International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

Smart Cyber Threat Detection and Prediction Using Machine Learning

Janani T[1], Dr. Weslin D[2]
[1]PG - Computer Science and Engineering, Mohammed Sathak A.J College of Engineering, Chennai, Tamil Nadu, India
[2]Professor - Information Technology, Mohammed Sathak A.J College of Engineering, Chennai, Tamil Nadu,India
Emails: *jananit412@gmail.com[1], it.weslin@msajce-edu.in[2]*

Abstract
Cybercrime is now being committed at an alarming rate due to the rise of digital communications, cloud computing and networks connecting devices across the globe. Modern cybercrime is much more difficult to find than previously and is evolving rapidly and dynamically. Traditional IDSs use mostly "signature based" techniques, which work well to recognize attacks if they have been identified before. However, new and innovative methods of cybercrime can be difficult to recognize by traditional IDS's. This problem can be solved by using a Machine Learning Based Cyber Attack Prediction and IDS Framework. In this research, we will develop a machine learning based cyber-attack prediction and intrusion detection framework using the NSL-KDD Dataset. It uses a Convolutional Neural Network (CNN) to automate feature extraction from network traffic data without having to manually engineer thousands of features. This CNN Model was developed to efficiently classify network traffic into either Normal or Malicious classes. We used this CNN Model to train and test against four types of attacks in the NSL-KDD Dataset (Denial of Service (DoS), Probing, Remote to Local (R2L), User to Root (U2R)).We also applied various forms of data preprocessing (normalization, one hot encoding, and feature transformation) to enhance the quality of the output of our model and increase the stability of our models during training. Our experiments showed that this CNN-based IDS detected malicious activity with a higher degree of accuracy than other approaches that relied on machine learning rules and/or signatures. It was able to quickly and reliably predict potential threats in real time. This method enhances the ability of intrusion detection systems to recognize known and unknown threats.

Keywords: *Cyber Security; Intrusion Detection System (IDS); Machine Learning; CNN; Network Security*

1. Introduction
The networking systems automatically track their networks traffic and report suspecting or doubtful behaviour, work in one of either style: misuse detection and anomaly detection [1], [6]. Misuse detection method go in search of precise signatures of a known malicious behaviour while anomaly detection will try to form a model for what it contains normal network traffic patterns, then flag deviations from those patterns [1]. The enormous, high-dimensional network traffic and how to precisely detect anomalous traffic is the principal task of intrusion detection systems [6]. Two main categories of NIDSs based on the detection approach exists in the literature including the signature-based (misuse) and anomaly-based intrusion detection system [1], [6]. Anomaly based NIDS are more striking for the research and practitioners as they are capable of detecting any deviation from the normal traffic pattern [1]. Due to this, signature-based antivirus software is becoming a non-entity and it is considered as inconvenient with spoofing of signatures and ignorant latest sophisticated attacks. Because of this, it struggles to remain as a dominant force in latest threat arena. The anomaly-based intrusion detection gives the idea of getting the power to find novel attacks even before they have been understood and

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

characterized by security analysts also as having the power to identify the variations on existing attack methods [6]. For creating data in the Intrusion Detection System (IDS), there is need to line the important working condition to find all the possible type of attacks, it is imperative to examine, transform and model the data. The data quality zeroes on the uprightness and reliability of knowledge obtained and used in an evaluation. Data quantity engages with the amount of knowledge obtained for the validation. The job needs different ground truth databases in its region and thus the practicability could be finished efficiently if the data quality and features of for the precise are good. Image processing, web site analysis, and other things have the standard and permitted ground truth databases for evaluation. Similarly, lot of the PC network intrusion detection systems use the NSL_KDD to classify and analyze network traffic and it clearly depicts the formation of NSL_KDD dataset and its features [6]. Eventually Machine learning has evolved to predict the long run data from the past data. The activity of coaching and prediction involves usage of very special algorithms. Machine Learning feeds the training data to an algorithm, and thus the algorithm utilizes this training data to supply predictions on a replacement test data [8]. Machine learning algorithms are often divided into three types. These are supervised learning, unsupervised learning and reinforcement learning [8]. It also provides the training algorithm, which evaluates the clustering of the input data. At last, the reinforcement learning dynamically interacts with its environment and it also receives good or negative feedback to reinforce its performance. In machine learning and statistics, classification is based on supervised learning approach during which the program learns from the information input given. Then it utilizes this learning to classify new observation [8].

2. Literature Survey

Researchers are developing new forms of Machine Learning and Deep Learning to help find the best ways to defend against cyber-attacks as fast as they appear. Here we will discuss some of the top research findings that have helped identify networks that may be under attack by an attacker using machine learning techniques. Several studies have focused on improving the performance of Intrusion Detection Systems (IDS) using artificial intelligence techniques.

- Chaofan Lu proposed the application of artificial intelligence in intrusion detection systems to enhance detection accuracy and reduce manual intervention [1]. However, the study primarily relied on traditional machine learning techniques, which may not effectively capture complex patterns in large-scale network traffic.
- Harshini Sivakami et al. introduced a Copula Generative Adversarial Network (GAN) combined with a Random Forest classifier to address the issue of class imbalance in intrusion detection datasets [2]. The proposed model improved detection performance for minority attack classes, but the complexity of GAN models increased computational overhead.
- Dashun Liao et al. developed a GAN-based intrusion detection framework to generate synthetic training data and enhance model robustness [3]. While this approach improved classification accuracy, it required extensive training time and computational resources.
- Shiji Zheng proposed a Convolutional Neural Network (CNN)-based model for intrusion detection, which demonstrated superior performance compared to traditional machine learning algorithms [6]. The model effectively captured spatial features in network traffic data,
- resulting in higher detection accuracy and reduced false positives.
- Beom-Su Lee et al. introduced a federated learning-based intrusion detection system that enables distributed model training without sharing raw data [5]. This approach enhances privacy and security but introduces challenges related to communication overhead and model synchronization.

The above studies have all shown significant improvement over traditional IDS's; however, most current IDS's still only examine network traffic at the network level and do not evolve with attackers

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

changing tactics or strategies. "To address the shortcomings described above, this study proposes an intrusion detection model based on CNN's that is capable of making predictions in real time and with high accuracy."

3. Methodology

3.1.Existing System

Current Network Intrusion Detection Systems (NIDS) that operate on the Internet are exposed to many forms of malicious activities including unauthorized access, breaches of information and Denial-of-Service (DoS) attacks [9]. Most of the intrusion detection approaches depend primarily on Signature-Based and Rule-Based methods to detect intrusions [1]. However, these traditional methods do not provide effective means for identifying new and unknown types of threats. Previous research has identified several Machine Learning (ML) techniques to support intrusion detection [6]. However, current systems continue to be limited by issues including low levels of detection accuracy, large amounts of false positives alarms and poor ability to generalize to new attack modes. Furthermore, the quality of datasets currently being used in these systems is generally inadequate. Specifically, the data used may contain redundant or noisy values and/or an unbalanced number of normal vs. abnormal instances [2], [3]. These factors can significantly impact the performance of ML models. Therefore, it would be beneficial to develop improved Data Preprocessing Techniques and Classification Models that will increase detection accuracies while reducing False Positive Rates.

3.2.Proposed System

A Convolutional Neural Network (CNN) was used in the proposed system to develop an automatic Network Intrusion Detection System to address shortcomings with current methods [6]. CNN uses Deep Learning to identify patterns and features in network traffic data without manual intervention.

In the proposed system, pre-processing of the data will be performed using data cleaning, normalization, feature scaling and class balancing on the dataset. These processes help reduce error within the data set and provide a cleaner dataset for learning purposes.

Once the data has been cleaned and processed, the data will then be fed into the CNN. The CNN model includes multiple layers including convolutional layers, pooling layers and fully connected layers. CNN identifies high level characteristics of network traffic data and identifies whether the data falls under the category of "normal" or "attack". Use of this method increases the systems accuracy in identifying attacks, decreases the number of false alarms generated by the system and enables the identification of both known and unknown types of attacks [6]. Overall, the proposed CNN-based intrusion detection system provides a more efficient, accurate and scalable solution than that provided through traditional Machine Learning methods.

Benefits of the Proposed CNN Methodology

- High detection accuracy
- Reduced false alarm generation
- No need for manual feature extraction
- Will work well on large scale networks
- More effective at detecting unknown attacks

4. System Architecture

System architecture overview

- Data Collection (NSL-KDD dataset)
- Data preprocessing
- Feature extraction
- Model training (CNN)
- Prediction Module (web application via Flask)[7 – 10]

4.1.Data Collection

Dataset used: NSL-KDD dataset The kddcup'99 dataset has historically served as one of the very few public datasets available to researchers investigating IDSs until the release of the NSL-KDD Dataset.The NSL-KDD Dataset has undergone processing to remove redundant/irrelevant information from the original kddcup'99 dataset.

- Each record within the dataset will contain 41 features and a single class label describing the nature of the network connection.Class labels: normal or an attack.
- There are 125,973 records in the training set and 22,544 records in the test set[11 – 15].

What is the NSL-KDD Dataset?

A widely recognized and utilized dataset for intrusion detection research, this dataset provides detailed records of numerous network connections, including

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

those labeled as being either "normal" or belonging to one of five distinct categories of attacksShown as Figure 1 Framework of proposed machine learning based Cyber Attack detection system

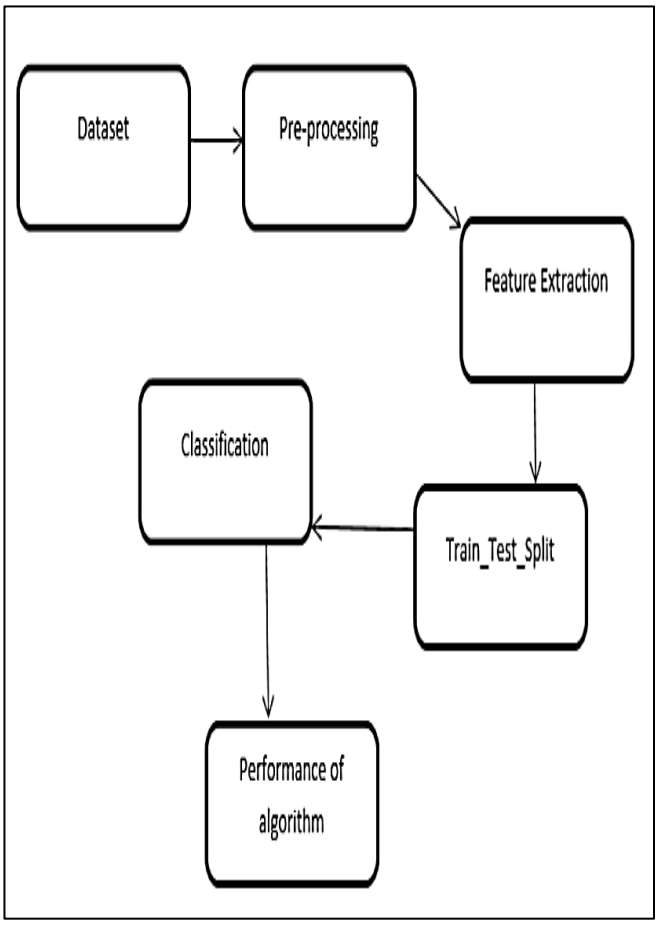

Figure 1 Framework of proposed machine learning based Cyber Attack detection system

The dataset categorizes anomalies into four major classes:

- Denial of Service (DoS): Overwhelming resources (e.g., Neptune, Smurf).
- Probe: Network reconnaissance (e.g., Nmap, Portsweep).
- Remote-to-Local (R2L): Unauthorized remote access (e.g., Guess_password).
- User-to-Root (U2R): Local privilege escalation (e.g., Buffer_overflow).Shown as Figure 2 Workflow of the proposed machine learning based Cyber Attack detection system.

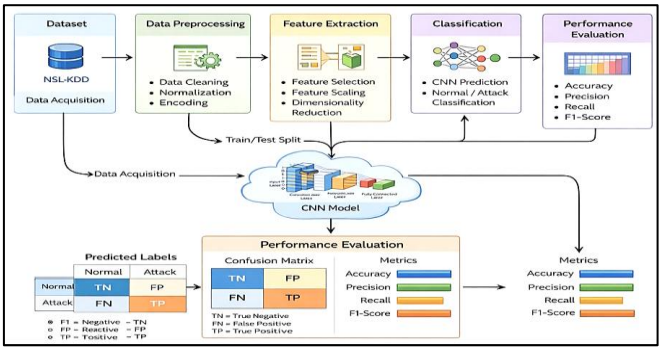

Figure 2 Workflow of the proposed machine learning based Cyber Attack detection system

4.2.Data preprocessing

Before data can be applied to machine learning, it typically requires some amount of pre-processing. Pre-processing may involve various activities with respect to data such as data cleansing, feature extraction, feature normalization, and/or dimensionality reduction. Data pre-processing is necessary because it assists in formatting the data appropriately for training machine learning models.
Steps involved[16 – 20]:

- **Data cleaning**
  Cleaning means removing unwanted data, incorrect data and other unneeded elements. A common example would be a blank field within a database.
- **Handling missing values**
  Missing values occur when there is no value present for a particular field. Some algorithms handle missing values differently than others.
- **Feature scaling**
  Scaling refers to the process of limiting the range of possible output from an algorithm. If all of the inputs to an algorithm have similar scales then the outputs will also tend to have similar scales.
- **Encoding categorical features**
  Categorical features are a special type of feature that represents a category. Examples might include gender or color. These types of features need to be encoded prior to feeding them into most machine learning algorithms.
- **Balancing dataset**
  Balancing means making sure each type of

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

data point is represented equally. For example if we're doing spam vs non-spam email filtering, we want equal amounts of both types of emails otherwise our results will always favor one type over another.

Identified categorical features for training set:Shown as Figure 3,4 Programming code [21]

```
print('Training set:')
for col_name in df.columns:
    if df[col_name].dtypes == 'object' :
        unique_cat = len(df[col_name].unique())
        print("Feature '{col_name}' has {unique_cat} categories".format(col_name=col_name, unique_cat=unique_cat))

print()
print('Distribution of categories in service:')
print(df['service'].value_counts().sort_values(ascending=False).head())
```

```
Training set:
Feature 'protocol_type' has 3 categories
Feature 'service' has 70 categories
Feature 'flag' has 11 categories
Feature 'label' has 23 categories

Distribution of categories in service:
http       40338
private    21853
domain_u    9043
smtp        7313
ftp_data    6860
Name: service, dtype: int64
```

Figure 3 Programming code

Identified categorical features for test set:

```
# Test set
print('Test set:')
for col_name in df_test.columns:
    if df_test[col_name].dtypes == 'object' :
        unique_cat = len(df_test[col_name].unique())
        print("Feature '{col_name}' has {unique_cat} categories".format(col_name=col_name, unique_cat=unique_cat))
```

```
Test set:
Feature 'protocol_type' has 3 categories
Feature 'service' has 64 categories
Feature 'flag' has 11 categories
Feature 'label' has 38 categories
```

Figure 4 Programming code

4.3.Extraction

It is an essential process which extracts meaningful attributes out of raw data so that it can be used by machine learning algorithms. Within the context of intrusion detection feature extraction involves taking raw network data and converting it into structured formats that can be fed into machine learning algorithms. For example:

- Protocol type
- Port number
- Packet size
- Network behavior are extracted for training.

4.4.Model Training Module

Within the Model Training Module, the previously processed data is used to train a convolutional neural network (CNN), allowing the CNN to develop an ability to detect normal versus malicious network traffic. Firstly, the cleaned data is divided into two parts: the training portion, and the testing portion. This allows us to ensure that our trained model is able to make predictions based upon new, unseen sample data[22].

The CNN model consists of:

- Convolution layers
- ReLU activation
- Pooling layers
- Fully connected layers

Using these components, during training, the CNN continuously updates its internal weights/biases using an optimizer/loss function in order to reduce prediction error. During training, we monitor the model's performance utilizing accuracy, precision/recall and loss across many training epochs. Once fully trained, we evaluate how well our model performs against a test set before saving the trained model for eventual deployment

4.5.Prediction Module

Flask is a lightweight web application framework built in Python that enables users to build web applications and APIs quickly and efficiently with minimal setup required. For your intrusion detection project, Flask serves as an intermediary between your trained CNN model and end-users. When an end-user visits your website via their browser, Flask presents them with a simple html form in which they can enter the network traffic features associated with their specific scenario.

Once submitted, Flask loads your previously-trained CNN model, processes the entered-in input values, and generates a response with your model's predicted result(s).This way, your machine learning model becomes interactive and accessible via a simple web-based interface rather than simply requiring code

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

execution to run.

5. Algorithm

CNN Algorithm (Convolutional Neural Network)

A Convolutional Neural Network (CNN) is a specific form of Deep Learning Algorithm used to identify and classify features within an input dataset; as such it can be used for both predictive analysis and classification based on extracted feature importance. Instead of working with visual imagery as typical CNN's would, a CNN will learn to find pattern's in your network traffic data.

Steps involved:

- Input preprocessed data
- Apply convolution filters
- Apply activation function (ReLU)
- Perform pooling
- Flatten data
- Fully connected layers
- Output classification (Normal/Attack)

5.1.Input Layer

Input is the original untransformed data. For example in images you have pixel values, or in intrusion detection, you have numeric values representing various traffic characteristics.

5.2.Convolutional Layers

Convolutional layers take advantage of spatial hierarchies by using multiple filters (kernels), which scan through the input data to compute the dot product of input values and filter values at different positions, creating a feature map. During training, each filter learns its own set of weights that enable the filter to identify patterns useful in distinguishing normal traffic from attack traffic.

5.3.Activation Functions

After the convolution process, an activation function (e.g., ReLU) is used to create nonlinearity, enabling the network to learn about complex interactions.

5.4.Pooling Layers

Pool layers (e.g., Max Pooling) reduce the spatial dimensionality of the feature maps, preserving the most relevant information. This results in less computational work needed and provides protection against overfitting.

5.5.Flatten Layer

Once there has been sufficient convolution and pooling done on the input image, the resulting multi-dimensional feature map will be flattened into one-dimensional vector, which can then be passed to the last classifier.

5.6.Fully Connected (Dense) Layers

Fully connected layers are similar to how a traditional neural network behaves; the layers utilize the higher level abstracted features generated earlier to determine what class the input belongs to. The final layer typically uses a softmax activation function (normal/intrusion).

6. Implementation

Modern programming tools, frameworks, and web technologies have been utilized to create an integrated system for identifying and predicting network attacks through machine learning techniques. This integrated toolset will allow for fast and effective data processing, model training, and real time predictions.

6.1.Programming Language

Python was selected as the primary programming language because of its ease-of-use, versatility, and wide range of libraries that provide advanced capabilities for both machine learning and data analytics. Some of the most useful libraries available include:

- **NumPy:** a library designed for high performance multi-dimensional array and matrix operations.
- **Pandas:** a library that creates data structures similar to SQL tables that can be easily manipulated and analyzed.
- **Scikit-Learn:** a library that includes many algorithms for classification and regression tasks.
- **TensorFlow or Keras:** libraries that are used for deep learning applications.

6.2.Development Environment (IDE)

Google Colab and PyCharm were selected as the IDEs for this project:

- Google Colab is being used to train the models, and experiment with different approaches. Google Colab is beneficial because it has cloud based environment that is supported by GPU/TPUs, and therefore can speed up computations related to deep learning. Additionally, it allows developers to

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

share their notebooks and collaborate with each other.
- PyCharm is being used to develop the application code, specifically to integrate the trained model into the web-based front end. PyCharm offers many benefits including; debugging, auto-completion of code, and project management.

6.3.Frontend Technologies

The user interface of the system was created using HTML/CSS:

- **HTML (HyperText Markup Language):** Is a markup language used to describe how to display information on the web page. Specifically, HTML is used to build the structural components of the web page, including input fields for displaying network traffic characteristics.
- **CSS (Cascading Style Sheets):** Is a style sheet language used to control the layout and visual presentation of a document written in a markup language like HTML. In particular, CSS is used to improve the usability and responsiveness of the web interface.

6.4.Backend Framework

The back-end of the system was built using the Flask framework:

- **Flask:** A micro web framework for Python that is ideal for creating small web applications. It will serve as a bridge between the front-end and the back-end providing functionality such as routing user inputs to the appropriate functions within the program, loading the pre-trained machine learning model, performing predictions using the loaded model, returning those results
- The ability to deploy this IDS as a web application makes it possible to interact with users in real-time.

6.5.Machine Learning Libraries

Some of the most important machine learning libraries are:

- **NumPy & Pandas:** Used for manipulating and pre-processing data
- **Scikit-learn:** Used for splitting datasets into subsets, normalizing data, calculating metrics for model evaluations
- **TensorFlow / Keras:** Used for developing and training Convolutional Neural Networks (CNN)

6.6.Dataset

For training and testing purposes the NSL-KDD dataset is being used. This is one of the standard benchmarks used in intrusion detection research. The dataset consists of labeled network traffic data where there are several types of attack vectors. “Using all of these tools together allows us to take advantage of scalability, efficiency in terms of computational resources, and enable real-time deployments of our Intrusion Detection System.”

7. Results And Discussion

The proposed intrusion detection system (ids) based on convolutional neural networks (CNN), which uses the NSL-kdd dataset for performance evaluation. The dataset was randomly partitioned into test and training datasets. This ensures that the test data is unbiased as well as provides a fair comparison of the models.

7.1.Performance Metrics

To assess the effectiveness of the model, the following evaluation metrics were used:

- **Accuracy:** Measures the overall correctness of the model
- **Precision:** Indicates the proportion of correctly predicted attack instances
- **Recall (Detection Rate):** Measures the ability to detect actual attacks

7.2.Experimental Results

The CNN model demonstrated strong performance in classifying network traffic into normal and attack categories. The results show that the model effectively learns complex patterns in the dataset.

- Accuracy: 97.2%
- Precision: 96.8%
- Recall: 97.5%
- F1-Score: 97.1%

These results indicate that the proposed model achieves high detection capability with minimal false positives.

7.3.Comparison with Existing Methods

Compared to traditional machine learning models such as Decision Trees and Support Vector Machines

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

(SVM), the CNN model shows:

- Higher accuracy
- Better generalization to unseen data
- Reduced false alarm rate

This improvement is due to CNN's ability to automatically extract important features from network traffic data without manual intervention. As shown in Figure 3.

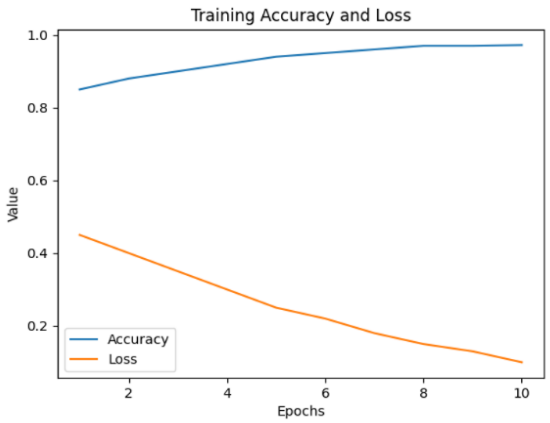

Figure 3 Training accuracy and loss of the proposed model over epochs

7.4.Discussion

The experimental results confirm that deep learning models, particularly CNN, are highly effective for intrusion detection tasks. The model successfully captures both linear and non-linear relationships in the data, leading to improved classification performance.

However, certain challenges remain:

- Imbalanced datasets affect detection of rare attacks (R2L, U2R)
- Training time is higher compared to traditional models
- Real-time deployment requires optimization

Despite these challenges, the proposed system provides a scalable and robust solution for modern network security applications.

Conclusion

This research is a development of an effective machine learning-based method to predict network attack based on the NSL-KDD dataset [6]. The proposed model uses a convolutional neural network (CNN) to develop a complex pattern recognition technique that can be used to identify the traffic in the networks into two main categories namely; Normal and Malicious [6]. The result of the experiment shows that the proposed model obtained very good accuracy, precision and recall for all four type of attacks i.e., DoS, Probe, R2L, and U2R [6]. Moreover, compared to the traditional methods of Intrusion Detection Systems (IDS), the CNN model has better detection capability and lower False Alarm Rate (FAR) because it can automatically extract features from the traffic in the networks [1], [6]. In addition, since the CNN model has been integrated with a flask web server for real time predictions, the proposed model can be effectively deployed in different scenarios of real world. Therefore, the proposed model presents an innovative, efficient and robust way to improve the security of computer networks [6].

Future Enhancements

The proposed system utilizes machine learning to successfully predict attacks on a network. However, this system primarily operates as a static intrusion detection mechanism based upon attributes of network traffic. Therefore, to continue enhancing security in line with current Zero Trust architectural paradigms, the proposed system will be enhanced with continuous authentication mechanisms utilizing user behavior analytics. With the additional method described above, the intrusion detection mechanism can be integrated with an AI-based continuous authentication mechanism. In addition to providing continuous authentication, the proposed mechanism will monitor all user activities including; keystroke dynamics, mouse movement, login activity and session activity. Continuous authentication differs from the traditional one-time authentication paradigm as the proposed mechanism provides verification of user identity during each user session. Therefore, if an attacker is able to obtain access through a single successful login attempt, they may still be denied access at some point within their session. Further, sophisticated deep-learning models including Long Short-Term Memory (LSTM) networks can be used to capture temporal and sequential relationships in user behavior. As such, the proposed mechanism will have the ability to provide

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

real-time anomaly detection and potential identification of inside threats or account takeover attempts. Finally, integration with Zero Trust Network Paradigm ensures that no users or devices are trusted by default. Each access request will be continually evaluated against a user's behavioral pattern(s) and risk score(s) provided by the AI model. As such, the proposed system will have a much greater overall impact on its security posture. Additionally, the proposed system can include:

- Real-time monitoring of user actions and network traffic
- Adaptive Risk-Based Authentication Mechanisms
- Multi-Factor Authentication Integration
- Federated Learning for Privacy-Preserving Model Training
- Scalability in Cloud Environments

Combining the two previously stated approaches will create a comprehensive security solution that detects both external cyber-threats and internal threats. The advancement from passive ML-based detection to an AI driven behavioral authenticating capability demonstrates a substantial advancement toward developing Zero Trust Networks which are both secure and intelligent. "The proposed enhancements will transform the existing passive intrusion detection system into an active, intelligent, and continuous security system aligned with Zero Trust."

Acknowledgement

The authors would like to extend the deepest gratitude to the members of our institution as well as faculty members for the support and guidance that you offered while we completed this research work.
We express our sincere thanks to the mentor and department staff for the guidance, suggestions and support given to complete the project "Smart Cyber Threat Detection and Prediction using Machine Learning" We would like to appreciate the researchers and authors whose published works and research papers helped us to understand cyber security, Intrusion Detection System, Machine Learning, CNN, NSL-KDD Dataset, Network Security. We express our gratitude to friends and family for the motivation and support given during the completion of this work.

References

[1]. C. Lu, "Research on the technical application of artificial intelligence in network intrusion detection system," in Proc. Int. Conf. Electronics and Devices, Computational Science (ICEDCS),2022.
[2]. H. Sivakami V, N. M, R. M. P, and E. S. G. S. R, "Copula GAN Boosted Random Forest based Network Intrusion Detection System," in Proc. 14th Int. Conf. Computing Communication and Networking Technologies (ICCCNT),2023.
[3]. D. Liao, S. Huang, Y. Tan, and G. Bai, "Network Intrusion Detection Method Based on GAN Model," in Proc. Int. Conf. Computer Communication and Network Security2020.
[4]. X. Zhan, H. Yuan, and X. Wang, "Research on Block Chain Network Intrusion Detection System," in Proc. Int. Conf. Computer Network, Electronic and Automation (ICCNEA),2019.
[5]. B.-S. Lee, J.-W. Kim, and M.-J. Choi, "Federated Learning Based Network Intrusion Detection Model," in Proc. Asia-Pacific Network Operations and Management Symposium(APNOMS),2023.
[6]. Y. Zhang et al., "Comparison of machine learning methods for intrusion detection," Simulation, vol. 92, no. 9, pp.861–871,2016.
[7]. B. Wang and J. Pineau, "Online bagging and boosting for imbalanced data streams," journal Trans. Knowledge and Data Engineering, vol. 28, 2016.
[8]. J. K. Jaiswal and R. Samikannu, "Application of Random Forest Algorithm," in Proc. WCCCT, 2017.
[9]. K. Manandhar et al., "Detection of faults and cyber-attacks in smart grid," IEEE Trans. Control of Network Systems, vol. 1, no. 4, 2014.
[10]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Information Systems Security and

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0328
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2288 - 2297

Privacy (ICISSP), 2018.
[11]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in Proc. Symp. Computational Intelligence for Security and Defense Applications, 2009.
[12]. W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in Proc. Int. Conf. Intelligence and Security Informatics, 2017.
[13]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[14]. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in Proc. 9th EAI Int. Conf. Bio-inspired Information and Communications Technologies, 2016.
[15]. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," Trans. Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, 2018.
[16]. G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," Expert Systems with Applications, vol. 41, no. 4, pp. 1690–1700, 2014.
[17]. R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI), 2017.
[18]. Umer, M. A., Junejo, K. N., Jilani, M. T., & Mathur, A. P. (2022). Machine Learning for Intrusion Detection in Industrial Control Systems: Applications, Challenges, and Recommendations. Journal of Information Security and Applications, 65, 103078. doi: 10.1016/j.jisa.2022.103078.
[19]. M. Ring, D. Schlör, D. Landes, and A. Hotho, "Flow-based network traffic generation using Generative Adversarial Networks," Computers & Security, vol. 82, pp. 156–172, 2019.
[20]. K. Kendall and C. Kendall, "Intrusion detection using neural networks and attack classification," in Proc. Int. Conf. Artificial Neural Networks in Engineering, 1999.
[21]. S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in Proc. Journl ref Int. Joint Conf. Neural Networks, 2002.