system.

## Abbreviations:

| *Abbreviations* | *Description* | *Abbreviations* | *Description* |
|---|---|---|---|
| SDN | Software-Defined Networking | DR | Dimensionality reduction |
| ML | Machine learning | OF | Open Flow |
| DL | Deep learning | FL | Federated learning |
| ANN | Artificial neural network | DRL | Deep Reinforcement Learning |
| AE | Autoencoder | NGMN | Next Generation Mobile Networks |
| PCA | Principle Component Analysis | DBN | Deep belief network |

## REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2023 WhitePaper; Cisco: San Jose, CA, USA.
[2] Ayoubi, S.; Limam, N.; Salahuddin, M.A.; Shahriar, N.; Boutaba, R.; Estrada-Solano, F.; Caicedo, O.M. Machine learning for cognitive network management. IEEE Commun. Mag. 2018, 56, 158–165.
[3] Mestres, A.; Rodriguez-Natal, A.; Carner, J.; Barlet-Ros, P.; Alarcón, E.; Solé, M.; Muntés-Mulero, V.; Meyer, D.; Barkai, S.; Hibbett, M.J.; et al. Knowledge-defined networking. ACM SIGCOMM Comput. Commun. Rev. 2017, 47, 2–10.
[4] Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Wang, C.; Liu, Y. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. IEEE Commun. Surv. Tutor. 2018, 21, 393–430.
[5] Latah, M.; Toker, L. Application of Artificial Intelligence to Software Defined Networking: A survey. Indian J. Sci. Technol. 2016, 9, 1–7.
[6] Latah, M.; Toker, L. Artificial Intelligence enabled Software-Defined Networking: A comprehensive overview. IET Netw. 2018, 8, 79–99.
[7] Zhao, Y.; Li, Y.; Zhang, X.; Geng, G.; Zhang, W.; Sun, Y. A survey of networking applications applying the Software Defined Networking concept based on machine learning. IEEE Access 2019, 7, 95397–95417.

[8] Thupae, R.; Isong, B.; Gasela, N.; Abu-Mahfouz, A.M. Machine learning techniques for traffic identification and classification in SDWSN: A survey. In Proceedings of the 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 4645–4650.
[9] Mohammed, A.R.; Mohammed, S.A.; Shirmohammadi, S. Machine Learning and Deep Learning based traffic classification and prediction in Software
Defined Networking. In Proceedings of the IEEE International Symposium on Measurements & Networking (M&N), Catania, Italy, 8–10 July 2019;
[10] Boutaba, R.; Salahuddin, M.A.; Limam, N.; Ayoubi, S.; Shahriar, N.; Estrada-Solano, F.; Caicedo, O.M. A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. J. Internet Serv. Appl. 2018, 9, 1–99.
[11] Sultana, N.; Chilamkurti, N.; Peng, W.; Alhadad, R. Survey on SDN based network intrusion detection system using machine learning approaches. Peer- Netw. Appl. 2019, 12, 493–501.
[12] Nguyen, T.N. The challenges in SDN/ML based network security: A survey. arXiv 2018, arXiv:1804.03539.
[13] Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M.P.; Shyu, M.L.; Chen, S.C.; Iyengar, S. A survey on deep learning: Algorithms, techniques, and applications. ACM Comput. Surv. (CSUR) 2018, 51, 1–36.
[14] Deng, L. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. Signal Inf. Process. 2014, 3, e2.
[15] Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of the Science and Information Conference, London, UK, 27–29 August 2014; pp. 372–378.
[16] Liu, H.; Yu, L. Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEETrans. Knowl. DataEng. 2005,
[17] 491–502. 17. Kobo, H.I.; Abu-Mahfouz, A.M.; Hancke, G.P. A survey on Software-Defined Wireless Sensor Networks: Challenges and design requirements. IEEE Access 2017, 5, 1872–1899.
[18] Domingos, P. A few useful things to know about machine learning. Commun. ACM 2012, 55, 78–87.
[19] Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. Science 2015, 349, 255–260.
[20] Mitchell, T.M. Machine Learning; McGraw-Hill: New York, NY, USA, 1997.
[21] Bengio, Y.; Lee, H. Editorial introduction to the neural networks special issue on deep learning of representations. Neural Netw. 2015, 64, 1–3.
[22] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 2011, 12, 2825–2830.
[23] Zhang, G.P. Neural networks for classification: A survey. IEEE Trans. Syst. Man, Cybern. 2000, 30, 451–462.
[24] Pacheco, F.; Exposito, E.; Gineste, M.; Baudoin, C.; Aguilar, J. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. IEEE Commun. Surv. Tutor. 2019, 21, 1988–2014.
[25] Aouedi, O.; Piamrat, K.; Parrein, B. Performance evaluation of feature selection and tree-based algorithms for traffic classification. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada 14–23 June 2021.
[26] Tomar, D.; Agarwal, S. A survey on Data Mining approaches for Healthcare. Int. J. Bio-Sci. Bio-Technol. 2013, 5, 241–266.
[27] Zhu,X.J. Semi-SupervisedLearningLiteratureSurvey; TechnicalReport; UniversityofWisconsin-MadisonDepartmentofComputer Sciences: Madison, WI, USA, 2005.
[28] Aouedi, O.; Piamrat, K.; Bagadthey, D. A semi-supervised stacked autoencoder approach for network traffic classification. In Proceedings of the 2020 IEEE 28th International Conference on Network Protocols (ICNP); Madrid, Spain, 13–16 October 2020.
[29] Sutton, R.S.; Barto, A.G.; others. Introduction to Reinforcement Learning, MIT Press: Cambridge, UK, 1998; Volume 135.
[30] Watkins, C.J.; Dayan, P. Q-learning. Machine Learn. 1992, 8, 279–292.
[31] Boyan, J.A.; Littman, M.L. Packet routing in dynamically changing networks: A reinforcement learning approach. In Advances in Neural Information Processing Systems. Available online: https://proceedings.neurips.cc/paper/1993/hash/4ea06fbc83cdd0a060 20c35d50e1e89a-Abstract.html (accessed on 30 December 2021) .
[32] Bitaillou, A.; Parrein, B.; Andrieux, G. Q-routing: From the algorithm to the routing protocol. In Proceedings of the International Conference on Machine Learning for Networking, Paris, France, 3–5 December 2019; pp. 58–69.
[33] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. arXiv 2013, arXiv:1312.5602.
[34] Li, Y. Deep reinforcement learning: An overview. arXiv 2017, arXiv:1701.07274.
[35] Ketkar, N.; Santana, E. Deep Learning with Python; Springer: Berlin/Heidelberg, Germany, 2017; Volume 1.
[36] Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. Inf. Fusion 2018, 42, 146–157.
[37] Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35, 1798– 1828. [CrossRef] [PubMed]
[38] Mayer, R.; Jacobsen, H.A. Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools. ACM Comput. Surv. (CSUR) 2020, 53, 1–37.
[39] Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. IEEE Access 2019, 7, 53040–53065.
[40] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 2012, 25, 1097– 1105.
[41] Haykin, S. Neural Networks: A Comprehensive Foundation; Prentice-Hall: Hoboken, NJ, USA, 2007.
[42] Woz´niak, M.; Grana, M.; Corchado, E. A survey of multiple classifier systems as hybrid systems. Inf. Fusion 2014, 16, 3–17.
[43] Li, Y.; Pan, Y. A novel ensemble deep learning model for stock prediction based on stock prices and news. Int. J. Data Sci. Anal. 2021, 1–11.
[44] Polikar, R. Ensemble based systems in decision making. IEEE Circuits Syst. Mag. 2006, 6, 21–45.
[45] Tang, T.A.; Mhamdi, L.; McLernon, D.; Zaidi, S.A.R.; Ghogho, M. Deep learning approach for network intrusion detection in Software Defined Networking. In Proceedings of the 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, 26–29 October 2016;
pp. 258–263.
[46] Freund, Y.; Mason, L. The alternating decision tree learning algorithm. In Proceedings of the 16th International Conference on Machine Learning (ICML), Bled, Slovenia, 27–30 June 1999; Volume 99, pp. 124–133.
[47] Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32.
[48] Abar, T.; Letaifa, A.B.; El Asmi, S. Machine learning based QoE prediction in SDN networks. In Proceedings of the 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 1395–1400.
[49] Schapire, R.E. The boosting approach to machine learning: An overview. In Nonlinear Estimation and Classification; Springer: New York, NY, USA, 2003;
pp. 149–171.

[50] Dainotti, A.; Pescape, A.; Claffy, K.C. Issues and future directions in traffic classification. IEEENetw. 2012, 26, 35–40.
[51] L'heureux, A.; Grolinger, K.; Elyamany, H.F.; Capretz, M.A. Machine learning with big data: Challenges and approaches. IEEE Access 2017, 5, 7776– 7797.
[52] Janecek, A.; Gansterer, W.; Demel, M.; Ecker, G. On the relationship between feature selection and classification accuracy. In Proceedings of the New Challenges for Feature Selection in Data Mining and Knowledge Discovery, Antwerp, Belgium, 15 September 2008; Volume 4, pp. 90–105.
[53] Chu, C.T.; Kim, S.K.; Lin, Y.A.; Yu, Y.; Bradski, G.; Olukotun, K.; Ng, A.Y. Map-reduce for machine learning on multicore. Adv. Neural Inf. Process. Syst. 2007, 19, 281–288.
[54] Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene selection for cancer classification using support vector machines. Mach. Learn. 2002, 46, 389–422.
[55] Motoda, H.; Liu, H. Feature selection, extraction and construction. Commun. IICM (Institute Inf. Comput. Mach. Taiwan) 2002, 5, 2.
[56] Rangarajan, L.; others. Bi-level dimensionality reduction methods using feature selection and feature extraction. Int. J. Comput. Appl. 2010, 4, 33–38.
[57] Pal, M.; Foody, G.M. Feature selection for classification of hyperspectral data by SVM. IEEE Trans. Geosci. Remote Sens. 2010, 48, 2297–2307.
[58] Stadler, R.; Pasquini, R.; Fodor, V. Learning from network device statistics. J. Netw. Syst. Manag. 2017, 25, 672–698. [CrossRef]
[59] Da Silva, A.S.; Machado, C.C.; Bisol, R.V.; Granville, L.Z.; Schaeffer-Filho, A. Identification and selection of flow features for accurate traffic classification in SDN. In Proceedings of the 14th International Symposium on Network Computing and Applications,Cambridge, MA, USA, 28–30 September 2015; pp. 134–141.
[60] Xiao, P.; Qu, W.; Qi, H.; Xu, Y.; Li, Z. An efficient elephant flow detection with cost-sensitive in SDN. In Proceedings of the 1st International Conference on Industrial Networks and Intelligent Systems (INISCom), Tokyo, Japan, 2–4 March 2015; pp. 24–28.
[61] Wang, P.; Lin, S.C.; Luo, M. A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In Proceedings of the IEEE International Conference on Services Computing (SCC), San Francisco, CA, USA, 27 June–2 July 2016; pp. 760–765.
[62] Amaral, P.; Dinis, J.; Pinto, P.; Bernardo, L.; Tavares, J.; Mamede, H.S. Machine learning in software defined networks: Data collection and traffic classification. In Proceedings of the 24th International Conference on Network Protocols (ICNP), Singapore, 8–11 November 2016; pp. 1–5.
[63] Zhang, C.; Wang, X.; Li, F.; He, Q.; Huang, M. Deep learning-based network application classification for SDN. Trans. Emerg. Telecommun. Technol. 2018, 29, e3302.
[64] Hongyan He, Guoyan Huang, Bing Zhang, Zhangqi Zheng, "Research on DoS Traffic Detection Model Based on Random Forest and Multilayer Perceptron", Security and Communication Networks, vol. 2022, Article ID 2076987, 11 pages, 2022. https://doi.org/10.1155/2022/2076987
[65] D. Tang, S. Zhang, J. Chen, and X. Wang, "The detection of low-rate DoS attacks using the SADBSCAN algorithm," Information Sciences, vol. 565, pp. 229–247, 2021.
[66] J. S. M. Osorio, J. A. V. Tejada, and J. F. B. Vega, "Detection of DoS/DDoS attacks: the UBM and GMM approach," in Proceedings of the 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 866–871, IEEE, France, May 2021.
[67] S. S. T. Reddy and G. K. Shyam, "An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment," Applied Soft Computing, vol. 100, Article ID 106997, 2021.
[68] J. A. Perez-Diaz, I. A. Valdovinos, and K. K. R. Choo, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," IEEE Access, vol. 8, no. 99, pp. 155859–155872, 2020.
[69] D. Kshirsagar and S. Kumar, "An efficient feature reduction method for the detection of DoS attack," ICT Express, vol. 7, no. 3, pp. 371–375, 2021.