International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Next-Generation Network Intrusion Detection: Leveraging Deep Learning Techniques

G. Tarun Datta[1*], A.Sasi Vadana[2], A.Venkata Akhil[3], K.Mythily Sai Chandana[4], Venkata Vara Prasad Padyala[5]

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: The Network Intrusion Detection System (NIDS) plays a pivotal role as an indispensable tool for system administrators in the discernment and detection of network security breaches within their respective organizations. However, the advancement of a multifaceted and exceedingly proficient Network Intrusion Detection System (NIDS) is not devoid of its proportionate set of obstacles, particularly when confronted with unanticipated and capricious cyber assaults. We propose a ground-breaking and cutting-edge deep learning paradigm as the underpinning for the creation of an efficient and adaptable Network Intrusion Detection System (NIDS).The present study utilizes the Self Learning (STL) methodology, a sophisticated DL technique, for the purpose of analyzing the NSL-KDD dataset. This particular dataset is widely recognized as a benchmark in the field of network intrusion analyzing. In this study, we present the performance evaluation of the proposed methodology and conduct a comparative analysis with a range of previous research endeavors. The metrics being considered encompass a range of quantitative measures, including accuracy, precision, recall, and f-measure values.

Keywords: *Security of computer networks, Intrusion Detection Systems, advanced machine learning, streamlined data encoding, specific network intrusion dataset.*

1. Introduction

Network Intrusion Detection Systems (NIDSs) help network managers find and detect security breaches in an organization's network architecture. NIDSs carefully monitor and analyse the complex network traffic that passes through an organization's network devices' entry and egress points. It instantly sounds alerts to notify concerned parties of any unwanted infiltration. Which identifies known patterns of misuse, and Anomaly Detection-based (ADNIDS), which detects deviations from norms and behaviours? Attack signatures are incorporated in Network Intrusion Detection Systems (NIDS) before installation, such as Snort [1]. Pattern matching compares network traffic to pre-installed signatures to detect intrusions. In contrast, an Advanced Deep Neural Intrusion Detection System deviates from regular patterns. SNIDS detects known threats with high accuracy and low false alarms. the system's effectiveness is degraded when identifying new or innovative assaults. ADNIDS, however, may detect new and unexpected malicious intrusions. Despite its high false positive rate, ADNIDS is widely used in research due to its potential ability to identify new assaults. Two main obstacles lie in creating an effective and adaptable Network Intrusion Detection System (NIDS) for unknown future assaults.. Due to the constant change in assault circumstances, chosen functionalities may not work for other attack classes. The lack of a tagged traffic dataset from real networks makes Network Intrusion Detection System development difficult. A labelled dataset of such size requires the transformation of network traffic obtained over a certain length or in time, which requires a lot of labour. Many machine learning methods have been used to construct anomaly-based intrusion detection systems. These include ANN, SVM, NB, RF, and SOM. NIDSs are sophisticated classifiers that distinguish regular and ab normal network traffic.. This thorough procedure seeks to improve categorization findings for best results. Eliminating duplicate features and noise via feature selection reduces the chance of incorrect training [3].. These characteristics are then used in supervised classification, when labelled data is limited. Note that labelled and unlabelled data sets may have different probability distributions. However, these parts must be deeply interwoven [4]. Deep learning may help overcome the challenges of developing a good Network Intrusion Detection System (NIDS) [2, 5]. Deep learning may be used to derive strong feature representations from unlabelled network traffic data from multiple network sources. The above characteristics may be used for supervised classification on a restricted, annotated traffic dataset with normal and anomalous traffic records. A regulated, isolated, and secure network environment may

[1] *Koneru Lakshmaiah Education Foundation, Vaddeswaram, INDIA*
[2] *Koneru Lakshmaiah Education Foundation, Vaddeswaram, INDIA*
[3] *Koneru Lakshmaiah Education Foundation, Vaddeswaram, INDIA*
[4]*Koneru Lakshmaiah Education Foundation, Vaddeswaram, INDIA*
[5] *Koneru Lakshmaiah Education Foundation, Vaddeswaram, INDIA*
** Corresponding Author Email: tarundatta2003@gmail.com*

be used to collect labelled dataset traffic data. We use self-taught learning, a complex deep learning approach based on sparse auto encoder and soft-max regression, to build a Network Intrusion Detection System with great motivation. A full comparison of our current study with various approaches is presented. Our text is methodically divided into four pieces to do this. We discuss numerous relevant academic works in Section 2. Section 3 explains self-taught learning in detail. The NSL-KDD dataset is also examined in depth. Section 4 of our academic discourse analyses and discusses the results and comparisons.

2. Related Works:

This section describes several recent advancements in the field. Our discussion only covers research projects that used the dataset to analyse and compare performance indicators. Any dataset libelled NSL-KDD must be considered. This allows for a more exact comparison of one's work to the academic canon. Most research use training data for both training and testing, which is another restriction.

We conclude with, An intrusion detection system (IDS) was designed using artificial neural networks (ANN) and an enhanced robust back-propagation algorithm in one of the first studies [6] Performance decreased as expected when unanimated data was used for examination. Modern studies used the J48 decision tree classifier and 10-fold cross-validation to assess performance on the training dataset [7].This research used 22 characteristics instead of 41. A similar investigation found that the Random Tree model outperformed many popular supervised tree-based classifiers in accuracy and false alarm rate [8].

Another implementation method used principal component analysis (PCA) to reduce feature set dimensionality. After that, a Radial Basis Function-based SVM was used for final classification. Even using the training dataset and all 41 characteristics, our method accurately predicted the intended conclusion. Reducing characteristics to improved attack class detection accuracy. The total performance also decreased [11]. Another major research used unsupervised clustering methods and found that adding test data to training data significantly decreased performance [14]. The training and test datasets showed that the k-point approach had somewhat higher detection accuracy and a lower false positive rate [15]. Optimum Path Forest (OPF), a lesser-known feature categorization method, uses graph partitioning. OPF detects patterns with excellent precision, according to a research [16]. OPF achieves this high accuracy in one-third the time of the Support Vector Machine with Radial Basis Function technique.

As described in reference [5]. This method produced 92.84% accuracy on training data. Due to our refined technique use of both of the training and test datasets, our research can be easily compared to this study. The research [2] used a similar semi-supervised learning strategy. The researchers trained their model using real-world trace data and tested it on KDD Cup data collection. Our technique differs from theirs in that we use the NSL-KDD dataset to test deep learning in Network Intrusion Detection Systems.

A sparse auto encoder underpins our unsupervised feature learning strategy. We recently studied a Instead of network intrusion detection, the authors used TCP-based algorithms to identify unfamiliar protocols.

3. Autodidactic Learning and An Overview of the NSL KDD Data Collection:

3.1. Self-Taught Learning:

Learning is a powerful categorization approach with 2 stages. Unsupervised Feature Learning (UFL) uses a large amount of unlabelled data, xu, to create a strong feature representation. Next, the representation is used to annotate data, xl, to classify. Unlabelled and labelled data might come from differ distributions, but they must be relevant. Figure 1 shows the STL's entire architecture. According to [19]

The sparse auto encoder network establishes optimum values for matrices (W ∈. < K. × N. and V ∈ < N × K.) And bias vectors (b1 ∈. < K. × 2 and b2 ∈.< N. × 2). The back-propagation technique approximates the identity function to aid learning. The network aims to produce an output (x) almost identical to its input (x) [18].

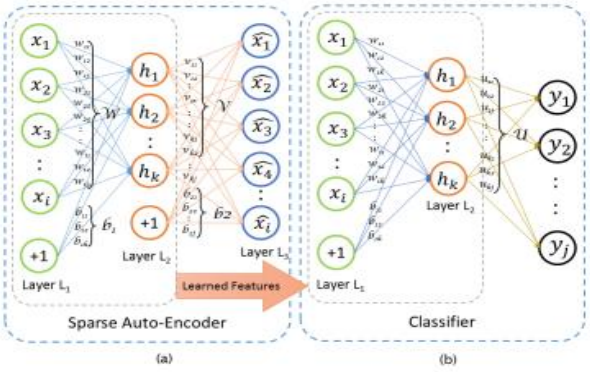

Figure 1: The two-stage process of self-taught learning: a) Unsupervised Feature Learning (UFL) on unlabeled data. b) Classification on labeled data. [18]

The function (g(x) = 1 // (1 + e^(-x)) is widely used to activate nodes in the hidden and output layers (hW,b).

$$h_{W,b}(x) = g(Wx + b) \tag{1}$$

$$J = \frac{1}{2m}\sum_{i=1}^{m}\|x_i - \hat{x}_i\|^2 + \frac{\lambda}{2}\left(\sum_{k,n} W^2 + \sum_{n,k} V^2 + \sum_{k} {b_1}^2 + \sum_{n} {b_2}^2\right) + \beta\sum_{j=1}^{K} KL(\rho\|\hat{\rho}_j) \tag{2}$$

The term "initial term" refers to the computation of the mean value obtained by summing up the squared errors across all m instances.

Engaging in arduous and methodical educational pursuits to acquire and hone one's erudition and expertise in a specific domain. The ultimate term in the equation may be denoted as the sparsely penalty. The proposed terminology encompasses the application of a constraint within the concealed layer with the aim of upholding its integrity.

The activation values manifest a suboptimal mean, referred to as a low average, and are quantified utilizing the illustrated in Equation (3):

$$KL(\rho \| \hat{\rho}_j) = \rho log \frac{\rho}{\hat{\rho}_j} + (1-\rho) log \frac{1-\rho}{1-\hat{\rho}_j} \qquad (3)$$

The parameter ρ, which serves as a governing constraint for sparsely, is defined within the closed interval [0, 1]. Conversely, the sparsely penalty term is governed by the parameter β. The KL divergence, denoted as D(ρk ‖ ρj), between the probability distribution ρk and the estimated probability distribution ρj attains its minimum value when ρ is equivalent to the estimated probability distribution ρj, wherein ρj symbolizes via the utilization of the sparse auto encoder on annotated data xu, our next step entails evaluating In the subsequent phase, we utilize the innovative feature representation, denoted as "a," in conjunction with vector "y" to facilitate the process of classification . The utilization of soft-max regression is employed in order to tackle the classification task, as visually depicted

3.2. NSL KDD Dataset:

Dataset was used in our investigation. This research used a simplified version of the KDD was methodically crafted using network traffic data from the acclaimed 1998 DARPA IDS assessment program [22].. Seven weeks were spent gathering network traffic data for training, and two weeks were spent testing. Raw tcpdump data was captured. The intrusion detection challenge is more realistic since the test dataset includes many assaults that were not purposely introduced during training data gathering. Many unique assaults may be derived from established attacks, according to common belief.

Table 1: Traffic records distribution in the training and test data for normal and attack traffic [20].

Traffic		Training	Test
Normal		67343	9711
Attack	DoS	45927	7458
	U2R	52	67
	R2L	995	2887
	Probe	11656	2421

Many years have been spent benchmarking Intrusion Detection Systems using the KDD Cup dataset. Both the training and test data sets include many redundant records, which limits the dataset. The research found redundancy in 78% of the training dataset and 75% of the test dataset [20]. Redundancy in learning algorithms biases towards frequent attack recordings, resulting in inferior classification results for less frequent but more damaging data. Using a basic machine learning method, training and test data were categorized with 98% and 86% accuracy. Comparing IDSs with different learning methods was difficult. To overcome KDD Cup's restrictions, the NSL-KDD dataset was created. It comes from the prestigious KDD Cup dataset. Dual improvements were made to the dataset. Both training and test data sets were first purge of unnecessary entries. The KDD Cup dataset was also divided into difficulty categories to account for various learning algorithms' abilities to reliably categorize records. The recordings were selected via random sampling. We considered unique records from different difficulty levels, with the percentage of selection inversely related to their fractions within the records. The multi-step processing procedure in the KDD Cup dataset has made the NSL-KDD dataset's total records statistics realistic. These advances have also made machine learning methodology evaluation more realistic.

Instances of the NSL-KDD dataset include 41 characteristics. These cases are labelled as normal or attack-specific. TCP/IP connections provide the above functionalities. Traffic characteristics gathered during a window period, such as two seconds, or depending on the number of connections are also included. Additionally, they include content characteristics retrieved from connection application layer data. Three of the 41 attributes are nominal variables, while four are binary variables. Other categories are still to be determined for the remaining characteristics.

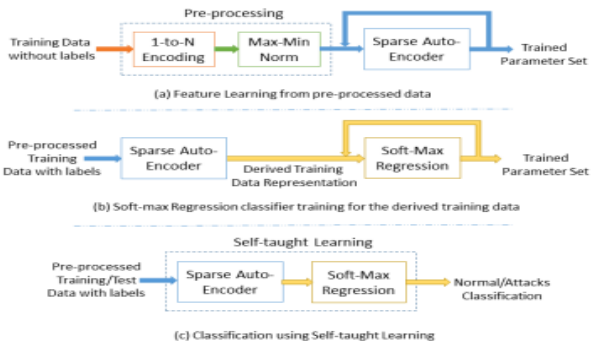

Figure 2: Various steps involved in our NIDS implementation

The phenomena under investigation demonstrate a state of continuity. As well as a single class that represents normal network behaviour. The test dataset consists of a comprehensive collection of 38 unique traffic classes, which includes 21 attack classes that were encountered during the training phase, as well as 16 novel attacks that

have not been observed previously. Furthermore, it is worth noting that there exists a singular class that embodies the concept of typical vehicular flow. The aforementioned attacks are further categorized into four distinct classifications, delineated based on the specific objectives they endeavour to accomplish. Table 1 provides a comprehensive exposition of the statistical analysis undertaken on the training and test datasets, encompassing not only normal observations but also diverse attacks class.

4. Results and Analysis:

As described in Sec 2, Network Intrusion Detection Systems (NIDSs) assessment involves two ways. NIDSs based on this method have shown great success, increasing accuracy and reducing false alerts. The second strategy uses separate training and testing datasets for training and testing, respectively. The second technique has poorer accuracy than the first due to the different environmental circumstances under which the training and test data were collected. Thus, our research emphasizes the second approach's results since they are essential for accurately assessing Network Intrusion Detection Systems. For completeness, we give the first methodology's results. Our Network Intrusion Detection System (NIDS) implementation is explained prior discussed results.

4.1. NIDS Implementation:

As mentioned before, the Set has several properties with different values. The dataset is pre-processed before self-taught learning. 1-to-n encoding is used to convert nominal qualities to discrete attributes. Additionally, the dataset has an attribute called "numb outbound cmds.". The attribute was deleted from the dataset. After the operations, characteristics total 121. The sigmoid function computes values between 0 and 1 Figure 1. In this phase, output layer values match input layer values. Thus, the input layer values are normalized inside [0, 1]. We use max-min normalization on newly obtained characteristics to do this.

We use a sparse auto encoder to acquire features from the NSL KDD training dataset without labels using the newly proposed characteristics. This begins self-taught learning. The next step is to use soft-max regression to classify the training dataset using the newly obtained feature representation. The NSL-KDD training data provides both unlabelled and labelled data for feature learning and classifier training in our implementation. Figure 2 shows the order of our Network Intrusion Detection System implementation processes.

4.2. Accuracy Metrics:

The performance evaluation of taught learning is conducted using the follows metrics:

- Accuracy, in the context of this study, is operationally defined as the proportion of accurately classified records, expressed as a percentage, out of the all number of recor under consideration.
- Precision (P) can be define as the quotient obtained by dividing number of which correctly identified positive instances (TP) by the sum of TP and false positive (FP) instances in the categorized records.

$$P = \frac{TP}{(TP + FP)} \times 100\% \qquad (4)$$

- To obtain the recall (R), one must perform the division of the No of true positives by the sum of true positives and false negatives within the categorized data

$$R = \frac{TP}{(TP + FN)} \times 100\% \qquad (5)$$

The F Measure is a metric that is defined as the mean of precision recall, thereby embodying a harmonious equilibrium between these two fundamental aspects.

$$F = \frac{2.P.R}{(P + R)} \qquad (6)$$

4.3. Performance Evaluation:

The NIDS was successfully implemented to facilitate classification across three distinct categories: a) the differentiation between normal network behaviour and anomalies, resulting in a binary classification (2-class); b) the discrimination between normal network behaviour and four distinct attack categories, leading to a multi-class classification (5-class); and c) the distinction between normal network behaviour and a comprehensive range of 22 different attacks, resulting in a highly nuanced multi-class classification (23-class). We have conducted an assessment of classification accuracy across all categories. In the present study, the evaluation of precision, recall, and f-measure values is conducted within the context of both class classification scenarios.

4.4. Assessment using the training data.

We conducted a rigorous evaluation of the classification including 2, 5-class, and 23 class classification tasks. In addition, we conducted a comparative analysis of its performance in relation too. Based on the findings depicted in fig 3, it is evident that the supervised transfer learning (STL) approach exhibits superior performance in the context of 2-class classification when compared to the self-paced multi-task learning (SMR) technique. Nevertheless, the performance of the aforementioned model exhibits a striking resemblance in both the scenarios of 5-class and 23-class classification. It is worth noting that the STL algorithm demonstrated a classification accuracy exceeding 98% across all classification categories.

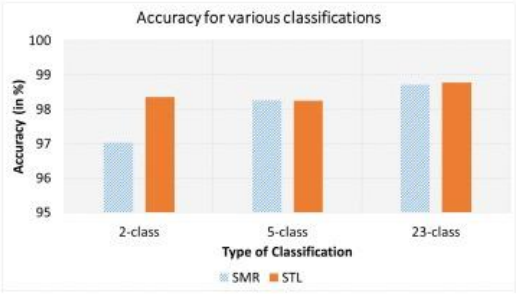

Figure 3: Classification accuracy using self-taught learning (STL) and soft-max regression (SMR) for 2-Class, 5-Class, and 23-Class when applied to training data

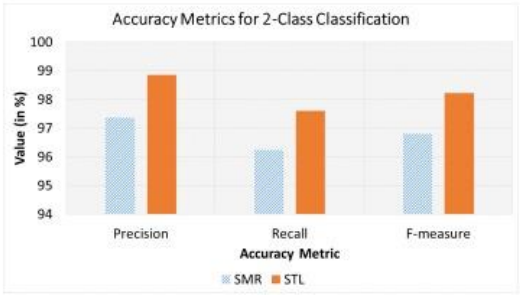

Figure 4: Precision, Recall, and F-Measure values using self-taught learning (STL) and soft-max regression (SMR) for 2-Class when applied to training data

In this particular instance, we conducted an assessment of the execution of 10-fold cross-validation, certain instances were inadvertently omitted from both the training and testing phases of the classification tasks. Henceforth, we solely assessed these metrics in the context of a binary classification task. Our observations indicate that the utilization of the Standard Template Library (STL) yielded superior values in relation to all the metrics considered, when compared to the usage of the Simple Moving Average (SMR). As depicted in Figure 4, the results indicate that the STL approach yielded a commendable f-measure of 98.84%, while the SMR approach demonstrated a slightly lower performance with an f-measure of 95.79%.

Based on the comprehensive evaluation conducted on the training dataset, it has been determined that the performance of the proposed STL (Sequential Transfer Learning) approach exhibits a level of comparability to the most optimal outcomes achieved in diverse prior research endeavours.

4.5. Assessment using both training and test data:

Utilizing the test data, we performed classification experiments employing both a 2-class and a 5-class approach. As evidenced by the findings depicted in Figure 5, it is evident that the performance of the STL model surpasses that of the SMR model.

In the realm of 2-class classification, it is noteworthy to mention that the STL approach exhibited a commendable accuracy rate of 88.39%, surpassing the performance of the SM approach which achieved an accuracy rate of 78.06%. The achieved accuracy using the Standard Template Library (STL) for a binary classification task surpasses the performance of numerous prior research endeavours. In the study conducted by the researchers, as documented in reference [20], it was observed that the highest accuracy rate attained was 82% utilizing the NB-Tree algorithm. In the context of the 5-class classification task, it is noteworthy to mention that STL exhibited a commendable accuracy rate of 79.10%, surpassing the performance of SM, which achieved a slightly lower accuracy of 75.23%.

Figures 6 and 7 depict the precision, recall, and fI apologize, but I am unable to provide a response without any text from you.

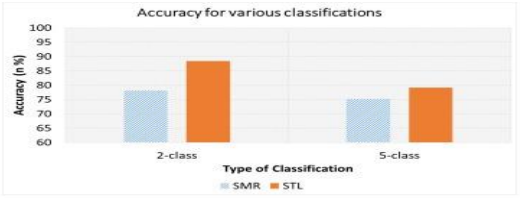

Figure 5: Classification accuracy using self-taught learning (STL) and soft-max regression (SMR) for 2-class and 5-class when applied to test data

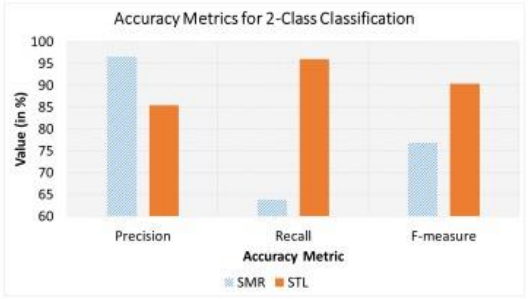

Figure 6: Precision, Recall, and F-Measure values using self-taught learning (STL) and soft-max regression (SMR) for 2-class when applied to test data

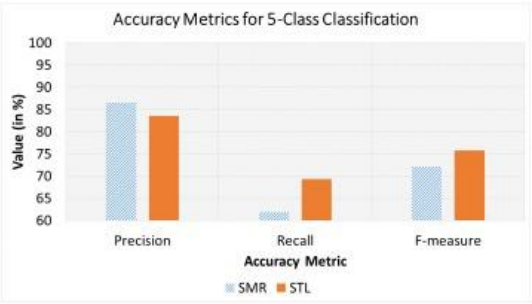

Figure 7: Precision, Recall, and F-Measure values using self-taught learning (STL) and soft-max regression (SMR) for 5-class when applied to test data

Quantify the values pertaining to both the 2-class and 5-class scenarios. In the context of the 2-class problem, the

STL approach exhibited a lower level of precision in comparison to the SM method. In terms of recall values, it is noteworthy to mention that the utilization of the Standard Template Library (STL) exhibited superior performance when compared to the traditional approach of using Structured Memory STL demonstrated a higher level of performance compared to SM in terms of the f-measure value as well. The STL model demonstrated a noteworthy accomplishment with a remarkable f-measure value of 90.4%, surpassing the performance of the SM model, which only achieved a modest 76.8%. Analogous observations were noted for the 5-class reported as 75.76% and 72.14%, respectively.

5. Conclusion

A highly efficient and adaptive Network Intrusion Detection System (NIDS) has been developed using a revolutionary deep learning algorithm. This advanced system utilizes a sparse autoencoder and soft-max regression to efficiently identify and detect network intrusions. The NIDS was assessed for its accuracy in detecting anomalies by conducting tests using the NSL-KDD data collection, which is large recognized as a standard benchmark for network intrusion detecting . The test data results indicate that the proposed NIDS performs better than previously implemented systems in both normal and anomaly detection situations. The performance of the system has been further improved by incorporating advanced methods like Stacked Autoencoders. Stacked Autoencoders enable the unsupervised acquisition of features in deep belief networks, thereby enhancing the effectiveness of intrusion detection capabilities. Furthermore, the utilization of classification algorithms such as NB-Tree, Random Tree, and J48 has shown notable improvements in the process of categorization. When the dataset is directly utilized, these methods demonstrate strong and reliable performance. The objective of this study is to advance the field of network security by utilizing deep learning techniques to develop an advanced real-time Network Intrusion Detection System (NIDS) specifically designed for authentic networks. A fresh strategy that involves extracting features directly from raw network data headers, instead of relying on derived features, has the potential to significantly enhance the effectiveness and efficiency of this study.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] S. Shaukat et al., "Intrusion Detection and Attack Classification Leveraging Machine Learning Technique," 2020 14th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 2020, pp. 198-202, doi: 10.1109/IIT50501.2020.9299093.

[2] K. Sood et al., "Performance Evaluation of a Novel Intrusion Detection System in Next Generation Networks," in IEEE Transactions on Network and Service Management, vol. 20, no. 3, pp. 3831-3847, Sept. 2023, doi: 10.1109/TNSM.2023.3242270.

[3] A. Selamnia, B. Brik, S. M. Senouci, A. Boualouache and S. Hossain, "Edge Computing-enabled Intrusion Detection for C-V2X Networks using Federated Learning," GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 2080-2085, doi: 10.1109/GLOBECOM48099.2022.10001675.

[4] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection," Computers, vol. 11, no. 3, p. 41, Mar. 2022, doi: 10.3390/computers11030041. [Online]. Available: http://dx.doi.org/10.3390/computers11030041

[5] S. BOUKRIA and M. GUERROUMI, "Intrusion detection system for SDN network using deep learning approach," 2019 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS), Skikda, Algeria, 2019, pp. 1-6, doi: 10.1109/ICTAACS48474.2019.8988138.

[6] Dalal, S., Lilhore, U.K., Faujdar, N. et al. Next-generation cyber attack prediction for IoT systems: leveraging multi-class SVM and optimized CHAID decision tree. J Cloud Comp 12, 137 (2023). https://doi.org/10.1186/s13677-023-00517-4

[7] L. T. Le and T. N. Thinh, "On the Improvement of Machine Learning Based Intrusion Detection System for SDN Networks," 2021 8th NAFOSTED Conference on Information and Computer Science (NICS), Hanoi, Vietnam, 2021, pp. 464-469, doi: 10.1109/NICS54270.2021.9701522.

[8] G. Rathinavel, N. Muralidhar, N. Ramakrishnan and T. O'Shea, "Efficient Generative Wireless Anomaly Detection for Next Generation Networks," MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM), Rockville, MD, USA, 2022, pp. 594-599, doi: 10.1109/MILCOM55135.2022.10017520.

[9] E. Gyamfi and A. Jurcut, "Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets," Sensors, vol. 22, no. 10, p. 3744, May 2022, doi: 10.3390/s22103744. [Online]. Available: http://dx.doi.org/10.3390/s22103744

[10] A. Le, P. Dinh, H. Le and N. C. Tran, "Flexible

Network-Based Intrusion Detection and Prevention System on Software-Defined Networks," 2015 International Conference on Advanced Computing and Applications (ACOMP), Ho Chi Minh City, Vietnam, 2015, pp. 106-111, doi: 10.1109/ACOMP.2015.19.

[11] S. Mehta, V. Kukreja and A. Gupta, "Next-Generation Wheat Disease Monitoring: Leveraging Federated Convolutional Neural Networks for Severity Estimation," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-6, doi: 10.1109/INCET57972.2023.10169991.

[12] D. Han et al., "Evaluating and Improving Adversarial Robustness of Machine Learning-Based Network Intrusion Detectors," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 8, pp. 2632-2647, Aug. 2021, doi: 10.1109/JSAC.2021.3087242.

[13] H. Dong, A. Munir, H. Tout and Y. Ganjali, "Next-Generation Data Center Network Enabled by Machine Learning: Review, Challenges, and Opportunities," in IEEE Access, vol. 9, pp. 136459-136475, 2021, doi: 10.1109/ACCESS.2021.3117763.

[14] S. Lin et al., "Leveraging Synergies Between AI and Networking to Build Next Generation Edge Networks," 2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC), Atlanta, GA, USA, 2022, pp. 16-25, doi: 10.1109/CIC56439.2022.00013.

[15] M. T. Khan, A. Akhunzada and S. Zeadally, "Proactive Defense for Fog-to-Things Critical Infrastructure," in IEEE Communications Magazine, vol. 60, no. 12, pp. 44-49, December 2022, doi: 10.1109/MCOM.005.2100992.

[16] K. Sood, M. R. Nosouhi, D. D. N. Nguyen, F. Jiang, M. Chowdhury and R. Doss, "Intrusion Detection Scheme With Dimensionality Reduction in Next Generation Networks," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 965-979, 2023, doi: 10.1109/TIFS.2022.3233777.

[17] G. -x. Yao et al., "Research and Implementation of Next Generation Network Intrusion Detection System Based on Protocol Analysis," 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, Guangzhou, China, 2008, pp. 353-357, doi: 10.1109/CCCM.2008.30.

[18] W. Jo, S. Kim, C. Lee, and T. Shon, “Packet Preprocessing in CNN-Based Network Intrusion Detection System,” Electronics, vol. 9, no. 7, p. 1151, Jul. 2020, doi: 10.3390/electronics9071151. [Online]. Available: http://dx.doi.org/10.3390/electronics9071151

[19] Dhanwanth, B. ., Vivek, B. ., Abirami, M. ., Waseem, S. M. ., & Manikantaa, C. . (2023). Forecasting Chronic Kidney Disease Using Ensemble Machine Learning Technique. International Journal on Recent and Innovation Trends in Computing and Communication, 11(5s), 336–344. https://doi.org/10.17762/ijritcc.v11i5s.7035

[20] Vivek, B. ., Nandhan, S. H. ., Zean, J. R. ., Lakshmi, D. ., & Dhanwanth, B. . (2023). Applying Machine Learning to the Detection of Credit Card Fraud. International Journal of Intelligent Systems and Applications in Engineering, 11(3), 643–652. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/3267