Math. Appl. 13(2)2024, 1–12
DOI: https://doi.org/10.13164/ma.2024.13201

A COMPARATIVE PERFORMANCE EVALUATION OF NETWORK INTRUSION DETECTION USING MACHINE AND DEEP LEARNING ALGORITHMS

IBTIHAL MOHAMMED KHALEEL[1*]

Abstract: *CICIDS2017 dataset used in this research paper comes from the Canadian Institute for Cybersecurity and comprises of labelled and tagged benign and botnet traffic in PCAP files. We also used three NSL KDD IoT datasets - kddubei (Ghernaouti, 2012), kddprediction (2012), kddaccessb (2012) obtained from the Australian Defence Science and Technology Group. In this paper, we perform a systematic and comparative performance evaluation of commonly used machine learning and deep learning algorithms to detect network intrusions. We evaluated the performance of these classification algorithms against four different datasets. The main focus being the use of these algorithms specifically in NIDSs, our comparative analysis will also focus on how these algorithms fare under various NIDS specific issues such as imbalanced data, scoring, etc. In recent times, there has been a growing trend of using machine learning (ML) and deep learning (DL) algorithms. Although much research has been conducted on IDSs, especially using NSL-KDD datasets, that propose various methods to detect intrusions and achieve high detection rates, the use of ML and DL in NIDSs is still relatively new, and especially there are only a scarce number of studies that perform a comprehensive and systematic comparison of the performances of various machine learning algorithms and deep learning models. NIDSs are one of the most prominent components of any network's security infrastructure. With the rise in cyber-crime, NIDS usage and deployment have become an essential need. A number of approaches have been devised to detect malicious traffic across the networks, and all of them employ different techniques to either store the signatures of known intrusions and then use these signatures to match the incoming traffic packets through machine learning algorithms.*

INTRODUCTION

The standard field of NIDS was developed to categorize a universal limit of rules in community communication. However, a huge variety of the structure completely outperformed the static talents. Under a smaller era of the model, network structures have gained from factory knowledge and destiny data requirements under NIDS preparation. Just like NIDS examination, they are created through structures of hidden nodes fed on a large number of perhaps protected statistics that require to be apprehended, so that certain non-linear Olin's can be adjusted individually to discover other patterns. The increase in community stakes and the complexity develops a direct trend in using mechanized gears to aid in solving the problem of cybersecurity. The network intrusion detection (NIDS) systems monitor packet exchange associated with network connections with the main goal of identifying unauthorized usage [30].
Nowadays, the actual communication and exchange of information are entirely

[1] Al Farahidi University, Computer Science, Turkey. Email: Ibtihal.mohammed@uoalfarahidi.edu.iq
Correspondence: Ibtihal.mohammed@uoalfarahidi.edu.iq

Keywords: Network Intrusion Detection Systems, Machine Learning Algorithms, Deep Learning Models, CICIDS2017 and NSL-KDD Datasets, Comparative Performance Evaluation.

dependent on the internet. But when the concept of online services was introduced, the internet was not designed to take into account security aspects. Security has become a significant and everlasting issue in the World Wide Web (WWW) over the last few years. Network security has been the main point of concentration for the last few decades after the inception of Field-Vital Networking Working Groups. As a result, a bunch of accelerating detection approaches have been introduced to protect systems and cell security. Network Intrusion Detection System (NIDS) is a very important component for any region of system security. Many cyber organizations tested their strength of securitization by building different types of NIDS [23].

LITERATURE REVIEW

NetGuard, an efficient and scalable network intrusion detection framework, is proposed. The machine learning-based network traffic classifier is introduced, which defends against malware algorithms and botnets from sophisticated distributed or direct attacks [18]. It reduces the risk of false positive and false negative as well. A new RAMBO technique is introduced, which employs semi-supervised learning for training and updating the learning model by an incremental learning algorithm. With the help of network flow features, they train and update the machine learning model in real time. Nitin and Dua proposed a new method to develop the Intrusion Detection System with a hybrid genetic algorithm and Random Forest classifier [26]. The proposed method uses KDD-Cup 99 Dataset for experimentation. Due to the stationary distribution of data instances, the KDD Dataset is not able to detect the traffic. The machine learning model identifies only the majority classes due to their high distribution in the data instance set [38].

In this section, we discuss the literature review on different machine learning algorithms for the network and performance monitoring systems. Some of the works used Support Vector Machine, Random Forest, K-Nearest Neighbor, and Deep learning on NSL-KDD dataset specifically. Subbalakshmi and Gnanadurai used Naive Bayes, Decision Trees, K-Nearest Neighbor, and Neural Network algorithm on KDD-Cup 99 and DARPA dataset [22]. Zakariah, et al. [43] described an artificial immune system that monitors the IT security of the network, whereas Medhat and Hassan utilize the feature selection technique for intrusion detection with different types of attacks in the network based on the machine learning algorithm. Alanazi and Hassan used Multilayer Perceptron (MLP) and Radial Basis Function (RBF) for the multiclass classification problem, and their results show the accuracy of 89.52% and 92.21% respectively for the detection of intrusion in the network. Many of the studies discussed in table-1 show that the machine learning algorithm can play a significant role in the detection of intrusion in the network as compared to the traditional rule-based methods [13].

METHODOLOGY

In the proposed comparative analysis of the various models, we first prepare the dataset. We scale and normalize the NSL-KDD dataset, which is an extension of KDD 99. The unwanted categorical features are encoded, and then the dataset is standardized to improve the performance of the model. These attributes are divided into different categories, which are separated as (i) basic features, (ii)

traffic features, and (iii) content features. The features table allows the observation of features involved in the dataset. We perform a feature selection process to make the dataset more effective and efficient with the use of the Recursive Feature Elimination (RFE) approach [43]. Initial experimental results have shown that RFE with Standard Function (SF) has proven effective. Then the classification models are finally applied to evaluate the network intrusion. For our study, three classification algorithms are employed to explore the network traffic detection with a deep learning approach. Millisec, while R2L and U2R attacks are detected based on the observation of the previous two trials [15].

The development of a network intrusion detection system (IDS) is essential to ensure the security of the network. The IDS uses a classification algorithm to distinguish between normal and abnormal activities to prevent and respond to malicious activities. Machine learning (ML) and deep learning (DL) are the best set of classification techniques and are very popular in the field of network activity mining. Researchers at various levels are conducting various experiments to ensure the level of performance of different classifiers. Considering this, the proposed work aims to provide a comparative performance evaluation of various ML and DL models in network intrusion detection [37]. The evaluation and comparison of classification techniques are done with the help of NSL-KDD dataset, which is a benchmark for finding the effectiveness of different models. The paper is implemented in three stages. The first phase consists of dataset preparation followed by feature selection in the second phase. The third and final phase consists of training and testing different models to predict the class labels [10].

Dataset Collection

The CICIDS-2017 dataset has been chosen to better which is publicly available freely, versatility and mobility were raised after production by softmax normalization. It is situated into 2 subsets: the original and the big. The big data is considered to have much more versatility in detecting intrusions, which is included nearly 40 GB, while the other is 6 GB. Both subsets have 80 features. Its 5 file formats have various included attack commands shown as follows: Thursday and Friday include DDoS attacks, Thursday contains DoS attacks [40]. Furthermore, other labs with no attack commands were shared as normal traffic. We separated files in terms of a kind of commands and clarified directions. Into the mixed directory, each kind of command is separated into 2 folders, with any normal traffic flowing to the first and last into connection [34].

Tools comparison data collection. The NSL-KDD dataset is popular in the network intrusion detection field. It has 3 versions: full train and test, 10% train and test, cross-fold train and test, which are optionally used most in classifier performance measurement. But its 41 features store information about network connections, exactly 22 features describe the particulars of network connections [35]. It was synthesized to improve the out-of-date dataset KDD'99. In each connection, 29 features were generated on purpose to distinguish attacks from normal. An anomaly detection dataset was used in LSTM network to implement the LSTM model [3].

Feature Extraction

Originally, features could be extracted according to each level of the Internet protocol. Many research reports have evolved into practical projects such as the BRO intrusion

detection system, including Intel's other open-source software contributions for cybersecurity, and various data-driven inspection systems based on very high-speed commercial networks that are also available in the open-source market. WHOIS data sets and domain-ranked information and label transfer systems are also representative feature extraction methods, according to relationships between connections or by studying specific Web connection interaction patterns [25]. Although these procedures include consideration, no follow-up licenses data sets should be collected and experiments conducted when deciding the discrimination capabilities of the machine learning algorithm. In order to detect the unknown malware or possible new intrusions, the features must also contain complete behavioral data of users and network devices related to the event [20].
Feature selection is an important stage in machine learning and plays a vital role in network intrusion detection systems (NIDS). It highlights the fact that, while some machine learning algorithms may perform well, there are often far too many selections of which features to use for the significant reduction of the overall workload for intrusion detection [11]. Features need to be discriminating and relevant for the NIDS with as few false positives, and should be evaluated before being passed into the machine learning algorithms. Inspired by recent machine learning successes, researchers have extended deep learning to energy efficiency, data caching, and data analysis applications. Data mining is introduced to achieve computer systems with big data and low data industry standards of intrusion detection [32].

PREPROCESSING

We do not remove any feature prior to evaluation, but after preprocessing it, we reduce the dimensionality of the data with the VTS. Feature selection helps to reduce overfitting, improve accuracy, and the effective time of analysis. It leads to simplification of models, and therefore, they are more interpretable. It is important to always apply techniques like the scree-plot to ascertain feature importance or Mann-Whitney U test in binary classification problems. Feature selection methods should be implemented on broad lines. Our observation solicits more research in finding physically significant features of the NSL-KDD dataset [28].
Preprocessing steps comprise of four main activities: 1) Cleaning, 2) Normalization or scaling, 3) Feature engineering, and 4) Dimensionality reduction. For cleaning, we do the imputation of missing values, and for normalizing our data, we apply Min-max or Z-score standardization. For some algorithms like the K-Nearest Neighbor and in our case CNN, data with outliers affect the algorithm's performance, so we treat them accordingly to mitigate their effects. We apply three types of scaling methods to our data: Min-max, log, and Z-score respectively. These are common scaling methods [39].

MACHINE LEARNING ALGORITHMS

Decision Tree (DT)

A decision tree (DT) is a supervised learning algorithm used for classification and can be used in regression tasks. It builds a classifier by scrutinizing the input feature vector and produces a class label. It divides the decision space into rectangular regions and assigns a class label to each decision region. The regions

are constructed such that the estimated class labels throughout are as consistent as possible [14]. DT is appealing as it can be used to elucidate complex decision rules and is easy to interpret, unlike classifiers such as Artificial Neural Network and k-nearest neighbor classifiers. Supervised Decision Tree has both a training phase, in which the concept represented by a DT is derived for an input feature set, and a test phase, in which a learned model is used to classify new input samples [19]. DT works by repeatedly comparing the training's output, by making decisions based on input features using an appropriate threshold value. A learning algorithm known as ID3, which was designed to generate Decision Trees from training data that minimizes the average number of tests for its test phase, was first introduced by Quinlan [2]. The results of the machine learning algorithms without normalization shown in Table 1:

Table 1. Without Normalization Results [27].

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	FI-Scare (%)
Naïve Bayes (NB)	80.2	96	57	71
Decision Tree (DT)	88.4	96	87	91
K- Nearest Neighbors (KNN)	73.3	91	72	81
Random Forest (RF)	89.1	95	85	90
Support Vector Machine (SVM)	88.2	88	87	87

Linear Model

Table 2. With Normalization Results [27].

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	FI-Scare (%)
Naïve Bayes (NB)	75.8	97	52	68
Decision Tree (DT)	88.1	96	86	91
K- Nearest Neighbors (KNN)	87.1	95	82	89
Random Forest (RF)	89.5	96	86	91
Support Vector Machine (SVM)	87.5	86	88	87

A linear model is an ordinary machine learning model that has been widely used in a variety of classification problems. Its decision function is learned using a linear equation of the input features, which is f(x) = wTx + b. Here, w is the weight parameter, x is the input features, and b is the bias. The decision is made based on the sign of f(x), where if f(x) > 0, the sample belongs to one class, and if f(x) < 0, the sample belongs to another class. An example of a linear model is logistic regression. It implements linear regression algorithms using the logistic function as the decision function [41]. The logistic function is defined as logit(x) = f(x) = 1 / (1 + e^(-x)). Given the input features X of m instances and n features, where now each row m represents an instance and each feature n represents columns, the decision function of logistic regression is given as Z = f(X) = -1 / (1 + e^ (-XW + b)). Here, X is an instance, and W is the weight matrix with n rows and 10 columns. The results of the machine learning algorithms with normalization shown in Table 1:

DEEP LEARNING ALGORITHMS

8.2. Convolutional Neural Networks A convolutional neural network, which is also called a CNN, ConvNet, or shift invariant or space invariant artificial neural

network, is an artificial neural network made up of neurons that attempt to replicate the way a human visual cortex works. It is designed to process data such as images. CNNs were responsible for a number of breakthroughs in image classification. They were developed and popularized in recent years [12]. CNNs were inspired by the biological processes taking place in our brains. Prior to the decade of CNNs, the digit recognition algorithm by Yann LeCun and researchers in 1998 was the best among machine learning algorithms. However, this algorithm was used extensively in financial institutions and banks for recognizing handcrafted digits and checks; it was not widely known that CNNs were superior. LeCun and others improved their architecture in 1998 to achieve the best performance in most benchmarks at the time [8, 9]. The results of the deep learning algorithms shown in the Table 3:

Table 3. Summarized the Results of Deep Learning Algorithms [27].

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	FI-Scare (%)
Multilayer Perceptron Model (MLP)	96.88	97.21	96.90	97.05
Long-Short Term Memory (LSTM)	97.76	97.22	98.62	97.91

Deep Belief Networks

A deep belief network is a type of artificial neural network design based on restricted Boltzmann machines that are stacked on top of each other. It is formed through a combination of multiple layers of binary latent variables, which are modeled as Bernoulli distributions and connected by symmetrically weighted edges at each layer [4]. Deep belief networks are capable of modeling and capturing different levels of abstraction in data and are used as feature extractors in many machine learning algorithms such as deep learning and other hierarchical models. In 2006, Geoffrey E. Hinton proposed a fast, layer-by-layer pretraining algorithm for deep belief networks. Following the success of the pretraining method, researchers applied deep belief network models to tasks such as dimensionality reduction, image recognition, and others to achieve significant improvements over previous methods [31].

PERFORMANCE METRICS

Precision measures the proportion of examples that were correctly classified. It focuses on the relevant elements of the returned set, measuring the accuracy of the returned set, which is given in Equation 4. F-measure, also known as F-score, measures the algorithm's performance, which is given in Equation 5. The F-measure has been a valuable tool for evaluation in information retrieval and intrusion detection. It can be considered as a way of evaluating the performance of an algorithm with respect to both its precision (P) and recall (R) in the form of a single value. It has been generally recognized as a quick and relatively simple alternative to AUC [7].

In intrusion detection systems, there are significant factors used to evaluate the performance of the models. The overall performance measures of the models used in this research include accuracy (ACC), true positive rate (TPR), false positive rate (FPR), precision (PRE), and F-measure (F-SCALE) [29]. Accuracy measures the effectiveness of the classification algorithms by the proportion of true positive and true negative classifications to the total classifications. It is defined using Equation

1. True positive rate (TPR) measures the proportion of positive examples that were correctly classified and is defined in Equation 2. True negative rate measures the proportion of negative examples that were correctly classified, which is the complement of false positive rate. False positive rate measures the proportion of misclassified negative examples, that is, the total negative examples that were classified as positive [29].

EXPERIMENTAL RESULTS

In this study, the effectiveness of numeric models was examined over a multi-layer network which is a hostile connection environment that is kept isolated without having traffic. Choosing the best model among deep learning, machine learning, and numeric models was evaluated using accuracy and recall sensitive metrics. When the metrics are evaluated, the existing unbalanced nature was taken into account and, in this case, the model that achieved the best result in the recall sensitivity performance was preferred. After determining the classifier to be used in multiclass detections, the dense multiclass image of the confusion matrix was examined in terms of three desirable features that can be used in the training of artificial intelligence such as accuracy principle, speed of learning, good generalization, and size of false alarm [5].
Table 2 depicts the information regarding the data source that we used for the attack data collection. We used various numeric and deep learning models to evaluate the attacks on multiple layers of the anomalous environment. We have developed models classified as multiclass and binary based on the dataset description. We verified models with class weights and without class weights on the dataset with different activation functions, optimizers, batch sizes, and number of epochs parameters and chose the optimum solution among various test results [44]. We also developed models to evaluate the effectiveness of the multiclass model after the tuning process and described how the future can be performed by the alarms generated by the multiclass model. According to the obtained result, the activation function is preferred as 'LeakyReLU' with the number of neurons obtained at the end of the dimension reduction stages of the CNN with the parameterization of 3, 64 for the number of filters and kernel rows, respectively, and the tanh function has been preferred with the layer dense structure in the RNN [33]. Since the dataset has the feature of being unbalanced, several weight functions were also tried and their effects were evaluated.

DISCUSSION

We investigate the performance of Open World Detection. We have applied One-Dimensional Convolutional Neural Network using NSL-KDD and UNSW-NB15 as the datasets across the models: one with an interpretable model as a feature extractor and another with a non-interpretable model. Both model performances are evaluated. After analyzing the performances, the result indicates that using interpretable models actually degrades the performance and, at the same time, provides lower accuracy when being used in real-time. We empirically prove that using an interpretable model is unnecessary for detecting new attacks. Here's why: Deep Neural Network is inherently explained for identifying the unknown pattern which can be interpreted by humans. Using an interpretable model for Network Intrusion Detection is an old-fashioned way to interpret the model outputs which

are understandable yet non-explainable [21].
We have evaluated several machine learning and deep learning algorithms on two intrusion detection datasets. They are NSL-KDD and UNSW-NB15 datasets. The algorithms are Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes (NB), Decision Tree (DT), Logistic Regression (LR), and One-Dimensional Convolutional Neural Network. The purpose is to do a comparative evaluation of machine learning and deep learning algorithms' performance using different types of datasets. Performance is evaluated using accuracy, precision, recall, F1-Score, detection rate, True Positive Rate (TPR), and False Positive Rate. The evaluation results confirm that One-Dimensional Convolutional Neural Network is the best algorithm with the highest accuracy on both datasets: 92.90% in UNSW-NB15 and 99.88% in NSL-KDD [24].

LIMITATIONS

The low detection accuracy of the system is due to the high imbalance of different KDD-99 classes, varying patterns for the same activity, inefficient pre-processing strategies, time-consuming data alignment techniques, routine classifier settings, and inefficient classifier tunings. To address these limitations, in future works, we will modify the classification balance of the NSL-KDD dataset by adding difficult-to-learn classes, employ efficient classifiers such as support vector machine and random forest, normalize large ranges of attributes using logarithm transformations and entropy estimations, design data alignment strategies that minimize superfluous data, and implement a classifier tuning system that automatically assembles the most effective hyper-parameters for the network traffic data. We expect that these strategies will significantly improve machine and deep learning models in all dimensions [6].
In this study, several machine and deep learning classification algorithms are evaluated using network traffic data to detect network intrusion. The performance of these algorithms is assessed using simulation data and two benchmark datasets: UNSW-NB15 and NSL-KDD. Although the studied algorithms are effective in predicting normal and intrusive networking activities using the simulated data, the detector systems that use these algorithms possess performance limitations when using the UNSW-NB15 and NSL-KDD datasets. The systems have detection accuracy and classification efficiency less than 98% and 90%, respectively. The results also show that, in general, deep learning algorithms perform better in terms of accuracy and processing time than machine learning algorithms, despite potential overfitting, when used to detect new intrusive activities that were not seen during the training of the system [1].

FUTURE WORK

Hence, should these issues be taken into account appropriately, the proposed approach would provide significant added value in the field. The proposed model architectures and performance evaluation steps also have reached an advancement leap in their contribution roadmap. Additional comparative behavior evaluations, involving the highly influenced sensitivity and response time evaluations, can also be conducted to outline the full picture in the future [42].
Another potential aspect of this study is to explore the utility aspect to validate the prominence of the proposed feature selection approach and hyperparameter

optimization. Nevertheless, the overwhelming majority of previous works just aimed to report the detection performance only, leaving other important issues behind, such as the interpretation and configuration of the model behavior against temporal, study, and deployment-based challenges and the cost aspect [16].
Additionally, a hyperparameter fine-tuning for all models is proposed. Because the best hyperparameters for a model can be different from those of the others, and existing works in the network intrusion detection domain have optimized a specific hyperparameter or none at all. The model suggestion space is generally of a large scale and it should be scoped with the state-of-the-art hyperparameter optimization algorithms [17].
On the performance viewpoint, suitable feature selection schemes should be utilized. However, the obtained feature sets were tailored for correlation-based algorithms in the current study, and these feature sets may not deliver the optimal performance for generative-based algorithms, and the converse can be true. Hence, to outline a standard feature selection technique used in the field, and to select the most prominent features for both model architectures and all attack categories promised on the introductory aspects of the proposal [36].

CONCLUSION

An in-depth profile to identify deep learning models that are the best fit for our problem was considered in future work. A new deep feedforward model is developed and used to evaluate and compare the performance of 14 different supervised machine and deep learning models in network intrusion detection. The deep feedforward model is enhanced and tested with complex features of the benchmark dataset of the Intrusion Detection Evaluation Center (IDEC) contest. The IDSGAN is used to further generate synthetic network intrusion data for training the deep feedforward into a hybrid adversarial deep feedforward model. The experimental results for evaluating and comparing the performance of 14 different supervised machine and deep learning models using 41 different metrics are presented and discussed.
Deep learning algorithms are used to detect network intrusion. A flexible deep feedforward model is developed to accurately evaluate and compare the performance of the new deep learning in network intrusion detection. The new flexible deep feedforward model outperforms other existing machine learning and deep learning models by 97 percent and 98 percent in detection and prediction of 20 different types of attacks. A detailed comparative performance evaluation of the convolutional neural network and the deep feedforward model is done using 19 different types of attacks. Experimental results show that the deep feedforward model accurately detects and predicts normal traffic and the 19 types of attacks.

REFERENCES

[1] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, F. Ahmad, *Network intrusion detection system: A systematic study of machine learning and deep learning approaches,* Transactions on Emerging Telecommunications Technologies **32** (2021), e4150.
[2] M. Ali Qasimi, *Examining and Evaluating Classification Algorithms Based on Decision Trees,* J Electrical Electron Eng **3** (2024), 01-07.
[3] M. Almseidin, J. Al-Sawwa, M. Alkasassbeh, *Generating a benchmark cyber*

multi-step attacks dataset for intrusion detection, Journal of Intelligent & Fuzzy Systems **43** (2022), 3679-3694.

[4] A. S. Alqahtani, *RETRACTED ARTICLE: FSO-LSTM IDS: hybrid optimized and ensembled deep-learning network-based intrusion detection system for smart networks,* The Journal of Supercomputing **78** (2022), 9438-9455.

[5] A. H. Azizan, S. A. Mostafa, A. Mustapha, C. F. M. Foozy, M. H. A. Wahab, M. A. Mohammed, B. A. Khalaf, *A machine learning approach for improving the performance of network intrusion detection systems,* Annals of Emerging Technologies in Computing (AETiC) **5** (2021), 201-208.

[6] E. Beede, E. Baylor, F. Hersch, A. Iurchenko, L. Wilcox, P. Ruamviboonsuk, L. M. Vardoulakis, *A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy*, in: Proceedings of the 2020 CHI conference on human factors in computing systems, 2020, 1-12.

[7] P. Christen, D. J. Hand, N. Kirielle, *A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives,* ACM Computing Surveys **56** (2023), 1-24.

[8] Y. Chychkarov, A. Serhiienko, I. Syrmamiikh, A. Kargin, *Handwritten Digits Recognition Using SVM, KNN, RF and Deep Learning Neural Networks,* CMIS **2864** (2021), 496-509.

[9] K. G. Demirkaya, Ü. Çavuşoğlu, *Handwritten Digit Recognition With Machine Learning Algorithms,* Academic Platform Journal of Engineering and Smart Systems **10** (2022), 9-18.

[10] A. Devarakonda, N. Sharma, P. Saha, S. Ramya, *Network intrusion detection: a comparative study of four classifiers using the NSL-KDD and KDD'99 datasets,* Journal of Physics: Conference Series **2161** (2022), 012043.

[11] M. Di Mauro, G. Galatro, G. Fortino, A. Liotta, *Supervised feature selection techniques in network intrusion detection: A critical review,* Engineering Applications of Artificial Intelligence **101** (2021), 104216.

[12] A. A. Elngar, M. Arafa, A. Fathy, B. Moustafa, O. Mahmoud, M. Shaban, N. Fawzy, *Image classification based on CNN: a survey,* Journal of Cybersecurity and Information Management **6** (2021), 18-50.

[13] R. Gangula, M. M. Vutukuru, M. Ranjeeth Kumar, *Intrusion Attack Detection Using Firefly Optimization Algorithm and Ensemble Classification Model,* Wireless Personal Communications **132** (2023), 1899-1916.

[14] B. T. Jijo, A. M. Abdulazeez, *Classification Based on Decision Tree Algorithm for Machine Learning,* Journal of Applied Science and Technology Trends **2** (2021), 20-28.

[15] G. A. Kumar, A. Katiyar, K. Srinivasan, *Elevating IDS Capabilities: The Convergence of SVM, Deep Learning, and RFECV in Network Security*, in: 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), IEEE, 2024, 1-16.

[16] Y. Li, H. Mao, R. Girshick, K. He, *Exploring Plain Vision Transformer Backbones for Object Detection*, in: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella, T. Hassner (Eds.) Computer Vision – ECCV 2022, Springer Nature Switzerland, Cham, 2022, 280-296.

[17] X. Liu, C. Wang, *An empirical study on hyperparameter optimization for fine-tuning pre-trained language models,* arXiv preprint arXiv:2106.09204 (2021).

[18] Y. Liu, *Residential Network Security: Using Software-defined Networking to Inspect and Label Traffic*, PhD diss., Worcester Polytechnic Institute, 2021.

[19] B. Mahbooba, M. Timilsina, R. Sahal, M. Serrano, *Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model,* Complexity **2021** (2021), 6634811.
[20] R. Mills, A. K. Marnerides, M. Broadbent, N. Race, *Practical intrusion detection of emerging threats,* IEEE Transactions on Network and Service Management **19** (2021), 582-600.
[21] S. Neupane, J. Ables, W. Anderson, S. Mittal, S. Rahimi, I. Banicescu, M. Seale, *Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities,* IEEE Access **10** (2022), 112392-112415.
[22] M. K. Ngueajio, G. Washington, D. B. Rawat, Y. Ngueabou, *Intrusion Detection Systems Using Support Vector Machines on the KDDCUP'99 and NSL-KDD Datasets: A Comprehensive Survey*, in: K. Arai (Ed.) Intelligent Systems and Applications, Springer International Publishing, Cham, 2023, 609-629.
[23] E.-u.-H. Qazi, M. Imran, N. Haider, M. Shoaib, I. Razzak, *An intelligent and efficient network intrusion detection system using deep learning,* Computers and Electrical Engineering **99** (2022), 107764.
[24] E. U. H. Qazi, A. Almorjan, T. Zia, *A One-Dimensional Convolutional Neural Network (1D-CNN) Based Deep Learning System for Network Intrusion Detection,* Applied Sciences **12** (2022), 7986.
[25] S. V. B. Rakas, M. D. Stojanović, J. D. Marković-Petrović, *A Review of Research Work on Network-based Scada Intrusion Detection Systems,* IEEE Access **8** (2020), 93083-93108.
[26] M. Rigter, B. Lacerda, N. Hawes, *RAMBO-RL: Robust Adversarial Model-Based Offline Reinforcement Learning,* Advances in Neural Information Processing Systems **35** (2022), 16082-16097.
[27] M. Saber, A. A. Abdelhamid, A. Ibrahim, *Metaheuristic Optimization Review: Algorithms and Applications,* Journal of Artificial Intelligence and Metaheuristics **3** (2023), 21-30.
[28] M. A. Salam, A. T. Azar, M. S. Elgendy, K. M. Fouad, *The effect of different dimensionality reduction techniques on machine learning overfitting problem,* Int. J. Adv. Comput. Sci. Appl **12** (2021), 641-655.
[29] A. A. Salih, A. M. Abdulazeez, *Evaluation of Classification Algorithms for Intrusion Detection System: A Review,* Journal of Soft Computing and Data Mining **2** (2021), 31-40.
[30] L. Santos, R. Gonçalves, C. Rabadão, J. Martins, *A flow-based intrusion detection framework for internet of things networks,* Cluster Computing **26** (2023), 37-57.
[31] O. R. Shahin, H. H. Alshammari, A. A. Alzahrani, H. Alkhiri, A. I. Taloba, *A robust deep neural network framework for the detection of diabetes,* Alexandria Engineering Journal **74** (2023), 715-724.
[32] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, E. Alanazi, *Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey,* Journal of Network and Computer Applications **181** (2021), 103005.
[33] T. T. Sivri, N. P. Akman, A. Berkol, *Multiclass Classification Using Arctangent Activation Function and Its Variations*, in: 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), IEEE, 2022, 1-6.
[34] A. Siyaev, G.-S. Jo, *Towards Aircraft Maintenance Metaverse Using Speech Interactions with Virtual Objects in Mixed Reality,* Sensors **21** (2021), 2066.

[35] T. Su, H. Sun, J. Zhu, S. Wang, Y. Li, *BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset,* IEEE Access **8** (2020), 29575-29585.
[36] B. Subedi, V. E. Sathishkumar, V. Maheshwari, M. S. Kumar, P. Jayagopal, S. M. Allayear, *[Retracted] Feature Learning-Based Generative Adversarial Network Data Augmentation for Class-Based Few-Shot Learning,* Mathematical Problems in Engineering **2022** (2022), 9710667.
[37] A. Thakkar, R. Lohiya, *A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions,* Artificial Intelligence Review **55** (2022), 453-563.
[38] S. S. Tirumala, N. Nepal, S. K. Ray, *Raspberry pi-based intelligent cyber defense systems for SMEs and smart-homes: An exploratory study,* EAI Endorsed Transactions on Smart Cities **6** (2022), e4.
[39] S. Ullah, J. Ahmad, M. A. Khan, E. H. Alkhammash, M. Hadjouni, Y. Y. Ghadi, F. Saeed, N. Pitropakis, *A New Intrusion Detection System for the Internet of Things via Deep Convolutional Neural Network and Feature Engineering,* Sensors **22** (2022), 3607.
[40] K. Vamsi Krishna, K. Swathi, P. Rama Koteswara Rao, B. Basaveswara Rao, *A Detailed Analysis of the CIDDS-001 and CICIDS-2017 Datasets*, in: G. Ranganathan, R. Bestak, R. Palanisamy, Á. Rocha (Eds.) Pervasive Computing and Social Networking, Springer Nature Singapore, Singapore, 2022, 619-638.
[41] A. L. Yadav, K. Soni, S. Khare, *Heart Diseases Prediction Using Machine Learning*, in: 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, 2023, 1-7.
[42] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, A. Kanazawa, *Plenoctrees for real-time rendering of neural radiance fields*, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 5752-5761.
[43] M. Zakariah, S. A. AlQahtani, A. M. Alawwad, A. A. Alotaibi, *Intrusion Detection System with Customized Machine Learning Techniques for NSL-KDD Dataset,* Computers, Materials & Continua **77** (2023), 4025-4054.
[44] J. Zhang, L. Pan, Q.-L. Han, C. Chen, S. Wen, Y. Xiang, *Deep Learning Based Attack Detection for Cyber-Physical System Cybersecurity: A Survey,* IEEE/CAA Journal of Automatica Sinica **9** (2021), 377-391.