Network Anomaly Detection System using Deep Learning with Feature Selection Through PSO

Rimjhim Rathore, Neeraj Shrivastava



Abstract: The more computer systems that communicate and cooperate, the more crucial it is to make our lives simpler. At the same time, it highlights faults that people are unable to correct. Due to faults, cybersecurity procedures are required to ensure the secure communication of data. Secure communication requires both the installation of security measures and the ongoing development of new security measures to address evolving security concerns. In this study, it is suggested that network intrusion detection systems be able to adapt and be resilient. This can be achieved by utilising deep learning architectures. Deep learning is employed in this article to identify and categorise network attacks. Some tools can help intrusion detection systems become more flexible and learn to recognise new or zero-day network behaviour features, which can help them identify malicious activity and make it less likely for bad actors to gain access to your network. The model's efficacy was tested using the KDD dataset, which combines real-world network traffic with fake attack operations.

Keyword: Intrusion Detection System, KDD, Deep Learning, Accuracy, Cyber-Security.

I. INTRODUCTION

The Principal Component Analysis (PCA) method is used to examine changes in feature variance across intrusion detection [1] data streams to determine if data and concepts have changed (PCA) [2]. As an added bonus, we demonstrate how to use an online technique to find outliers [3] that are distinct from both historical and temporally close data [4]. This is addressed by using an online deep neural network [5] that changes the hidden layer [6] size through Hedge weighting to mitigate the problem [7]. This enables the model to adjust to new information [8] as it comes in. At the other end of the spectrum from the static deep neural network model [9] often used for intrusion detection [10], our technique retains performance on both training and testing data, which is essential since it simplifies the process of troubleshooting. [11]. On diverse devices, we want to investigate how well pre-trained models perform to determine if deep learning-based intrusion detection can be used on embedded devices with restricted resources [12].

Manuscript received on 19 April 2022 | Revised Manuscript received on 27 March 2023 | Manuscript Accepted on 15 April 2023 | Manuscript published on 30 April 2023. *Correspondence Author(s)

Rimjhim Rathore*, Department of Computer Science and Engineering, IES, IPS Academy Indore, Indore (M.P.), India. E-mail: <u>rimjhimrathore1@gmail.com</u>, ORCID ID: <u>https://orcid.org/0000-0003-</u> 3752-8725

Dr. Neeraj Shrivastava, Head of Department, Department of Computer Science and Engineering, IES, IPS Academy Indore, Indore (M.P.), India. E-mail: <u>neeraj0209@gmail.com</u>

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an <u>open access</u> article under the CC-BY-NC-ND license <u>http://creativecommons.org/licenses/by-nc-nd/4.0/</u>

Retrieval Number: 100.1/ijese.F25310510622 DOI: <u>10.35940/ijese.F2531.0411523</u> Journal Website: <u>www.ijese.org</u> In total, four deep learning models will be installed on every device, each trained on a separate well-known intrusion detection dataset. We will measure precision, recall, and prediction rate, which is the time it takes to predict one sample per second. Precision is defined as the accuracy of a prediction. A variety of datasets will be utilised to examine how the models respond to various assault patterns.

II. LITERATURE WORK

To determine if data and ideas have evolved, we employ Principal Component Analysis to examine changes in the variance of features across intrusion detection data streams. We also demonstrate how to utilise an online method to identify outliers that differ from both historical and temporally nearby data, and provide guidance on how to do so. To solve this problem, we built an online deep neural network that changes the size of the hidden layer through Hedge weighting. This allows the model to change as new information becomes available. Unlike the static deep neural network model used for intrusion detection, our method works well on both training and testing data. This is important because it makes troubleshooting easier. [13]

We aim to evaluate the performance of pre-trained models on various device types to determine if deep learning-based intrusion detection can be effectively applied to embedded devices with limited space and resources. As part of the project, each device will be equipped with four deep learning models. Each model will be trained on a different well-known intrusion detection dataset. It will examine precision, F1 score, recall, and prediction rate, which measures how long it takes to predict each sample per second. All of these things will be looked at. A variety of datasets will be used to assess how the models respond to other types of attacks. [14]. FLUIDS is a federated learning approach for unsupervised Intrusion Detection Systems. FLUIDS transforms intrusion detection into semi-supervised learning, combining supervised and unsupervised learning. Incorporating federated and semi-supervised learning improves user privacy, training and inference efficiency, outcomes, and cost. [15] The researchers claim to have developed an IDS model that can detect several threats simultaneously. Multi-Task Learning (MTL). To do so, we aggregated samples from the UNSW-NB15 and CICIDS2017 datasets into a single feature vector. Both the training and testing sets must include a danger. During testing, the proposed method proved to be the most effective. [16] Ensemble learning improves intrusion detection by combining many individuals who aren't particularly effective at what they do.

Published By: Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) © Copyright: All rights reserved.



1

Deep learning methodologies are becoming increasingly popular as approaches to extract the most value from large datasets and real-time applications.

Techniques like Transfer Learning (TL) work well when data is scarce. Use these approaches to spot new risks. This document shows the latest progress on intrusion detection. The study may educate readers on how the research began, where it is today, and where it may go in the future. [17]

This research employs deep transfer learning to build a reliable IDS model. It beats many existing approaches. Effective attribute selection, a dependable deep transfer learning-based model, and a mechanism for testing using real-world data are among the most significant distinctive contributions. A comprehensive experimental performance assessment was conducted. Longevity, efficiency, and better outcomes than other models demonstrate that the recommended model is likely to be dependable, having undergone extensive study and performance testing. [18]

This study describes a deep learning-based hybrid intrusion detection system. It can quickly detect network risks and intrusions. This method uses RCNN and GBR (Gradient Boost Regression) to detect network incursions on Kaggle's NIDS dataset (Version 10, 2017). According to previous research, the suggested approach outperforms earlier algorithms. Previous research has shown that the recommended strategy is more accurate and faster than alternative methods. [19]

Study: This one examined intrusion detection systems and how machine and deep learning protect data from malicious actors. Builds an operational intrusion detection system using the latest machine learning and deep learning technologies. This operational system examines various network implementations, applications, algorithms, and learning approaches. [20]

Unique use of unlabeled and labelled data in this study. Use unlabeled local or private data to train an AutoEncoder (AE) on the most significant and least complex characteristics of each device. A cloud server then uses Federated Learning to integrate all of these models into a global AE (FL). Finally, the cloud server builds an intelligent, supervised neural network by adding fully connected layers (FCN) to the worldwide encoder, the initial part of the global autoencoder (AE). In two real-world datasets, our approach (a) ensured no private data was exposed, (b) accurately recognised attacks, (c) functioned even when there was minimal marked data, and (d) sent quickly. [21]

An ensemble of Deep Learning (DL) Intrusion Detection Systems finds DDoS attack traffic in SDNs (IDS). We recommend mixing models from three distinct kinds of neural networks: convolutional, deep, and RNN. Train a model using the Canadian Institute for Cybersecurity's Intrusion Detection System (CIC-IDS2017). To train the model, we employed published feature selection algorithms. The findings show that our proposed ensemble deep learning model outperforms the ensemble CNN, ensemble RNN, and ensemble voting models. [22]. Anomaly-based Intrusion Detection System in this research. This is how we made it (IDS). The primary purpose of this solution is to develop a system that can detect unauthorised activity both within and outside the system. Many models have been tested to find one that matches the system and is precise enough. An XGBoost classifier, a Logistic Regressor, a Random Forest classifier, and a Multi-Layer Perceptron classifier (MLP) were all explored. Additionally, the correctness of the models was evaluated, and their performance was compared. In this scenario, the Random Forest Classifier excelled. It was 99.8% accurate with a macro average F1-Score of 0.98. [23]

III. PROPOSED WORK

Instead of a fully linked feed-forward neural network, the suggested deep learning model uses a CNN with a regularised multi-layer perceptron (FNN). CNN, unlike FNN, does not employ multiplication or the dot product as a math operation. Convolution is used instead. Custom hyperparameters are utilised in the convolution process, such as the filter's size, the number of filters, and the number of steps required to create the output matrix. We added padding to the input to account for the fact that the size of the tensors decreases as the input passes through additional convolutional layers. It's utilised between each convolutional layer to reduce or increase the size of the sample feature dimensions. Finally, the classification output layer is described, followed by a fully linked layer with regularisation. UNSW-NB15, which is a good depiction of real-world network traffic and exhibits the most prevalent vulnerabilities and exposures, will be the dataset we utilise to test our model, as illustrated. Many different models have looked at the data set, but the outcomes have been less than optimal. There are still improvements that can be made to the models. Even though the raw data contains almost two million simulations, the architects do not utilise it. Describe nine attack families' imputed datasets for training and testing. Table I lists the many sorts of attacks, along with a brief explanation of each.

Table 1. Attacks in detail



Figure 1. Architecture of our work

Figure 1 illustrates a two-step process, first involving feature selection and then a classifier. For feature selection, we apply

PSO, and for classification, we use a deep neural network.

Published By: Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) © Copyright: All rights reserved.





conv1d_input: InputLayer
conv1d: Conv1D
conv1d_1: Conv1D
max_pooling1d: Maxpooling1D
dropout: Dropout
conv1d_2: Conv1D
Conv1d_3: Conv1D
max_pooling1d_1: Maxpooling1D
dropout_1: Dropout
conv1d_4: Conv1D
conv1d_5: Conv1D
max_pooling1d_2: Maxpooling1D
dropout_2: Dropout
flatten: Conv1D
dense: Dense
dropout_3: Maxpooling1D
Dnese_1: Dense

Figure 2. Proposed classification steps.

A dropout between hidden layers and a dropout between dense layers is shown in <u>Figure 2</u>. This is a way to stop overfitting. ReLU is used as a nonlinear activation function in the output layer, and then Softmax is employed to enhance its performance further. If you have 10 classes, 56K people are in the top class, and just 130 are in the bottom class. Under-sampled classes degrade the model's performance, indicating that bootstrapping is crucial for improving model performance. To avoid redundancy and duplication, and to ensure a fair comparison with other models, we utilised the original datasets. We considered pooling the datasets and splitting them 70-30 between training and testing, but ultimately decided against doing so. In this text, the second option is referred to as "user-defined datasets," which is its official name.

IV. IMPLEMENTATION AND RESULT

4.1. Setup configuration

Intel Core i3 Processor (10th Gen), 8 GB DDR4 RAM. 64bit Windows 10 Operating System. 512 GB SSD. 39.62 cm (15.6 inch) Display. Microsoft Office Home and Student 2019, HP Documentation, HP BIOS Recovery, HP Smart, HP Support Assistant, Dropbox. Python libraries like numpy, pandas, TensorFlow, Keras, sklearn, and matplotlib.

4.2 Dataset

We utilised the NSL-KDD dataset for our study. The dataset is KDD Cup 99 [24]. The KDD Cup dataset was created from DARPA IDS evaluation data from 1998. Regular network traffic includes DoS, probing, user-to-root (U2R), and rootto-local communication (R2L). Raw TCPdump network traffic was collected for seven weeks for training and then for an additional two weeks for testing. There are several attacks in the test data that weren't in the training data. Most new attacks are assumed to be based on old ones. The test and training data generated five million and two million TCP/IP connection records.

It has long been used for NIDS testing. The KDD Cup dataset has been heavily utilised. One of the dataset's drawbacks is that the training and test sets include many of the same items. The training and test datasets share almost 78% of records. Consequently, learning algorithms are biased toward frequent attacks, resulting in poor outcomes for less common but more harmful records. The training and test data were correctly classified with 98% and 86% accuracy using basic machine learning. Comparing IDSs with various learning methods, NSL-KDD was offered as a workaround for the KDD Cup dataset's restrictions. This dataset was created using the KDD Cup. It improved the prior dataset in two ways. First, it checked for duplicate items in the training and test data. Second, it grouped all KDD Cup recordings into difficulty categories based on the number of learning algorithms that correctly classified them. It also picked the recordings at random from various degrees of difficulty, with a percentage inversely related to the number of records. The NSL-KDD dataset contains a substantial number of records, as the KDD Cup dataset was processed in stages. These enhancements also simplify the comparison of machine learning algorithms.

Table 2. Dataset details

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





Figure 3: Result of DNN Model with 3, 4 and 5 Hidden Layers



Figure 4: Result of DNN Model with 1 and 2 Hidden Layers

Published By: Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) © Copyright: All rights reserved.



Retrieval Number: 100.1/ijese.F25310510622 DOI: <u>10.35940/ijese.F2531.0411523</u> Journal Website: <u>www.ijese.org</u>

Network Anomaly Detection System using Deep Learning with Feature Selection Through PSO



Figure 5: Result of Decision Tree



Figure 6: Result of Linear Regression and Naive Bayes Algorithm

4.4 Result

This section discusses how the proposed system operates, its differences from other systems, and its internal functioning. Need to determine some key factors to consider for the analysis. The values for various parameters are presented in this chapter, allowing you to view their specific details.

Our proposed method was evaluated in below parameters:

- Recall
- Precision
- Accuracy
- F1-Score

Table 3. Evaluate Metric with Contingency Table

		Prediction	
		Predicted Negative	Predicted Positive
	Actually Negative	True Negative (TN)	False Positive (FP)
Reality	Actually Positive	False Negative (FN)	True Positive (TP)

$$recall = \frac{tp}{tp+fn}.....(1)$$

$$precision = \frac{tp}{tp+fp}....(2)$$

$$accuracy = \frac{tp+tn}{tp+tn+fp+fn}....(3)$$

4.5 Experimental Results

The NSL-KDD dataset is the most effective way to explore new methods for improving IDS in the world today. For both training and testing, we have used a 10% dataset from the

Retrieval Number: 100.1/ijese.F25310510622 DOI: <u>10.35940/ijese.F2531.0411523</u> Journal Website: <u>www.ijese.org</u> NSL-KDD, which has 3,11,029 records. We have employed two different methods to categorise the ordinary and intrusive elements in the informative index. <u>Table 3</u> presents the results of tests on six different methods.

Table 4.	Result	Existing
----------	--------	----------

Algorithm		Accuracy	Precision	Recall	F-Score
	Linear Regression	84.8	98.9	82.4	89.7
	Decision Tree	93	99.9	91.4	95.4
	Naive Bayes	92.9	98.9	92.3	95.5
	Random Forest	92.7	99.9	91	95.2
	Adaboost	92.5	99.5	91.1	95.1
	DNN_1	92.9	99.8	91.4	95.4
	DNN_2	93.1	99.9	91.5	95.5
	DNN_3	93.1	99.8	91.6	95.5
	DNN_4	93.1	99.8	91.6	95.5
	DNN 5	93.1	99.9	91.6	95.5

4.5.1 Accuracy

The accuracy calculation, which is a proportion of the correctly anticipated categories to the entire Test Dataset, includes both True Positives and True Negatives. <u>Equation 3</u> is used to figure out the precision.



Figure 7. Accuracy of the existing and our results

In Figure 7, the accuracy values for all methods, both existing and proposed, are calculated. Additionally, a figure illustrates the results. When compared to supervised classification algorithms, we discovered that the proposed technique had a greater actual positive rate.

4.5.2 Precision

Precision is a metric for determining how many positive class expectations are genuinely related to the positive class in question. The precision of the measurement is determined using Equation 2.



Figure 8: Precision of existing and our results

In Figure 8, the accuracy value for all algorithms, both existing and proposed, is calculated.

Published By: Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) © Copyright: All rights reserved.





Additionally, a figure illustrates the results. When compared to supervised classification algorithms, we discovered that the proposed technique had a greater actual positive rate.

4.5.3 Recall

The number of positive class expectations created by each and every positive model in the dataset is measured by recall. With the aid of equation 1, the recall is calculated. In Figure 9, the recall value for all algorithms is calculated. The terms "existing work" and "proposed work" are not interchangeable. Additionally, a figure illustrates the results. When compared to supervised classification algorithms, we discovered that the proposed technique had a greater actual positive rate.



Figure 9. Recall of existing and our results

4.5.4 F1-Score

The F1 measure, primarily used in binary classification, is employed to assess the validity of the test results. The accuracy and recall are taken into account while calculating the F1 metric. In a particular circumstance, the F1 score shows the balance between precision and recall.





Figure 10. F1_Score of existing and our result

V. CONCLUSION

This section outlines our approach to writing an essay. The following are key points to consider for the recommended arrangement project. Because the system's security was compromised due to the interruption of acknowledgement learning, many solutions have been proposed. This thesis's primary goal is to categorise system traffic data as good or poor. Particle Swarm Optimization (PSO) is used to optimise system information production. Then, using taught learning, a separate Deep Neural Network (DNN) is created to form a Network Intrusion Detection System (NIDS). The NSL-KDD dataset was used to construct deep neural system models that outperformed the previous KDD Cup 2009 intrusion detection datasets. Some users of NSL-KDD datasets indicate

that deep neural networks with molecular swarm augmentation are pretty accurate and can discover IoT.

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval or consent to participate, as it presents evidence that is not subject to interpretation.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	All authors have equal participation in this article.

REFERENCE

- A. Takeda and D. Nagasawa, "A Simple Deep Learning Approach for Intrusion Detection System," 2021 Thirteenth International Conference on Mobile Computing and Ubiquitous Network (ICMU), 2021, pp. 1-2, doi: 10.23919/ICMU50196.2021.9638850. [CrossRef]
- A. Das and S. G. Balakrishnan, "A Comparative Analysis of Deep Learning Approaches in Intrusion Detection System," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2021, pp. 555-562, doi: 10.1109/RTEICT52294.2021.9573685. [CrossRef]
- S. Newaz, H. M. Imran and X. Liu, "Detection of Malware Using Deep Learning," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), 2021, pp. 1-4, doi: 10.1109/GUCON50781.2021.9573991. [CrossRef]
- K. Yadav, B. B. Gupta, C. -H. Hsu and K. T. Chui, "Unsupervised Federated Learning based IoT Intrusion Detection," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021, pp. 298-301, doi: 10.1109/GCCE53005.2021.9621784. [CrossRef], [PMid]
- H. C. Altunay, Z. Albayrak, A. N. Özalp and M. Çakmak, "Analysis of Anomaly Detection Approaches Performed Through Deep Learning Methods in SCADA Systems," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021, pp. 1-6, doi: 10.1109/HORA52670.2021.9461273. [CrossRef]
- O. A. Wahab, "Intrusion Detection in the IoT under Data and Concept Drifts: Online Deep Learning Approach," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2022.3167005. [CrossRef]
- J.-R. Jiang and C.-L. Li, "Binary- and Multi-class Network Intrusion Detection with Adaptive Synthetic Sampling and Deep Learning," 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), 2021, pp. 1-2, doi: 10.1109/ICCE-TW52618.2021.9603206. [CrossRef]
- S. Varshney, Shikha, S. Singhi and B. Sharma, "Intelligent Intrusion Detection System Using Deep Learning Models," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 787-793, doi: 10.1109/ICOEI51242.2021.9452880. [CrossRef]
- I. Ullah and Q. H. Mahmoud, "Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks," in IEEE Access, vol. 9, pp. 103906-103926, 2021, doi: 10.1109/ACCESS.2021.3094024. [CrossRef]
- M. Zeeshan et al., "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," in IEEE Access, vol. 10, pp. 2269-2283, 2022, doi: 10.1109/ACCESS.2021.3137201. [CrossRef]
- S. Thirimanne, L. Jayawardana, P. Liyanaarachchi and L. Yasakethu, "Comparative Algorithm Analysis for Machine Learning Based Intrusion Detection System," 2021 10th International Conference on Information and Automation for Sustainability (ICIAfS), 2021, pp. 191-



Retrieval Number: 100.1/ijese.F25310510622 DOI: <u>10.35940/ijese.F2531.0411523</u> Journal Website: <u>www.ijese.org</u>

5

Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) © Copyright: All rights reserved.

Published By:

Network Anomaly Detection System using Deep Learning with Feature Selection Through PSO

196, doi: 10.1109/ICIAfS52090.2021.9605814. [CrossRef]

- S. Singh, S. V. Fernandes, V. Padmanabha and P. Rubini, "MCIDS-Multi Classifier Intrusion Detection system for IoT Cyber Attack using Deep Learning algorithm," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021, pp. 354-360, doi: 10.1109/ICICV50876.2021.9388579. [CrossRef]
- O. A. Wahab, "Intrusion Detection in the IoT under Data and Concept Drifts: Online Deep Learning Approach," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2022.3167005. [CrossRef]
- J. Hunter, B. Huber and F. Kandah, "Towards feasibility of Deep-Learning based Intrusion Detection System for IoT Embedded Devices," 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 2022, pp. 947-948, doi: 10.1109/CCNC49033.2022.9700706. [CrossRef]
- O. Aouedi, K. Piamrat, G. Muller and K. Singh, "FLUIDS: Federated Learning with semi-supervised approach for Intrusion Detection System," 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 2022, pp. 523-524, doi: 10.1109/CCNC49033.2022.9700632. [CrossRef]
- S. A. Albelwi, "An Intrusion Detection System for Identifying Simultaneous Attacks using Multi-Task Learning and Deep Learning," 2022 2nd International Conference on Computing and Information Technology (ICCIT), 2022, pp. 349-353, doi: 10.1109/ICCIT52419.2022.9711630. [CrossRef]
- V. Varanasi and S. Razia, "Network Intrusion Detection using Machine Learning, Deep Learning - A Review," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022, pp. 1618-1624, doi: 10.1109/ICSSIT53264.2022.9716469. [CrossRef]
- S. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed and I. Rafiqul, "Dependable Intrusion Detection System for IoT: A Deep Transfer Learning-based Approach," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2022.3164770. [CrossRef]
- P. Aravamudhan and T. Kanimozhi, "A Robust Adaptive Intrusion Detection System using Hybrid Deep Learning," 2022 International Conference on Computer Communication and Informatics (ICCCI), 2022, pp. 1-6, doi: 10.1109/ICCCI54379.2022.9741046. [CrossRef]
- A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi and R. Ahmad, "Machine Learning and Deep Learning Approaches for CyberSecurity: A Review," in IEEE Access, vol. 10, pp. 19572-19585, 2022, doi: 10.1109/ACCESS.2022.3151248. [CrossRef]
- O. Aouedi, K. Piamrat, G. Muller and K. Singh, "Federated Semi-Supervised Learning for Attack Detection in Industrial Internet of Things," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2022.3156642. [CrossRef]
- U. Mbasuva and G.-A. L. Zodi, "Designing Ensemble Deep Learning Intrusion Detection System for DDoS attacks in Software Defined Networks," 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), 2022, pp. 1-8, doi: 10.1109/IMCOM53663.2022.9721785. [CrossRef]
- S. Kejriwal, D. Patadia, S. Dagli and P. Tawde, "Machine Learning Based Intrusion Detection," 2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAECC), 2022, pp. 1-5, doi: 10.1109/ICAECC54045.2022.9716648.
 [CrossRef], [PMid]
- Jing, Dishan, and Hai-Bao Chen. (2019) "SVM Based Network Intrusion Detection for the UNSW-NB15 Dataset." In 2019 IEEE 13th International Conference on ASIC (ASICON), 1-4. IEEE. [CrossRef], [PMCid]



AUTHOR PROFILE

Rimjhim Rathore is a student at Indore Professional Studies Academy (IPS Academy), Institute of Engineering and Science, Department of Computer Science & Engineering, Indore (M.P). She is currently pursuing her post-graduation. She received her B.E. in Computer Science & Engineering from Jabalpur Engineering College in 2020. Her research interest includes networking and deep learning.



Dr. Neeraj Shrivastava is an HOD and Associate Professor in the IPS Academy, Institute of Engineering and Science, Department of Computer Science & Engineering, Indore. He completed his PhD at the Maulana Azad National Institute of Technology, Bhopal, in 2021. He received the M. Tech in Computer Science & Engineering from MANIT Bhopal in 2009 and B. Tech (CSE) from

RGPV Bhopal in 2006. He has published more than 35 research papers in various international journals and conferences. He is also a reviewer for many Scopus/SCI journals. His research interests include Image Processing, Ad-Hoc Networks, and Algorithms.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)/ journal and/or the editor(s). The Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Retrieval Number: 100.1/ijese.F25310510622 DOI: <u>10.35940/ijese.F2531.0411523</u> Journal Website: <u>www.ijese.org</u> Published By: Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) © Copyright: All rights reserved.