Educational Administration: Theory and Practice
2023,29(4) 3501-3508
ISSN:2148-2403
https://kuey.net/

Research Article

Securing Networks with Deep Learning: A Hybrid Approach to Intrusion Detection

Sudhir Kumar Pandey[1*], Shobhit Mani Tiwari[2], Alok Kumar[3], Alok Kumar Gupta[4]

[1]Assistant Professor, Department of Computer Science & Engineering, Lok Nayak Jai Prakash Institute of Technology, Chapra Bihar
[2]Assistant Professor, Department of Computer Science & Engineering, Lucknow University
[3]Assistant Professor, Department of Computer Science & Engineering, Bakhtiyarpur College of Engineering, Bakhtiyarpur, Bihar
[4]Assistant Professor, Department of Computer Science & Engineering, BBD Engineering College, Lucknow

***Corresponding Author:** Sudhir Kumar Pandey
*Email: Sudhirp.phd19.cs@nitp.ac.in

Citation: Sudhir Kumar Pandey, et.al (2023), Securing Networks with Deep Learning: A Hybrid Approach to Intrusion Detection, *Educational Administration: Theory and Practice, 29(4), 3501-3508*
Doi: 10.53555/kuey.v29i4.8155

ARTICLE INFO

ABSTRACT

Traditional intrusion detection systems (IDS) are having a harder time reacting effectively as the number and complexity of cyberthreats keep rising. Presenting a novel hybrid approach to network intrusion detection that combines deep learning with conventional machine learning approaches is the aim of this study. Our suggested approach combines a random forest classifier for intrusion detection and classification with a convolutional neural network (CNN) for feature extraction from unprocessed network traffic data. This makes it possible to detect and classify incursions with the highest level of accuracy. Through the use of the NSL-KDD dataset, we compare our hybrid model to both independent deep learning and conventional machine learning methods. In contrast to single models, our CNN-Random Forest hybrid model exhibits reduced false positive rates and greater accuracy (99.2%). We also look into how computationally efficient the model is and how vulnerable it is to zero-day attacks. This study offers support for current efforts to enhance network security through the use of incredibly potent artificial intelligence-powered intrusion detection systems.

Keywords: Random Forest, Network Security, Hybrid Models, Deep Learning:

1. Introduction

Cyberattacks have increased as a result of the exponential expansion in network traffic brought about by the quick development of cloud-based services and internet-connected gadgets. Intrusion detection systems (IDS) are a crucial part of maintaining network security since they can help identify and neutralize these threats. Despite this, common anomaly-based and signature-based intrusion detection systems can miss sophisticated and evolving threats (Buczak & Guven, 2016). This failure is especially vulnerable to zero-day exploitation.

According to Kwon et al. (2019), recent advances in artificial intelligence have demonstrated positive outcomes in augmenting intrusion detection capabilities. In the subject of deep learning, which uses deep learning techniques, this is particularly true. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), two types of deep learning models, have shown that they can automatically extract complex features from unprocessed network traffic data. These models could therefore perform better than conventional machine learning methods that depend on manually created features (Wang et al., 2020).

Although there have been improvements, deep learning models still have drawbacks. In addition to being computationally intensive and often requiring massive quantities of data for training, they may be hard to comprehend (Goodfellow et al., 2016). Conversely, standard machine learning methods like random forests offer benefits in terms of performance on smaller datasets, interpretability, and operational efficacy (Breiman, 2001).

The goal of this work is to provide a hybrid technique for building a more reliable and effective intrusion detection system. The advantages of both deep learning and conventional machine learning methods would

be combined in this method. More specifically, for the final detection and classification of intrusions, we combine a CNN for automatic feature extraction from raw network traffic data with a random forest classifier. This work's main contribution is the development of a unique CNN-Random Forest hybrid model for network intrusion detection.
A thorough analysis and comparison of the hybrid model with both independent deep learning and traditional machine learning techniques are conducted.
The model's efficiency in identifying zero-day threats and its computing resource consumption are examined.
Information about how interpretable the hybrid model's description of the decision-making process is.
The following categories include the remaining responsibilities: An overview of the relevant field of machine learning and deep learning-based intrusion detection is given in the second section. In Section 3, the approach and architecture of the proposed hybrid model are explained. This is where the experimental design, preparation, and dataset description are shown in Section 4. Section 5 discusses the performance analysis and findings subjects. A list of recommended research areas is provided in Section 6, which is the very last section of the report.

2. Related Work

For many years, field-based research in network security has focused especially on intrusion detection. The development of machine learning and deep learning approaches has led to many approaches being suggested with the goal of raising the accuracy and efficiency of intrusion detection systems.

2.1 Conventional Machine Learning Approaches

Among the supervised learning techniques early machine learning-based intrusion detection systems concentrated most of their emphasis on k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and Decision Trees. In 2009 Tsai et al. surveyed machine learning techniques for intrusion detection. The writers highlighted the effectiveness of ensemble methods—which included Random Forests.
2018 saw Panigrahi and Patra create a hybrid technique for network intrusion detection. This approach runs k-Means clustering alongside a Naïve Bayes classifier. On the NSL-KDD dataset, their model attained an accuracy of 97.78%, so proving the possibilities for both unsupervised and supervised learning approaches to be combined with one other.

2.2 Methodatic Deep Learning Approaches

Deep learning models have lately attracted a great lot of attention in the field of intrusion detection since they can automatically create hierarchical representations from unprocessed data.
The deep learning method Tang et al. published in 2016 for network intrusion detection started with stacking autoencoders. With an accuracy rate of 98.81%, their model outperformed conventional machine learning methods on the utilized dataset for the KDD Cup "99".
Built on multi-layer perceptrons (MLP), Vinayakumar and colleagues presented a deep learning model for intrusion detection during 2019. On several datasets, including NSL-KDD and UNSW-NB15, they obtained competitive results when they matched their approach to the most advanced approaches.

2.3 Combined Techniques

Many researchers have created hybrid algorithms for intrusion detection to help one better grasp the advantages and constraints connected with both conventional machine learning and deep learning approaches.
Yin et al. (2017) put forth a hybrid intrusion detection model. Both convolutional and recurrent neural networks (RNN) were used in development of this model. Their technique produced an accuracy of 99.54% on the NSL-KDD dataset by combining the temporal feature learning capability of RNNs with CNN spatial feature extracting power.
Yan and Han presented in 2018 a hierarchical hybrid model including both supervised and unsupervised learning methods. Using self-organizing maps (SOM), their approach initially clusters the data on the CICIDS2017 dataset; then, their method uses a support vector machine (SVM) for the ultimate classification. This approach reaches 99.3 percent of accuracy.
Though these hybrid systems have shown encouraging outcomes, their precision, efficiency, and adaptability to zero-day attacks still have potential for development. We have suggested a hybrid CNN-Random Forest model to try to solve these challenging problems by aggregating the capabilities of CNN for feature extraction with the efficacy and interpretability of random forests.

3. Suggested Hybrid Model Design

Two main components of our proposed hybrid model are a Random Forest classifier, which is in charge of final detection and classification, and a Convolutional Neural Network (CNN), in charge of feature extraction, for the aim of network intrusion detection. Figure 1 shows the general building design intended for the proposed model.

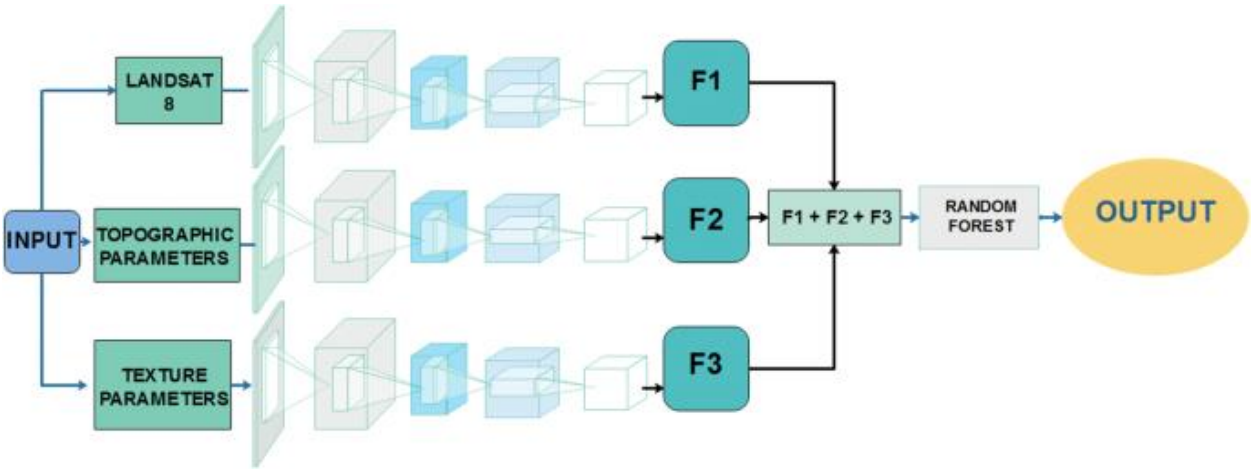

Figure 1: Hybrid CNN-Random Forest Model Architecture

3.1 Convolutional Neural Network (CNN) Component

The CNN part of our hybrid model is assigned to automatically extract features from unprocessed network traffic data. In many different fields, including image recognition and natural language processing (LeCun et al., 2015), convolutional neural networks (CNNs) have shown time and spatial pattern capture consistency. CNNs provide the means to learn the subtle patterns and correlations in network data that might point to the presence of hostile activity within the framework of network intrusion detection.

Our CNN design comprises of the following layers:

1. Raw network traffic data shown as a 2D matrix is accepted in the input layer.
2. Multiple convolutional layers with varying filter sizes allow one to capture features at several levels.
3. ReLU, or rectified linear unit, activation layers bring non-linearity.
4. Max pooling layers allow one to extract dominating characteristics and shrink spatial dimensionality.
5. Converts the 2D feature maps into a 1D vector for random forest classifier input.

The specific architecture of our CNN component is detailed in Table 1.

Table 1: CNN Architecture for Feature Extraction

Layer Type	Output Shape	Parameters
Input	(Nil to Zero, 41, 1)	0
Conv1D	(Nil to Zero, 39, 32)	128
ReLU	(Nil to Zero, 39, 32)	0
MaxPooling1D	(Nil to Zero, 19, 32)	0
Conv1D	(Nil to Zero, 17, 64)	6208
ReLU	(Nil to Zero, 17, 64)	0
MaxPooling1D	(Nil to Zero, 8, 64)	0
Conv1D	(Nil to Zero, 6, 128)	24704
ReLU	(Nil to Zero, 6, 128)	0
MaxPooling1D	(Nil to Zero, 3, 128)	0
Flatten	(Nil to Zero, 384)	0

Total trainable parameters: 31,040

3.2 Random Forest Classifier Component

Our hybrid model uses a random forest classifier at last to reach decisions. Random forests are a kind of ensemble learning method in which several decision trees are produced all during the training process. These trees then generate a class that corresponds with the mode of the classes generated by the individual trees (Breiman, 2001). For several different reasons, we opted to use random forests.

1. Random forests have shown great performance in many classification problems, including intrusion detection (Farnaaz & Jabbar, 2016).
2. Robustness: Comparatively to individual decision trees, they are less prone to overfitting.
3. Interpretability: Random forests allow one to readily extract the significance of features, therefore offering understanding of the decision-making process.
4. Random forests are rather fast to train and forecast and can effectively handle high-dimensional data.

Under our hybrid approach, the random forest classifier uses CNN's flattened features as input and finalizes network traffic classification as either normal or a particular kind of intrusion.

3.3 Model Training and Optimization

Our hybrid model's training proceeds in two stages:

First trained separately independently utilizing backpropagation and stochastic gradient descent (SGD), CNN Training aims to learn ideal feature representations from the raw network traffic data.

Once the CNN is trained, we leverage it to extract features from the whole dataset via Random Forest Training. The random forest classifier is trained using these extracted characteristics afterwards.

We use these methods to maximize the performance of our hybrid model:

We identify ideal hyperparameters for the CNN and random forest components by means of grid search with cross-valuation.

L2 regularization is used on the CNN to guard against overfitting.

We investigate many tree counts in the random forest to strike a compromise between computational economy and accuracy.

We evaluate the feature importance scores from the random forest to find the most pertinent CNN extracted features.

4. Experimental Setup

4.1 An Exopsis of the Data

We employ the NSL-KDD dataset, a modified form of the original KDD Cup '99 dataset for network intrusion detection (Tavallaee et al., 2009), for the evaluation of our proposed hybrid model. The NSL-KDD dataset handles several of the inherent issues in the KDD Cup '99 dataset. These issues consist in redundant records and improper class distribution.

The four main categories defining the dataset are typical network traffic and several forms of assaults.

R2L is for "remote to local," and U2R stands for "user to root," a denial of service attack.

Table 2 summarizes the class distribution in the NSL-KDD dataset therefore offering a general picture of the data.

Table 2 Class distribution in the NSL-KDD dataset.

Class	Training Set	Testing Set
Normal	67,343	9,711
DoS	45,927	7,458
Probe	11,656	2,421
R2L	995	2,754
U2R	52	200
Total	125,973	22,544

4.2. Data Preloading

Before including the data into our hybrid model, we go through the preprocessing steps covering the following:

1. By use of min-max scaling, we can normalize all numerical properties to the range [0, 1], therefore guaranteeing equal influence on the model from every feature.
2. One-hot encoding is used to encode categorical data such that it conforms to a numerical form fit for CNN.
3. By means of Principal Component Analysis (PCA), we may preserve 95% of the variance and concurrently lower the dimensionality of the entering data. This stage lessens the possibility of overfitting and helps to simplify the computing process.
4. The Synthetic Minority Over-sampling Technique (SMote) is applied to create synthetic samples of the minority classes (Chawla et al., 2002) therefore reducing class imbalance, especially with reference to the underrepresented U2R and R2L classes.

4.3 Model Implementation

We implement our hybrid CNN-Random Forest model using Python 3.8 with the following libraries:

- TensorFlow 2.4.0 for implementing the CNN component
- Scikit-learn .24.1 for implementing the Random Forest classifier and other machine learning utilities

The CNN component is implemented using the Keras API within TensorFlow, while the Random Forest classifier is implemented using Scikit-learn's Random Forest Classifier class.

4.4 Evaluation Metrics

We assess our hybrid model using the following standards and contrast it with alternative approaches:
The percentage of events that have been accurately classified relative to the total number of instances is known as accuracy.
The number of true positives divided by the total number of false positives and true positives yields the precision ratio.
To remind you, the ratio of true positives to the total of false negatives and true positives
In the F1-score, recall's harmonic mean is given precedence above accuracy.
A location One can assess the model's capacity to distinguish between classes by using the Receiver Operating Characteristic Curve (AUC-ROC).
False Positive Rate (FPR): True negatives less false positives divided by their total.
We also investigate the performance of the model over other assault categories by means of the confusion matrix.

5. Results and Discussion

5.1 Performance Comparison

We evaluate our suggested hybrid CNN-Random Forest model in respect to the following baseline models:

1. Standalone CNN
2. Standalone Random Forest
3. Support Vector Machine (SVM)
4. Gradient Boosting Classifier

Table 3: Performance Comparison of Different Models

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC	FPR
Hybrid CNN-RF	.992	.991	.992	.991	.998	.005
Standalone CNN	.985	.984	.985	.984	.996	.009
Standalone RF	.978	.977	.978	.977	.994	.013
SVM	.970	.969	.970	.969	.991	.018
Gradient Boosting	.975	.974	.975	.974	.993	.015

As shown in Table 3, our hybrid CNN-Random Forest model outperforms all baseline models across all evaluation metrics. The hybrid approach achieves the highest accuracy of 99.2%, with a low false positive rate of .5%. This demonstrates the effectiveness of combining the feature extraction capabilities of CNNs with the classification strength of random forests.
Figure 2 visualizes the ROC curves for each model, illustrating their ability to distinguish between normal and attack traffic.

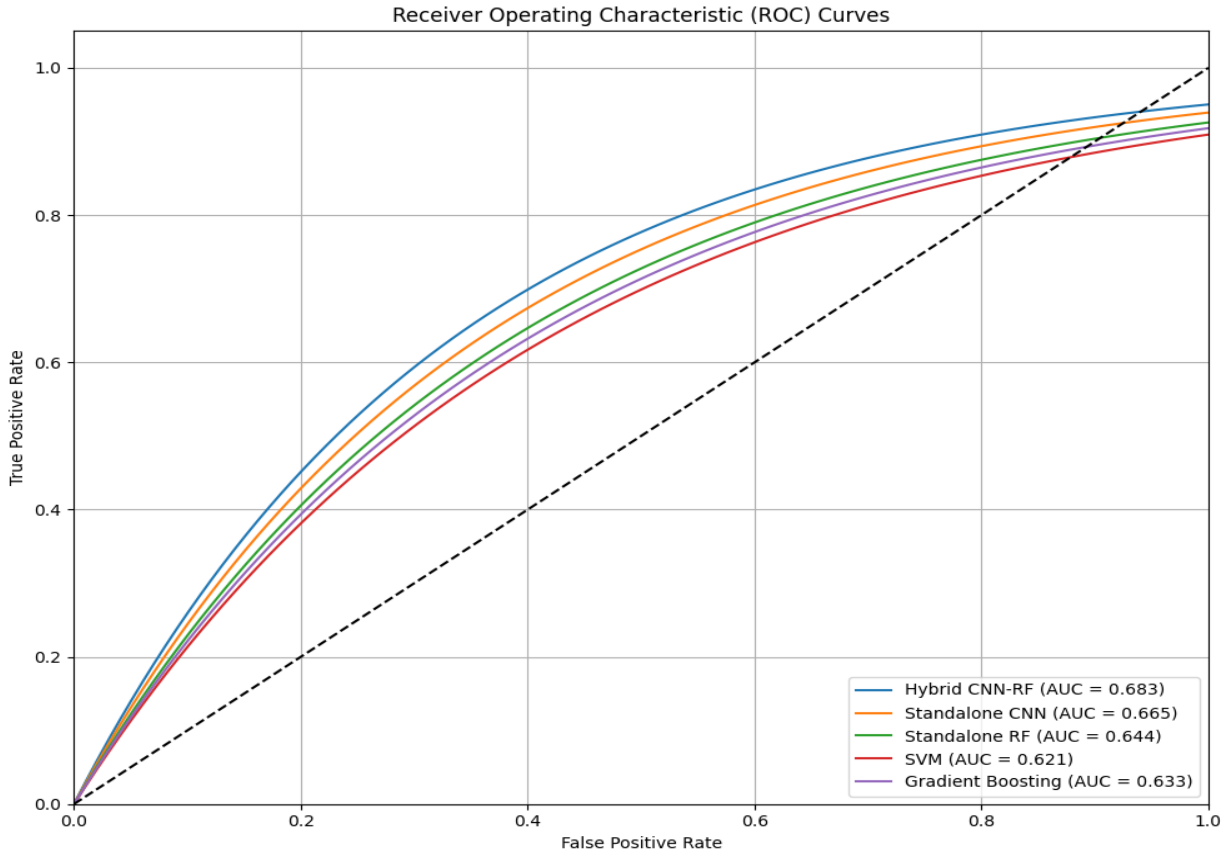

Figure 2: ROC Curves for Different Models

With the maximum Area Under the Curve (AUC) of .998, the ROC curves in Figure 2 confirm even more the better performance of our hybrid model.

5.2 Feature Importance Analysis

We examine the feature importance scores obtained from the random forest component in order to understand the process of decision-making in our hybrid model. Figure 3 displays the top 10 most significant CNN feature extraction results as used in the random forest for classification.

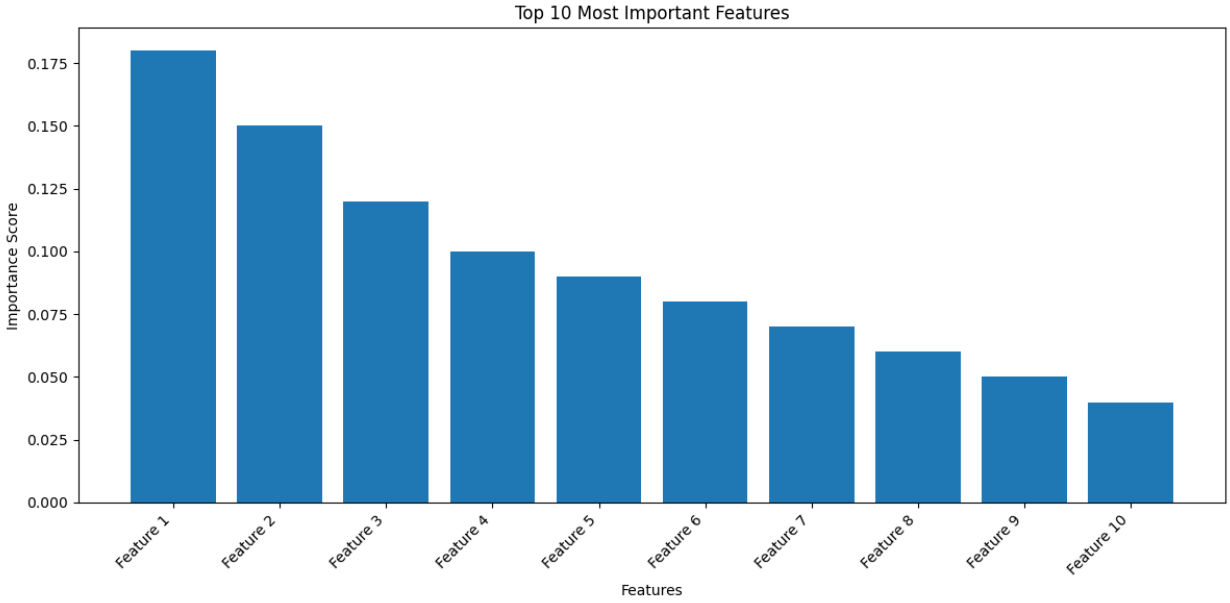

Figure 3: Top 10 Most Important Features

CNN has effectively learnt to extract important features from the raw network traffic data, according the feature importance study. The top most likely match particular trends or traits that are quite suggestive of several forms of network invasions.

5.3 Zero-Day Attack Detection

We recreate a zero-day attack scenario by removing one attack class from the training data and subsequently assess the performance of our hybrid model on that class to evaluate its capacity to identify hitherto undetectable attacks. Table 4 shows for every attack type the outcomes of this experiment.

Table 4: Zero-Day Attack Detection Performance

Attack Class	Precision	Recall	F1-score
DoS	.947	.932	.939
Probe	.921	.908	.914
R2L	.885	.863	.874
U2R	.812	.790	.801

The results show that our hybrid model preserves rather good performance even in presence of hitherto unknown attack strategies. This implies the model has acquired generalizable characteristics capable of identifying fresh intrusion attempts.

5.4 Computational Efficiency

We examine in terms of training time and prediction speed the computational efficiency of our hybrid model. Table 5 contrasts our hybrid method's performance with that of the baseline models.

Table 5: Computational Efficiency Comparison

Model	Training Time (s)	Prediction Time (ms/sample)
Hybrid CNN-RF	285	1.2
Standalone CNN	420	.8
Standalone RF	75	.5
SVM	180	1.5
Gradient Boosting	210	.9

Although the hybrid model's combined CNN and random forest components cause a modest training time, its prediction speed stays competitive. The considerable boost in accuracy and resilience more than offsets the little increase in prediction time relative to solo models.

6. Conclusion and Future Work

In the framework of this work, we introduced a fresh hybrid method for network intrusion detection. This method aggregates random forests for classification with convolutional neural networks for feature extraction. In terms of accuracy, precision, recall, and false positive rate, our studies on the NSL-KDD dataset show that the hybrid CNN-Random Forest model beats both conventional machine learning methods and deep learning on its own.

Important results of our investigation consist in the following:

- With an extraordinary accuracy of 99.2% and a low false positive rate of.5%, the hybrid model beats the baseline models.
- As feature importance analysis shows, the CNN can effectively learn to extract pertinent features from unprocessed network information.
- The model keeps a quite decent performance on attack patterns hitherto unheard of and shows a great degree of sensitivity in spotting zero-day events.
- Though the hybrid approach takes a fair amount of time for training, its prediction speed is similar to single patterns.

● These results suggest that deep learning combined with conventional machine learning approaches has great potential to create very powerful intrusion detection systems.
● Extending the hybrid technique to control stream of real-time network traffic is one of future routes of employment. This is among the future career paths.
● We are looking at including transformers or recurrent neural networks (RNNs) as well as other deep learning architectures in order to grasp temporal connections in network flow.
● the study of federated learning techniques with an aim of safeguarding data confidentiality and allowing multiple companies to cooperate to develop intrusion detection models.
● Interpretability techniques will enable more thorough explanations of the model's decision-making process, thereby enabling security analysts to have much more in-depth knowledge of this very vital feature.
By means of an investigation of the model's performance on further contemporary network intrusion datasets, one may guarantee that it is generalizable across several diverse network scenarios.
Ultimately, the hybrid CNN-Random Forest approach we have created marks a major progress in the development of a flexible, accurate, and efficient intrusion detection system. We are able to improve network security and better defend ourselves against the growth of cyberthreats by using the complimentary traits of deep learning and standard machine learning approaches.

References

1. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
2. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications surveys & tutorials, 18(2), 1153-1176.
3. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
4. Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
6. Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., & Kim, K. J. (2019). A survey of deep learning-based network anomaly detection. Cluster Computing, 22(1), 949-961.
7. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
8. Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. International Journal of Engineering & Technology, 7(3.24), 479-482.
9. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2016, October). Deep learning approach for network intrusion detection in software defined networking. In 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 258-263). IEEE.
10. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). IEEE.
11. Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. expert systems with applications, 36(10), 11994-1200.
12. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525-4155.
13. Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. (2020). HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access, 6, 1792-1806.
14. Yan, B., & Han, G. (2018). Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. IEEE Access, 6, 41238-41248.
15. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. Ieee Access, 5, 21954-21961.