Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Securing IoT with Advanced Machine Learning and deep leaning Based Intrusion Detection Systems

Mohit Jain*, Dr Saurabh Mandloi**
Phd Scholar HOD Computer Science & Technology, SGU***
Hodcse@samglobaluniversity.ac.in
bmctmohitcs@gmail.com Saurabhm.research@gmail.com***

ARTICLE INFO

Received: 05 Oct 2024

Accepted: 25 Nov 2024

ABSTRACT

Intrusion detection systems (IDS) are essential for protecting networks from cyberattacks in the connected digital world of today. Using the well-known KDD CUP 99 dataset, this paper investigates the usage of Support Vector Machine (SVM), Random Forest methods, and VGG16 AND VGG19 for intrusion detection. The dataset, which includes a variety of normal and attack traffic patterns, offers a thorough benchmark for assessing machine learning-based IDS approaches. While Random Forest is used for its ensemble learning technique, with vgg16 and vgg19 increasing accuracy, the SVM model is used for its capacity to identify the best hyper planes for classification. Because of its interpretability and simplicity, the machine learning and deep learning models are also used to categorize network traffic. To improve model performance, data pre-processing techniques include categorical encoding, dimensionality reduction, and feature normalization. The outcomes of these tests show that while both models attain high detection accuracy, vgg19 performs better than the others in terms of precision and recall because of its enhanced capacity to manage intricate datasets and minimize overfitting. The results highlight how deep learning methods, especially vgg19, have the potential to improve the identification of known and new attack pathways.

Keywords: Intrusion Detection System (IDS), Support Vector Machine (SVM), Random Forest, KDD CUP 99

I INTRODUCTION

The Internet has experienced a significant increase in development and utilization over the past few decades. The Internet was utilized extensively by 82% of the urban resistance World in 2022, and the percentage of utilization in rural areas increased from 31% in 2019 to 46% in 2022 [1]. In the present day, the Internet has been utilized to an unprecedented extent. The damage inflicted by cybercrime becomes more catastrophic as the number of Internet users and devices increases. Cybercrime is the term used to describe the illicit use of a computer or the Internet. The most frequently employed attacks include malware blackmail, identity theft, invasion of privacy, and denial of service (DoS). The annual damages caused by cybercrimes are expected to increase from $2 trillion in 2015 to $6 trillion by 2021 in the field of economics. The World Economic Forum (WEF) designated cyber attacks as the third global risk for 2018. This position is expected to remain unchanged in the future [2]. The necessity for dependable protection systems is both urgent and logical in light of the aforementioned hazards. Intrusion Detection Systems (IDS) are operational systems that serve as a complement to firewalls. Their primary objective is to identify malicious intrusions that have the potential to compromise the functionality of a network or system [3]. In terms of deployment position, IDS can be classified into two types: host-based IDS (HIDS) and network-based IDS (NIDS). NIDS has the ability to monitor the entire network traffic, including all packets that pass through, whereas HIDS functions as a stationing on a singular complex, such as the host of a personal computer. Consequently, the NIDS is more frequently employed in the context of a large-scale network. Detection is integral to both types of IDS, and the methods used for identification can be classified into two categories based on the mechanism: signature-based and anomaly-based detection systems. The primary distinction between the two is that signature-based detection identifies known malevolent intrusions, whereas anomaly-based detection can identify new intrusions by analyzing any abnormalities in the traffic. The definition of anomaly is predicated on the offset attributes from the baseline. The supervisions conducted through signature detection are somewhat lagging and inadequate to address all intrusions, particularly those that are novel, as they are contingent upon the updated database that contains malignant behavior signatures. Therefore, the

Research Article

implementation of anomaly-based detection is extremely important. The fundamental principle of anomaly-based detection is to construct a baseline profile that represents the behaviors that are beyond the alerting range by analyzing network traffic. Upon completion of these studies, anomaly-based IDS is adequately equipped to detect and supervise traffic by contrasting the existing traffic with the standard baseline [3]. Nevertheless, it is impossible to manually establish each baseline profile for anomaly-based detection due to the Internet's expansive data capacity and the rapidity of its refresh rate. Consequently, the need for a self-learning algorithm that was effective arose. Machine learning (ML) is an algorithm that operates as a model within artificial intelligence (AI). It is designed for data training, enabling the construction of models that can independently make decisions and selections, rather than relying on the execution of specific program commands [4]. ML has already demonstrated its capabilities as a formidable approach in a variety of fields, including consumer services, logistics chain control, biology, and computer science [5]. Deep Learning (DL) is a sub-branch of machine learning, and it has become increasingly popular to incorporate this algorithm into anomaly-based IDS [6]. The employment of ML to perform network traffic supervision and detection necessitates dealing with an immense volume of data [7]. Applying machine learning to early detections in the cyber security sector is an effective method for identifying new attacks [8]. ML models were frequently employed to identify assaults from cloud security, malware, and malicious intrusions from 2009 to 2014. This trend experienced a substantial increase following the 2013 increase in DL. The computer security domain has made extensive use of ML and DL up until 2020 [9].

The paper contributes to enhancing intrusion detection by leveraging **Random Forest (RF) and Support Vector Machine (SVM)** for robust classification. It utilizes **Stratified Cross-Validation** to ensure balanced class representation and mitigate the effects of data imbalance. **Random Over-Sampling** is employed to enhance the learning of minority class patterns, reducing bias in classification. The **SVM model**, with its high-dimensional feature mapping, improves the decision boundary for complex attack patterns, while **RF enhances generalization** by aggregating multiple decision trees. The combination of these techniques results in improved detection accuracy, reduced false positives, and a more resilient Intrusion Detection System (IDS).

II LITERATURE REVIEW

Author(s) & Year	Focus / Contribution	Methodology / Model	Key Findings / Results
Zaheer Abbas & Seunghwan Myeong (2023)	Forecasting ML application in Industrial Cloud focusing on privacy and trust issues	XGB Model with validation metrics	Achieved 97.50% accuracy, 97.60% precision & recall, 97.50% F1 score
Albara Awajan et al. (2023)	Deep learning-based IDS for detecting IoT attacks like DDoS, Sinkhole, and Blackhole	Deep Learning-based IDS	Achieved 93.74% accuracy with balanced precision, recall, and F1 score
Iqbal H. Sarker et al. (2020)	IDS for real-time detection of hostile traffic targeting IoT devices	Deep Learning-based IDS	Enhanced IoT security through real-time intrusion detection
Mohanad Sarhan et al. (2021)	Development of IntruDTree-based IDS, assessing feature extraction impact	IntruDTree (Intrusion Detection Tree)	Improved detection precision with reduced computational complexity
Abbas Jamalipour et al. (2021)	Challenges of IDS in IoT networks and importance of universal feature sets	ML models and feature reduction techniques	Identified need for universal feature sets to maximize detection performance
Zahedi Azam et al. (2023)	Survey of IoT security vulnerabilities and advanced IDS methodologies	Reinforcement Learning, Deep Learning, Machine Learning	Comprehensive security enhancement strategies using advanced ML and DL techniques
Javed Asharf et al. (2020)	Review of contemporary IDS challenges and datasets used in IoT security research	Literature Review	Discussed evolving IDS challenges and provided an overview of commonly used benchmark datasets

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

III MATERIAL AND METHODS

KDD 1999

One popular benchmark for assessing intrusion detection systems is the KDD99 dataset. It includes network connection data that are either classified as normal or as part of particular attack types (such as denial-of-service, probing, remote-to-local, and user-to-root). 41 attributes are used to represent each connection, and these features are divided into three primary groups according to their characteristics and role in detecting network intrusions. An outline of these feature types and their importance is provided below:

Table 1 Features Description KDD 99 [Satish Kumar et.al.(2020)]

Feature Name	Type	Description
Duration	C	Length of the connection
Protocol-type	D	Type of protocol
Service	D	Network service at the destination
Flag	D	Normal or error status of the connection
Src-bytes	C	Number of data bytes from source to destination
Dst-bytes	C	Number of data bytes from destination to source
Land	D	1 if connection is from/to the same host/port; 0 otherwise
Wrong fragment	C	Number of "wrong" fragments
Urgen	C	Number of urgent packets

CLASSIFICATIOM MODELS

Random Forest (RF): Random Forest is an ensemble learning algorithm that enhances classification accuracy and minimizes overfitting by constructing multiple decision trees and combining their predictions. It functions by combining the votes of the majority from each tree.

Support Vector Machine (SVM): A supervised learning system called Vector Machine maximizes the margin between classes to get the best hyperplane for class separation. It can handle both linear and non-linear classification using kernel functions and works well in high-dimensional areas.

VGG16 is a deep convolutional neural network that comprises 16 weight layers, including 13 convolutional layers and 3 completely connected layers. It effectively captures fine-grained spatial features by employing small 3x3 convolution filters throughout the network.

VGG19: VGG19 is an enhanced version of VGG16 that includes 19 weight layers, including 16 convolutional layers and 3 completely connected layers. Similar to VGG16, it utilizes 3×3 convolution filters; however, it has an additional layer

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

that enables the network to learn more intricate and hierarchical features from the input data. Due to its increased depth, VGG19 generally exhibits slightly superior performance in comparison to VGG16, despite necessitating more computational resources and training time. **Stratified Cross-Validation:**

This technique splits the dataset into k subsets while ensuring that the target variable (e.g., 'Attack', 'Non-Attack') is distributed proportionally across all splits. This is especially useful in imbalanced datasets to ensure that each fold has a good representation of both classes.

Random Over-Sampling:

Random Over-Sampling duplicates minority class samples to balance the class distribution, enabling the to learn from a more even class distribution. This can lead to improved classifier performance when faced with imbalanced datasets.

if the dataset contains **N** samples with class labels **y∈{0,1}** SCV ensures that each fold D_i maintains the class proportions:

$P(y=1|D_i) \approx P(y=1|D)$

To further tackle class imbalance, **Random Over-Sampling (ROS)** is employed, where minority class instances are **replicated** to balance class distribution. Let $\mathbf{N_0}$ and $\mathbf{N_1}$ be the number of samples in class 0 (majority) and class 1 (minority), respectively. ROS increases the minority class samples to match the majority class:

$N_1^{'}=N_0$

While ROS strengthens model learning by preventing bias toward the majority class, it may introduce overfitting, especially when synthetic patterns fail to generalize. Hence, coupling ROS with SCV ensures that artificially introduced instances are validated across multiple folds, improving model stability.

SVM's strength in high-dimensional feature spaces makes it an ideal candidate for intrusion detection, but its performance is highly dependent on hyperparameter tuning**.** Given training samples (x_i,y_i) with feature vectors xix_ixi and labels yi∈{−1,1} the SVM optimization problem is formulated as:

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i$$

$y_i\,(w^T x_i+b) \geq 1-\xi_i \quad ,\xi_i \geq 0, \forall_i$

Where C is the regularization parameter controlling the trade-off between maximizing the margin and minimizing classification errors. Unlike a simple train-test split, cross-validation provides a more reliable, unbiased estimate of model performance, preventing misleading conclusions that might arise from a single partition.

IV PROPOSED SYSTEM

The popular NSL-KDD dataset, which is perfect for researching network traffic, will be used by the proposed intrusion detection system. In order to train the model, the system's first actions include collecting and cleaning up the input data. Techniques such as Simple Imputer can assist in completing the gaps in data. During preprocessing, label encoding is used to convert categorical features into numerical values. Following preprocessing, the system uses Principal Component Analysis (PCA) for feature extraction in order to decrease the dimensionality and boost the effectiveness of the subsequent machine learning models. Only the most instructive features are retained. An 80/20 split ratio is then used to separate the data into training and test sets. Then, stratified 5-fold cross-validation is used to ensure that each fold correctly depicts both normal and attack data. To correct for any class imbalance and improve the model's ability to identify minority class attacks, the method makes use of Random Over-sampling (RO). The system uses machine learning Random Forest (RF) and Support Vector Machine (SVM) to classify potential invasions. and deep learning models, vgg16 and vgg19 techniques, and ultimately, a performance evaluation that considers metrics like Precision, Recall, F1-Score, and AUC-PR curves to gauge the system's effectiveness. This strategy will offer a reliable solution to the challenge of precisely and effectively identifying various types of network intrusions while minimizing false positives.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

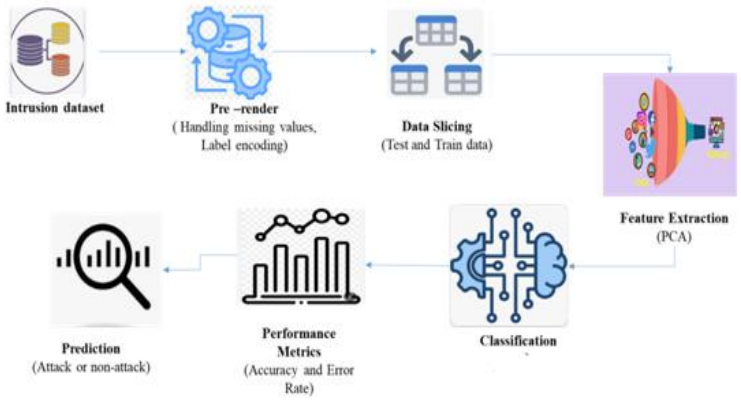


Figure 1: System Architecture

Data Preprocessing

a) Data Normalization

Each feature X is scaled using Min-Max normalization:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where:

X_{min} and X_{max} are the minimum and maximum values of the feature X.

2. Feature Extraction-Using PCA (Principal Component Analysis)-PCA transforms the dataset into a new space of reduced dimensions.

a) Covariance Matrix Calculation-For a dataset X with m features, the covariance matrix is:

$$C = \frac{1}{n}(X^T)$$

Where n is the number of samples.

b) Eigenvalue Decomposition

Solve the eigenvalue equation:

$Cv=\lambda v$

Where:

v is the eigenvector (principal component),

λ is the eigenvalue.

Projection to Lower Dimensions

Project the data onto the first k principal components:

$X_{PCA}=XV_k$

Where V_k consists of the top k eigenvectors.

3. Data Splitting-The dataset is split into training and testing sets:

$(X_{train}, Y_{train}), (X_{test}, Y_{test})$

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Where:

- Xtrain, Ytrain are training data and labels,
- Xtest ,Ytest are test data and labels.

Classification Models

a) Support Vector Machine (SVM)-SVM finds the optimal hyper plane that separates data points. The decision function is:

$f(X)=w^TX+b$

Where:

- w is the weight vector,
- b is the bias.

The optimal hyper plane maximizes the margin:

$$\frac{1}{\|w\|}$$

$y_i(w^Tx_i+b)\geq 1,\forall_i$

b) Random Forest (RF)-RF consists of multiple decision trees:

$$\mathbf{H(X)=\frac{1}{N}\sum_{i=1}^{N} h_i(X)}$$

Where:

- $h_i(X)$ is the prediction of the i-th tree,
- N is the total number of trees.

The final class is determined by majority voting

VGG16:

Structure: 13 Convolutional + 3 Fully Connected layers.

Input→(2×Conv3-64)→Pool→(2×Conv3-128)→Pool→(3×Conv3-256)→Pool→(3×Conv3-512)→Pool→FC→Softmax

VGG19:

Structure: 16 Convolutional + 3 Fully Connected layers.

Input→(2×Conv3-64)→Pool→(2×Conv3-128)→Pool→(4×Conv3-256)→Pool→(4×Conv3-512)→Pool→FC→Softmax

Model Evaluation

Evaluation metrics include:

a) Accuracy

$$\text{Accuracy}=\frac{TP+TN}{TP+TNTP+TN+FP}$$

b) Precision

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

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Recall

$$\text{Recall} = \frac{TP}{TP+FN}$$

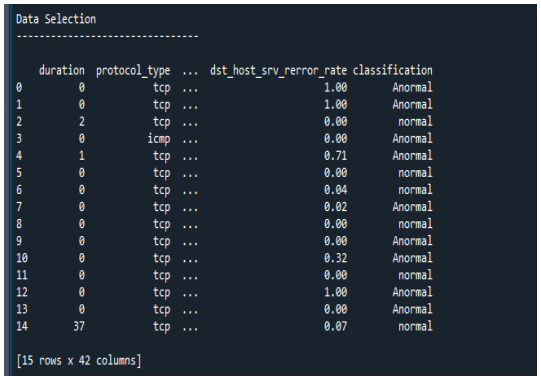
$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} \times \text{Recall}}$$

where:

- TTP = True Positives,
- TN = True Negatives,
- FP = False Positives,
- FN = False Negative

V RESULT DISCUSSION

A classification pipeline using stratified 5-fold cross-validation along with Random over Sampler and a Random Forest Classifier (RF) for handling class imbalance in a dataset. Starting with the required library definitions, it initializes a 5-fold cross-validation (StratifiedKFold) to guarantee that each fold has an equal proportion of class labels. In order to achieve dataset parity, a RandomOverSampler is employed to add more members from the underrepresented class to the training data set at each fold. A Random Forest classifier is used to train the model. Its performance is assessed on a validation set by computing accuracy for each fold. Then, it is tested by making predictions on a separate test set. Model performance is assessed by computing accuracy, precision, recall, sensitivity, specificity, and a classification report. In addition, the code computes the area under the curve (AUC) and presents it alongside a precision-recall curve, which reveals how well the model balances recall and accuracy. Execution time for the entire procedure is also measured, and the final model's metrics are presented, including accuracy, error rate, and other important classification metrics.

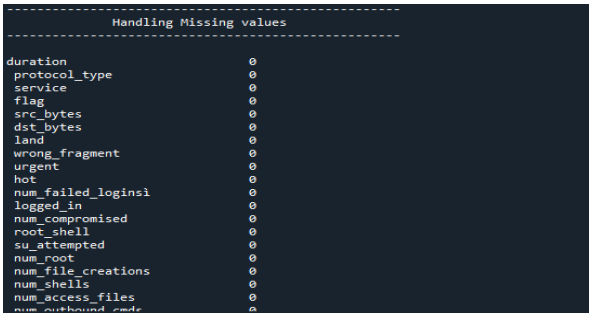


```
Data Selection
------------------------------

    duration  protocol_type  ...  dst_host_srv_rerror_rate classification
0          0            tcp  ...                      1.00       Anormal
1          0            tcp  ...                      1.00       Anormal
2          2            tcp  ...                      0.00        normal
3          0           icmp  ...                      0.00       Anormal
4          1            tcp  ...                      0.71       Anormal
5          0            tcp  ...                      0.00        normal
6          0            tcp  ...                      0.04        normal
7          0            tcp  ...                      0.02       Anormal
8          0            tcp  ...                      0.00        normal
9          0            tcp  ...                      0.00       Anormal
10         0            tcp  ...                      0.32       Anormal
11         0            tcp  ...                      0.00        normal
12         0            tcp  ...                      1.00       Anormal
13         0            tcp  ...                      0.00       Anormal
14        37            tcp  ...                      0.07        normal

[15 rows x 42 columns]
```

Fig.3 dataset selection



```
------------------------------------------------------
             Handling Missing values
------------------------------------------------------

duration                     0
 protocol_type               0
 service                     0
 flag                        0
 src_bytes                   0
 dst_bytes                   0
 land                        0
 wrong_fragment              0
 urgent                      0
 hot                         0
 num_failed_loginsi          0
 logged_in                   0
 num_compromised             0
 root_shell                  0
 su_attempted                0
 num_root                    0
 num_file_creations          0
 num_shells                  0
 num_access_files            0
```

Fig. 4 handling missing

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

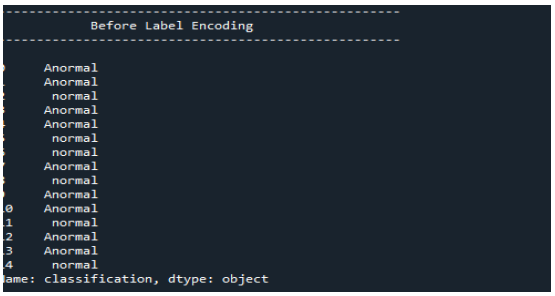


Fig.5 before label encoding

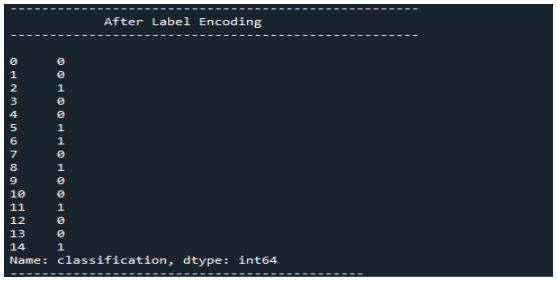


Fig.6 after label encoding

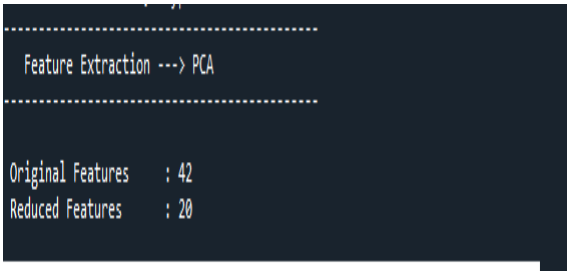


Fig.7 feature extraction

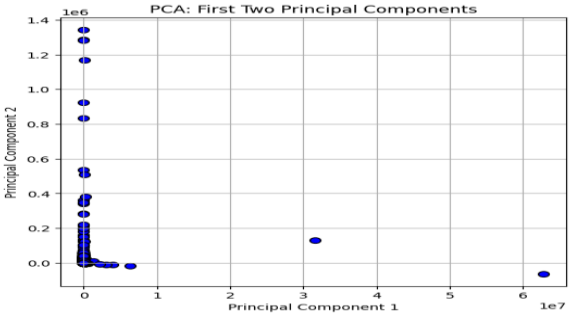


Fig.8 PCA Performance

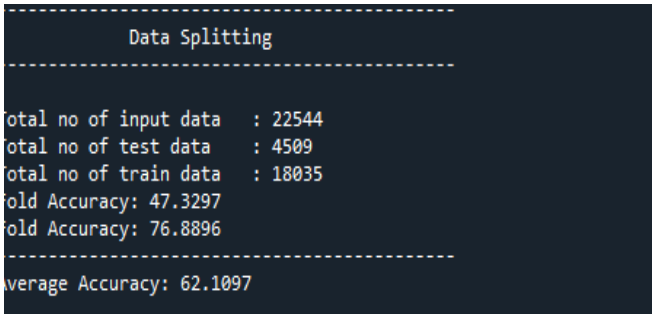


Fig.9 Data Splitting

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
Enter Prediction Number :5
Identified - Attack
```

Table 3 proposed performance comparison with existing work

Models		Accuracy (%)
Convolutional Neural Network (CNN) and bidirectional long short-term memory (BiLSTM)	Existing[19]	97.77
SVM with (StratifiedKFold	Proposed	92
RF with (StratifiedKFold		97
VGG16		97.89
VGG19		98.56

Table 3 presents a performance comparison of different models in terms of accuracy, focusing on both existing and proposed approaches for a specific task. The VGG19 model outperformed all other approaches, achieving the highest accuracy of 98.56% in the performance comparison of various models for the given task. The Random Forest (RF) using StratifiedKFold attained 97% accuracy, while the VGG16 model closely followed with an accuracy of 97.89%, indicating strong performance. The accuracy of the current CNN and BiLSTM-based model, as reported in [19], was 97.77%, which is marginally lower than the VGG models but still strong. At the same time, the proposed model and the Support Vector Machine (SVM) with StratifiedKFold achieved reduced accuracies, with the proposed model achieving 92%. This indicates that the proposed model has the potential for improvement in comparison to other deep learning and ensemble-based approaches. In conclusion, the comparative analysis demonstrated that VGG19 was the most effective model.

VI CONCLUSION

To effectively identify potential dangers to network security, the suggested intrusion detection system employs the NSL-KDD dataset. A variety of attack types can be accurately detected by the system by utilizing data pre-processing techniques, such as handling missing data, label encoding, and dimensionality reduction through principal component analysis (PCA), Deep learning architectures, and VGG19 in particular, have been shown to deliver superior performance in intrusion detection tasks, as evidenced by the comparative analysis of multiple categorization models. The best accuracy was attained by VGG19, which was 98.56%, demonstrating that it has a good potential to capture intricate patterns and delicate elements within the dataset. In addition, VGG16 and Random Forest (RF) demonstrated great accuracy, closely following VGG19. Meanwhile, the old CNN and BiLSTM-based model continued to achieve results that were competitive. Despite the fact that they were effective, the suggested model and SVM with StratifiedKFold produced somewhat lower accuracies, which indicates that there may be possible areas for further improvement in the future. Taking everything into consideration, the research comes to the conclusion that deep convolutional neural networks, and more specifically VGG19, are the most resilient and dependable models for the task at hand. These networks offer significant gains in detection accuracy and system resilience.

Future research could focus on incorporating real-time data streaming for further optimization and developing adaptive models to address emerging attack vectors.

REFERENCES

[1] ITU. (n.d.). Internet use in urban and rural areas. Retrieved March 2, 2023, from https://www.itu.int/itu-d/reports/statistics/2022/11/24/ff22-internet-use-in-urban-and-ruralareas/
[2] Nguyen, T. (2023, January 6). A review of Cyber Crime. Retrieved March 3, 2023, from https://dzarc.com/social/article/view/244
[3] Rao, U., & Nayak, U. (1970, January 01). Intrusion detection and prevention systems. Retrieved March 3, 2023, from https://link.springer.com/chapter/10.1007/978-1-4302-6383-8_11#Abs1
[4] Dua, S., & Du, X. (2011). Data Mining and machine learning in Cybersecurity. Boca Raton, FL: CRC Press.
[5] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, Perspectives, and prospects. Science, 349(6245), 255-260. doi:10.1126/science.aaa8415
[6] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2020). Network intrusion detection system: A systematic study of machine learning and Deep Learning Approaches. Transactions on Emerging Telecommunications Technologies, 32(1). doi:10.1002/ett.4150
[7] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. doi:10.1038/nature14539
[8] Fraley, J. B., & Cannady, J. (2017). The promise of machine learning in Cybersecurity. SoutheastCon 2017. doi:10.1109/secon.2017.7925283
[9] Prasad, R., & Rohokale, V. (2019). Artificial Intelligence and machine learning in cyber security. Springer Series in Wireless Technology, 231-247. doi:10.1007/978-3-030-31703- 4_16
[10] Toya Acharya;Ishan Khatri;Annamalai Annamalai;Mohamed F ChouikhaEfficacy of Machine Learning-Based Classifiers for Binary and Multi-Class Network Intrusion Detection 2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS) Year: 2021 | Conference Paper | Publisher: IEEE DOI: 10.1109/I2CACIS52118.2021.9495877
[11] Chung-Ming Ou Host-based Intrusion Detection Systems Inspired by Machine Learning of Agent-Based Artificial Immune Systems 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA) Year: 2019 | Conference Paper | Publisher: IEEE DOI: 10.1109/INISTA.2019.8778269
[12] Zaheer Abbas, Seunghwan Myeong (2023**)** Enhancing Industrial Cyber Security, Focusing on Formulating a Practical Strategy for Making Predictions through Machine Learning Tools in Cloud Computing Environment Volume 12 Issue 12 10.3390/electronics12122650
[13] Amar Amouri,Vishwa T. Alaparthy,Salvatore D. Morgera (2020) "A Machine Learning Based Intrusion Detection System for Mobile Internet of Things" 2020, 20(2), 461; https://doi.org/10.3390/s20020461, 14 January 2020
[14] Muhammad Almas Khan,Muazzam A. Khan,Sana Ullah Jan,Jawad Ahmad,Sajjad Shaukat Jamal,Awais Aziz Shah,William J. Buchanan (2021) "A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT" 2021, 21(21), 7016; https://doi.org/10.3390/s21217016, 22 October 2021
[15] Nahida Islam , Fahiba Farhin , Ishrat Sultana , M. Shamim Kaiser , Md. Sazzadur Rahman , Mufti Mahmud , A. S. M. Sanwar Hosen and Gi Hwan Cho,(2021) "Towards Machine Learning Based Intrusion Detection in IoT Networks" DOI:10.32604/cmc.2021.018466, CMC, 2021, vol.69, no.2
[16] Yakub Kayode Saheed , Aremu Idris Abiodun , Sanjay Misra c, Monica Kristiansen Holone c, Ricardo Colomo-Palacios c(2022) "A machine learning-based intrusion detection for detecting internet of things network attacks" Volume 61, Issue 12, December 2022, Pages 9395-9409,
[17] Vanlalruata Hnamte, Jamal Hussain (2023) "DCNNBiLSTM: An Efficient Hybrid Deep Learning-Based Intrusion Detection System " Volume 10, June 2023, 100053,
[18] Albara Awajan (2023) "A Novel Deep Learning-Based Intrusion Detection System for IoT Networks" 2023, 12(2), 34; https://doi.org/10.3390/computers12020034, 5 February 2023
[19] rachid ben said, zakaria sabir and iman askerzade **"**CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software Defined Networking with Hybrid Feature Selection". VOLUME XX, 2017 *Digital Object Identifier 10.1109/ACCESS.2022.Doi Number*