VOL. 12 NO. 9,
MAY, 2026
E-ISSN 3026-958X
P-ISSN 3027-1169

African Scholar Publications & Research International
www.africanscholarpub.com

Journal of Science Innovation & Technology Research (JSITR)

Performance Benchmarking of ANN and LSTM Architectures for Network Intrusion Detection Using Chi-Square Feature Optimization

Patrick Moses[1]; Oladunjoye John Abiodun[2]; Agu Edward Onyebueke[3]; Andrew Ishaku Wreford[4]; & Madugu Jeremiah Omanga[5]

[1,2,3,4]Department of Computer Science, Federal University Wukari, Nigeria. [5]Taraba State University Jalingo, Taraba, Nigeria
Corresponding Author: patrickmoses292@gmail.com
DOI: https://doi.org/10.70382/ajsitr.v12i9.075

Abstract

The increasing complexity and frequency of cyber-attacks continue to pose significant challenges to network security, thereby necessitating the development of efficient and reliable Intrusion Detection Systems (IDS). To address this challenge, this study proposes deep learning-based Network Intrusion Detection Models (NIDMs) using Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM), integrated with Chi-Square feature selection for improved model efficiency. The study adopts a structured three-phase approach comprising data acquisition and preprocessing, model implementation, and performance evaluation. To ensure robustness and enhance the generalizability of the proposed models, the evaluation is conducted on two distinct benchmark datasets, NSL-KDD and WUSTL, representing different network environments and attack characteristics. This dual-dataset validation framework enables a more comprehensive assessment of model performance beyond single-dataset limitations. Experimental results demonstrate that the ANN-based model achieved accuracies of 99.99% and 99.96% on the NSL-KDD and WUSTL datasets respectively, while the LSTM-based model achieved 99.95% and 96.5% on the same datasets. These findings indicate that the proposed approach not only improves detection accuracy but also maintains consistent performance across heterogeneous datasets, thereby confirming

its effectiveness and practical applicability for real-world intrusion detection scenarios.

Keywords: Network Intrusion Detection, Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), Chi-Square Feature Selection, Cybersecurity.

Introduction

Information and data security form the main priority in any given organization. The emergence and spread of cybercrimes in the past years have threatened and violated the security of the systems, hence affecting the integrity of information and data in the systems (Tarter, 2017). Moreover, with the rise in the use of the internet, the volume of personal, sensitive, and crucial information passing through the internet has significantly risen, hence empowering malicious attempts and intrusions into the network with the aim of accessing crucial information that has the potential to influence the functioning of the systems, and hence compromise data security due to security loopholes in the systems, and the complexity and sophistication of the attacks (Ashfaq et al., 2017). Therefore, cybersecurity has emerged as a significant security feature in future networks, enabling it to react to and counter these threats. However, artificial intelligence, particularly in intrusion detection and prevention, has attracted significant attention in terms of efficiency and simplicity in identifying attacks (Vasilomanolakis et al., 2015). Nevertheless, as technology continues to evolve, various strategies are being devised by cyber-crime offenders to destroy or steal valuable data in an organization.

Intrusions, therefore, can be defined as any activity that leads to damage to the information system. An Intrusion Detection and Prevention System, which is referred to as IDS and IPS respectively, is a software tool used to monitor a network for malicious activities or the violation of policies concerning cybercrime, generating an immediate report to the management system to prevent the occurrence of such events (Vasilomanolakis et al., 2015). A comparison to a firewall, which acts as a gatekeeper at the border of the network to monitor the traffic entering the network or the endpoint and determine whether the traffic should be permitted to enter the network at all, IDS also relate to network security but differ from firewalls since the concern here is intrusions (Sharma & Kaur, 2019). IDS primarily concern themselves with the traffic that is within the network to determine whether there is any suspicious activity. This enables the IDS to detect intrusions that might have escaped the attention of the firewall. Intrusion Detection System is a security management system for devices on a network that collects data from various areas of a computer or a network to identify potential security breaches, which include intrusions as well as misuse of the organization's security, thus optimizing vulnerability assessment to assess the security of a computer system or a network (Prasad & Rohokale, 2020). Therefore, the main purpose of IDS is to monitor the activities on a network or a device to identify any

suspicious activities or known threats and thus generate a warning message on detection of any threat. IDS is able to identify abnormal behavior that cannot be detected by a conventional packet filter.

In the past decade, different solutions based on the application of the Machine Learning and Deep Learning techniques have been proposed by researchers aiming to improve the efficiency of the Network Intrusion and Prevention systems in detecting malicious attacks, considering the tremendous growth in network traffic and the security threats challenges that have been posed by cybercriminals (Yang et al., 2020; Yao et al., 2018; Li et al., 2019). It is worth mentioning that the research related to the application of the DL techniques for the development of the Network Intrusion and Prevention systems is still in its early stage, with tremendous possible solutions.

However, despite all these attempts carried out in the intrusion detection domain, the existing IDS approaches have a high false positive rate and low detecting rate. For mitigating these issues, deep learning, one of the categories of machine learning approaches, is widely utilized to handle different categories of intrusion and security issues in different environments to increase the security of the cyber world, as it can easily model complex architectures of data to carry out different non-linear data transformations and identify some patterns of data with different architectures of neural networks (Ashfaq et al., 2017). Therefore, with the consideration of the continuously increasing diversity and attempts of cybersecurity attacks, it is essential to offer an efficient and effective solution to prevent these attacks. Therefore, to offer network security, an intrusion detection and prevention system using LSTM and ANN is proposed in this study to handle two different network intrusion datasets. The proposed dataset is collected from the Kaggle machine learning repository to predict the different types of IDS attacks.

Despite the significant progress made in intrusion detection systems, many existing approaches continue to suffer from high false-positive rates, which result in unnecessary alerts and increased operational overhead for network administrators. This limitation reduces the practical effectiveness of IDS in real-world deployments, where precision is as critical as detection accuracy. To address this challenge, this study proposes the integration of Chi-Square feature selection with ANN and LSTM models to enhance the discriminative capability of intrusion detection. By selecting only, the most statistically relevant features, the proposed approach aims to reduce noise and redundancy in the input data, thereby improving classification precision and minimizing the likelihood of falsely identifying benign traffic as malicious. Consequently, this research is designed not only to improve detection accuracy but also to reduce false alarm rates, thereby enhancing the reliability and usability of intrusion detection systems in practical cybersecurity environments.

RELATED WORKS

A wrapper-based feature selection model, namely Tabu Search and Random Forest (TS-RF), for the purpose of intrusion detection has been proposed by Nazir and Khan (2021). Tabu

Search has been used for the exploration of attribute subsets, while Random Forest has been used as a learning algorithm for the model. The model has been tested with the UNSW-NB15 dataset, and it has been reported that the model has achieved 83.12% accuracy with a False Positive Rate (FPR) of 3.7%. The model has not considered the imbalance that might occur with the model.

The bi-phase lightweight intrusion detection system proposed by Srivastava and Sinha (2024) uses Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) for feature selection, along with Genetic Algorithm (GA) optimized ensemble classifiers. PSO is used for feature selection based on dataset imbalance in the first phase, while ACO is used for feature extraction based on informativeness and independence in the second phase. GA is then used for optimization of classifiers. Evaluation of the proposed approach using NSL-KDD, UNSW-NB15, and CSE-CIC-IDS2018 datasets showed that XG-Boost had the highest accuracy of 90.38%, 92.63%, and 97.87%, respectively. Statistical validation proved that the proposed approach was better than other approaches.

In another research work, Zhang et al. (2024) proposed a bagging ensemble method for intrusion detection in networks. Extreme Random Trees were used to determine feature importance and feature subset selection. Bootstrap sampling was used, and Classification and Regression Trees, Support Vector Machine, and K-Nearest Neighbor algorithms were combined by the ensemble method. Among all the ensemble algorithms proposed by Zhang et al., KNN-Bagging performed better. Bayesian Optimization was used to tune the hyperparameters of the KNN algorithm. The proposed BO-KNN-Bagging algorithm achieved an accuracy of 82.48% by using the NSL-KDD dataset.

Maddireddy & Maddireddy (2024) have proposed the use of various machine learning-based intrusion detection techniques like the Random Forest model, SVM model, and Decision Tree classifier model for detecting intrusion attacks in the NSL-KDD dataset. The techniques of dimensionality reduction adopted by the author are PCA and t-SNE techniques for improving the computational efficiency of the model. The performance metrics adopted by the author for evaluating the performance of the model are accuracy, precision, recall, F1 score, false positive rate (FPR), false negative rate (FNR), and Area Under the ROC Curve (AUC) for the model. The accuracy of the SVM model is 95%, the precision is 92%, the recall is 94%, and the F1 score is 93% compared to the Decision Tree model, which has an accuracy of 92.7%, and the Random Forest model, which has an accuracy of 94.5%.

While recent studies have made significant progress in applying machine learning and deep learning techniques for intrusion detection, a critical analysis reveals several persistent limitations. Many existing approaches primarily focus on optimizing classification algorithms without adequately addressing feature redundancy and irrelevance, which can negatively impact model efficiency and increase computational overhead. For instance, optimization-based methods such as TS-RF and PSO-ACO frameworks emphasize search efficiency but often lack robust statistical feature selection mechanisms to ensure the relevance of input variables. Similarly, ensemble and traditional machine learning models, including SVM,

Random Forest, and bagging-based techniques, have demonstrated moderate performance improvements but still suffer from issues related to high false positive rates and limited generalization across diverse datasets.

Deep learning models such as LSTM have shown strong capability in capturing sequential patterns in network traffic, their performance is often constrained by high-dimensional input spaces and the absence of effective feature selection techniques. This gap highlights the need for a hybrid approach that combines statistical feature selection with deep learning architectures. In this study, the integration of the Chi-Square (χ^2) feature selection method with LSTM and ANN models addresses these limitations by systematically selecting the most informative features based on statistical dependency, thereby reducing dimensionality and enhancing model interpretability. Furthermore, unlike many previous studies that rely on a single dataset, this research validates the proposed approach on both NSL-KDD and WUSTL datasets, thereby improving robustness and demonstrating stronger generalization across heterogeneous network environments. This combination of statistical feature optimization and deep learning-based modeling represents a more comprehensive and effective solution to intrusion detection challenges than previously reported methods.

RESEARCH METHODOLOGY

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are of major importance when it comes to protecting computer networks from advanced cyber-attacks (Arafat, 2025). However, it is evident that anomaly-based intrusion detection systems face challenges characterized by inconsistent performance, which is attributed to the application of unstructured methodologies and validation data (Zhen et al. 2025). To ensure robustness, reliability, and reproducibility of the study, it is proposed that a structured three-phase research methodology be employed. In the first phase of the study, data acquisition is performed. In this case, data is acquired from the Kaggle machine learning repository using the Python panda's library, which is used to import data from comma-separated value (CSV) files. The next phase is data preprocessing, where complexity is reduced to improve the performance of the model while avoiding increased computational complexity. This phase includes dealing with missing values in the dataset, followed by feature selection through the Chi-Square (χ^2) method to select the most important features in the dataset. This minimizes redundancy and improves the efficiency of the classification process. The last step in the process is the development and evaluation of the models. The preprocessed data is input into the ANN and LSTM models for the purpose of intrusion detection. The performance of the models is evaluated in order to determine the efficiency of the intrusion detection process. The step-by-step implementation of the proposed framework is shown in Figure 1.

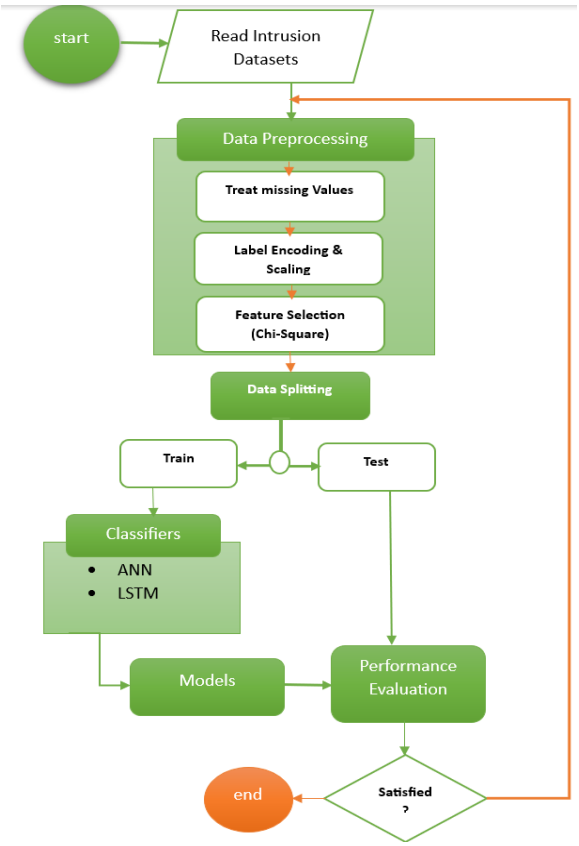


Figure 1: Research Methodology

Dataset Description

This research as aforementioned proposed the evaluation of the LSTM and ANN algorithm on two network intrusion datasets sourced from the Kaggle machine learning repository.

Dataset I (WUSTL-IIOT-2021 Dataset)

The WUSTL-IIoT-2021 dataset was retrieved from the Kaggle website, which contains a variety of machine learning datasets (https://www.kaggle.com/datasets/mostafa2372/wustl-iiot-2021). This dataset contains a total of 1,194,464 records, including 49 features that describe different characteristics of network flows. In this dataset, a key feature labeled as "Target" is identified, which contains values of either "0" or "1." In this dataset, a value of "0" represents 1,107,448 records of normal attacks, which are considered non-security threats, while a value of "1" represents 87,016 records of malicious attacks, which are considered security threats to the IIoT system. The overall goal of this study is to identify records that contain a threat to the system, i.e., records with a value of "1."

Dataset II (NSL-KDD Dataset)

The second dataset is NSL-KDD Dataset. NSL-KDD is a dataset which is developed by deriving data from KDD cup 99. It has been employed by different authors, such as (Li et al.,

2020; Guezzaz et al., 2021; Awotunde et al., 2021), in the development of Intrusion Detection System for IIoT. It has 125,973 entries for the training data and 22,544 entries for the test data. It has 42 features. NSL-KDD data was obtained from GITHUB by the following link *https://www.kaggle.com/datasets/hassan06/nslkdd*. It has one target feature labels with values 0 for non-threat (normal) and 1 for threat (attack).

Data Preprocessing

To enhance the performance and efficiency of the proposed intrusion detection models, appropriate preprocessing techniques tailored to tabular network traffic data were applied. Unlike image-based datasets, the NSL-KDD and WUSTL datasets consist of structured records; therefore, standard tabular data preprocessing methods were employed. First, data cleaning was performed to handle missing or inconsistent values, ensuring data integrity. Next, categorical features such as protocol type, service, and flag attributes were transformed into numerical representations using label encoding and one-hot encoding techniques, enabling compatibility with machine learning models. Subsequently, feature scaling was applied using Min-Max normalization to map feature values into a uniform range [0,1], thereby improving model convergence and preventing bias toward features with larger magnitudes. Following preprocessing, feature selection was conducted using the Chi-Square (χ^2) statistical method to identify the most relevant features for intrusion detection. The Chi-Square test evaluates the dependency between each feature and the target variable, with higher χ^2 scores indicating stronger relevance. To ensure reproducibility and optimal model performance, a fixed feature selection threshold of k = 20 was adopted for both datasets. This implies that the top 20 features with the highest Chi-Square scores were selected for model training. The choice of k = 20 was determined empirically based on preliminary experiments, balancing dimensionality reduction with classification accuracy. This approach reduces redundancy, lowers computational complexity, and enhances the predictive capability of the ANN and LSTM models. Figure 2 presents the data preprocessing steps.

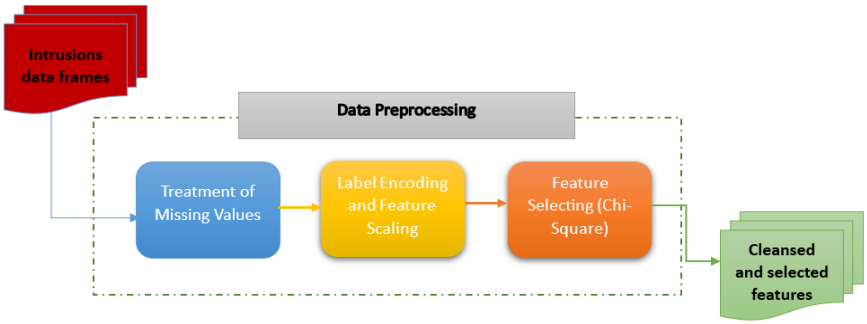


Figure 2: Data Preprocessing

Chi-Square Feature Selection

Feature selection improves the performance of the model and reduces the complexity of the

computation by using only the most important attributes before training the model. The Chi-Square method is a statistical approach that measures the dependency between the target variable and the predictor attribute using categorical data. It measures the difference in the observed and expected values. The Chi-Square statistic is defined as:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \qquad (3.1)$$

where O_i represents observed frequencies and E_i denotes expected frequencies.

In feature selection, variables with higher χ^2 values indicate stronger dependence on the response variable and are therefore considered more informative. Conversely, lower χ^2 values suggest independence and limited predictive relevance. Features with the highest χ^2 scores are selected for model training to improve classification accuracy and efficiency.

Algorithm 3.1: Chi-Square Algorithm

Step 1: Define Hypothesis.
Step 2: Build a Contingency table.
Step 3: Find the expected values.
Step 4: Calculate the Chi-Square statistic.
Step 5: Accept or Reject the Null Hypothesis.

Long Short-Term Memory (LSTM)

LSTM is a variant of Recurrent Neural Network (RNN) designed to address the vanishing gradient problem in sequential learning. It incorporates memory cells regulated by three gates: input gate (i_t), forget gate (f_t), and output gate (o_t). These gates control information flow using sigmoid activation functions.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \qquad (3.2)$$
$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \qquad (3.3)$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \qquad (3.4)$$

The candidate cell state and updates are computed as:

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \qquad (3.5)$$
$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \qquad (3.6)$$
$$h_t = o_t \tanh(c_t) \qquad (3.7)$$

Here, $x_t$represents input at time t, $h_{t-1}$the previous hidden state, and W, $b$denote weights and biases. The gating mechanism enables LSTM to retain long-term dependencies, making it suitable for detecting sequential intrusion patterns.

Artificial Neural Network (ANN)

The ANN model is implemented using a Multi-Layer Perceptron (MLP) architecture consisting of input, hidden, and output layers. Each neuron computes a weighted sum of inputs plus bias:

$$A = B + \sum_{i=1}^{n} w_i x_i \quad (3.8)$$

where $w_i$represents weights, x_i inputs, and B bias. The designed ANN architecture comprises of the with input which comprises of 128 neurons (ReLU activation), with 64 neurons (Dense layer) at the input layer and 1 neuron (Sigmoid activation for binary classification) at the output layer. Backpropagation is applied for training, with a learning rate of 0.0001 and momentum of 0.7. Binary cross-entropy is used as the loss function due to the binary nature of intrusion detection (benign vs. attack).

Activation Functions

ReLU ($f(x) = \max(0, x)$) is applied in hidden layers to introduce non-linearity and mitigate vanishing gradient issues. For the output layer, the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.9)$$

is employed to map outputs into the probability range (0,1), enabling binary classification. The combined use of ReLU and sigmoid ensures efficient gradient propagation and probabilistic interpretation of intrusion detection outcomes.

Evaluation Parameters

For the evaluation of the efficiency level of the developed model, the following metrics use several features. These metrics are True Positive (TP) or Recall (R), True Negative (TN), False Positive (FP), and False Negative (FN).

True Positive Ratio (TPR) or Recall (R): TPR or Recall for a particular class, i.e., the number of samples for the particular class that are correctly predicted, divided by the total number of samples for the particular class, can be expressed as shown in the following formula:

$$TPR = \frac{TP}{TP + FN} \quad (3.10)$$

Accuracy is presented in terms of a percentage. Accuracy is a measure of the extent to which the instances are predicted correctly. It can be computed in the following manner:

$$Accuracy = \frac{TP + FN}{TP + FP + TN + FN} \quad (3.11)$$

Precision is presented in terms of a ratio of the number of decisions, which are considered to be correct. It is presented in terms of TP, out of the sum of FP and TP. It can be computed in the following manner:

$$Precision = \frac{TP}{TP + FP} \quad (3.12)$$

RESULTS AND DISCUSSION

In this section, the results of implementing the proposed network intrusion detection models using the WUSTL and NSL-KDD datasets are discussed. In this regard, deep learning algorithms such as Long Short-Term Memory (LSTM) and Artificial Neural Network (ANN) were developed to detect network intrusions.

Environmental Setup

To develop the network intrusion detection models, the Anaconda environment was utilized to implement Python scripts for network intrusion detection systems on a Windows environment with an Intel Core i5 dual-core processor and 4GB RAM. In this context, TensorFlow was utilized to develop deep learning models, while NumPy was utilized for numerical computations and pandas for data preprocessing. Matplotlib was also utilized to perform visualization for network intrusion detection model performance analysis.

Parameter Setting

Table 1 shows the hyperparameters of the proposed models, i.e., LSTM and ANN models. A batch size of 32 is adopted for efficient weight update and fast convergence during training. Additionally, the maximum number of epochs is set to 10 for both models to ensure sufficient learning of the training data without overfitting. For training purposes, the RMSprop optimizer is used for the proposed LSTM model because of its ability to update the learning rate adaptively, which is efficient for sequential data. On the other hand, the Adam optimizer is used for training the proposed ANN model because of its computational efficiency and fast convergence. In addition, the Mean Square Error (MSE) is used as the loss function to measure the average difference between actual and predicted outputs. Finally, the activation functions used for efficient computation were Hard-Sigmoid and Tanh functions, while ReLU is used to introduce non-linearity.

Table 1. Neural Network (LSTM-ANN) Model Parameter Setting

Parameter	Value
Batch Size	32
Max epochs	10
Optimizer	Rmsprop
Loss	mean square error
Activation	Relu/Tanh/hard-Sigmoid

Feature Selection Result

The Chi Square feature selection technique was used to reduce the dimensionality of the dataset and select the most important features for intrusion classification. This process was regulated by the k-threshold parameter, where the number of the most important features is defined by the threshold. After the experiment, the threshold was set to k=20 for both datasets.

The most important features were selected, where the statistical dependency is high, to improve the efficiency of the classification process and reduce the computational complexity.

Table 2. Selected Features Using Chi-Square

WUSTL Dataset selected dataset features	NSL-KDD Dataset selected dataset features
Sport	Duration
Dport	src_bytes
SrcPkts	dst_bytes
TotPkts	logged_in
DstBytes	num_compromised
SrcBytes	num_root
TotBytes	Count
SrcLoad	serror_rate
DstLoad	srv_serror_rate
Load	rerror_rate
SrcRate	srv_rerror_rate
DstRate	same_srv_rate
Rate	dst_host_count
SIntPkt	dst_host_srv_count
sIpId	dst_host_same_srv_rate
dIpId	dst_host_serror_rate
SAppBytes	dst_host_srv_serror_rate
DAppBytes	dst_host_rerror_rate
TotAppByte	dst_host_srv_rerror_rate
SrcJitAct	Label

Model Results

Table 3 presents the summary of the results obtained by the ANN and LSTM models when tested on the WUSTL and NSL-KDD datasets with a batch size of 32. The ANN model had an accuracy of 99.96% on the WUSTL dataset and 99.99% on the NSL-KDD dataset. The LSTM model had an accuracy of 96.5% on the WUSTL dataset and 99.95% on the NSL-KDD dataset. It is clear that the ANN model outperforms the LSTM model on the WUSTL dataset.

Table 3. Network Intrusion Models Result for the WUSTL dataset

Models	Batch	Optimizer	Accuracy (%) WUSTL	Accuracy (%) NSL-KDD-99
ANN	32	adam	99.96	99.99
LSTM	32	rmsprop	96.5	99.95

Classification and Confusion Matrix Analysis

On the WUSTL dataset from figure 4.1, the LSTM model performed well for Class 0 (Precision - 97.0%, Recall - 99.3%, F1-score - 98.2%) but performed poorly for Class 1 (Precision - 89.1%, Recall - 63.2%, F1-score - 74.0%). From the confusion matrix in figure

4.2, it is evident that there were 1,833 correct predictions for Class 0 with 12 false positives, and 98 correct predictions with 57 false negatives for Class 1.

The ANN model had a good performance and showed good balance for both classes. The Precision for Class 0 was 99.7%, and the Recall and F1-score were 99.8%. The Precision for Class 1 was 97.4%, and the Recall and F1-score were 96.8% and 97.1%, respectively. There were minimal false positives and false negatives, with 4 false positives and 5 false negatives.

```
|: evaluation_model (lstm_model, "LSTM")

63/63 [==============================] - 1s 6ms/step
LSTM Accuracy: 0.9655

 LSTM Classification Report:
              precision    recall  f1-score   support

           0      0.970     0.993     0.982      1845
           1      0.891     0.632     0.740       155

    accuracy                          0.966      2000
   macro avg      0.930     0.813     0.861      2000
weighted avg      0.964     0.966     0.963      2000
```

```
evaluation_model (mlp, "Artificial Neural Network")

Artificial Neural Network Accuracy: 0.9955

 Artificial Neural Network Classification Report:
              precision    recall  f1-score   support

           0      0.997     0.998     0.998      1845
           1      0.974     0.968     0.971       155

    accuracy                          0.996      2000
   macro avg      0.986     0.983     0.984      2000
weighted avg      0.995     0.996     0.995      2000
```

Figure 3: LSTM & ANN WUSTL dataset classification Report

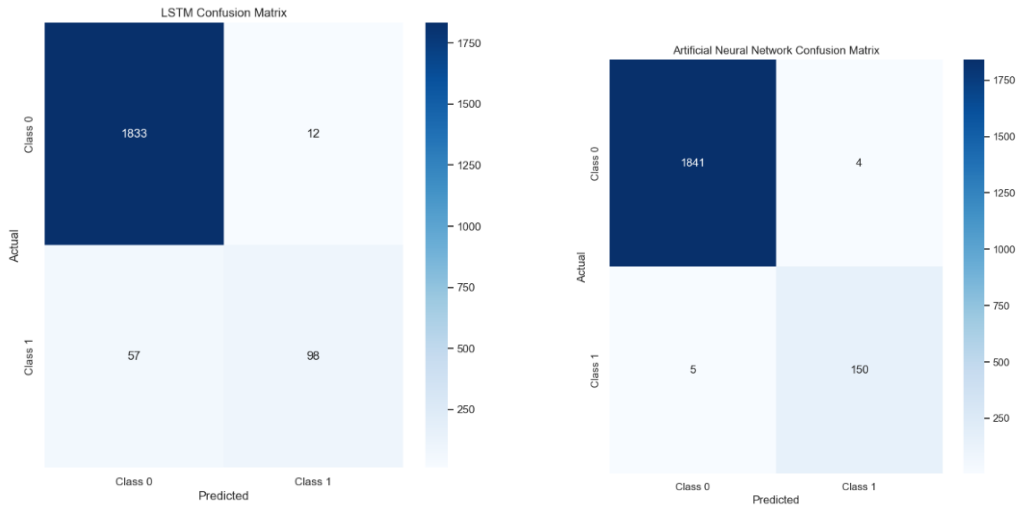


Figure 4: LSTM & ANN WUSTL dataset Confusion Matrix Report

On the NSL-KDD dataset of figure 3, the performance of both models was great. The Precision, Recall, and F1-score of the LSTM model were almost 99.99%. The model misclassified just one data point, as is clear from the confusion matrix. The Precision, Recall, and F1-score of the ANN model were 99.99%, and there were no false classifications. The number of true negatives and true positives was 1,035 and 965, respectively.

```
: evaluation_model (lstm_model, "LSTM NSL-KDD-99")

63/63 [==============================] - 0s 3ms/step
LSTM NSL-KDD-99 Accuracy: 0.9995

 LSTM NSL-KDD-99 Classification Report:
              precision    recall  f1-score   support

           0      1.000     0.999     1.000      1035
           1      0.999     1.000     0.999       965

    accuracy                          1.000      2000
   macro avg      0.999     1.000     0.999      2000
weighted avg      1.000     1.000     1.000      2000
```

```
evaluation_model (mlp, "Artificial Neural Network NSL-KDD")

Artificial Neural Network NSL-KDD Accuracy: 1.0

 Artificial Neural Network NSL-KDD Classification Report:
              precision    recall  f1-score   support

           0      1.000     1.000     1.000      1035
           1      1.000     1.000     1.000       965

    accuracy                          1.000      2000
   macro avg      1.000     1.000     1.000      2000
weighted avg      1.000     1.000     1.000      2000
```

Figure 5: LSTM & ANN Classification Report on NSL-KDD Dataset

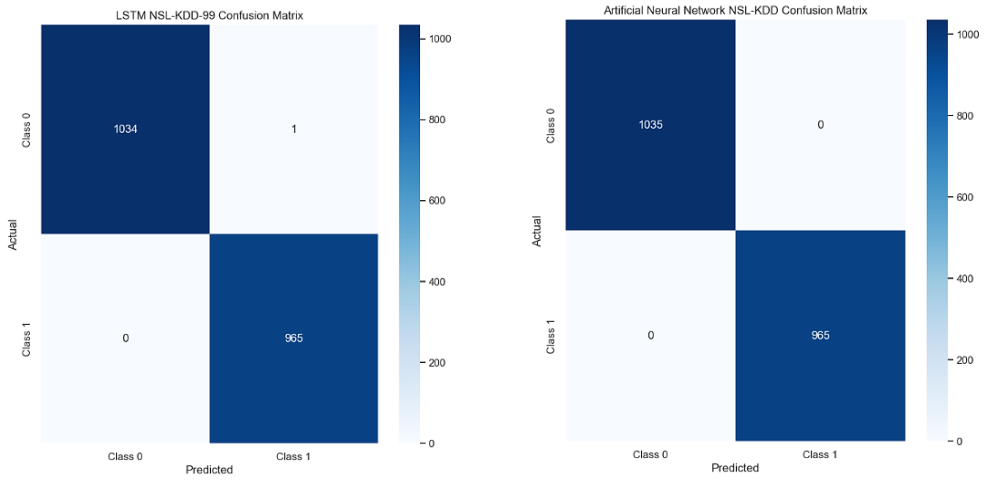
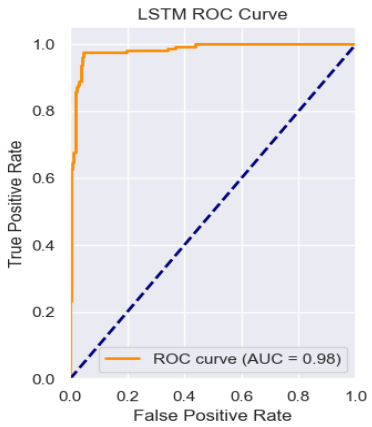


Figure 6: LSTM & ANN Confusion Matrix Report on the NSL-KDD Dataset

ROC and AUC Analysis

The Receiver Operating Characteristic (ROC) curve analysis also validated the robustness of the models. Both the ANN and the LSTM models had an AUC score of approximately 0.98-1.00 on both datasets as shown in figure 7 through 8. The ANN model had a perfect score on the AUC for both datasets.



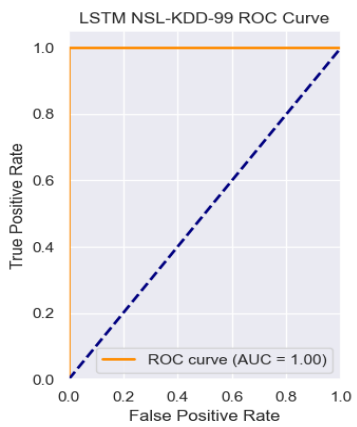
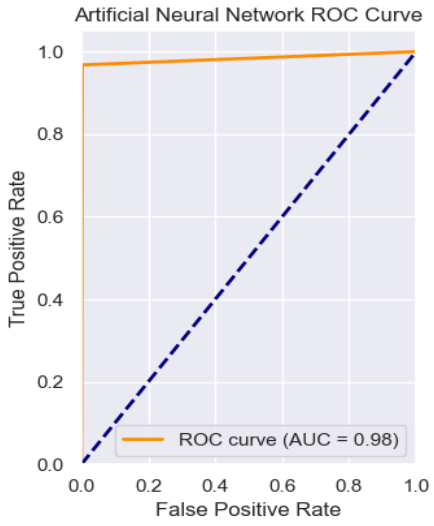


Figure 7 (a): LSTM (NSL-KDD) **Figure 7 (b)**: LSTM (WUSTL) ROC and AUC Report



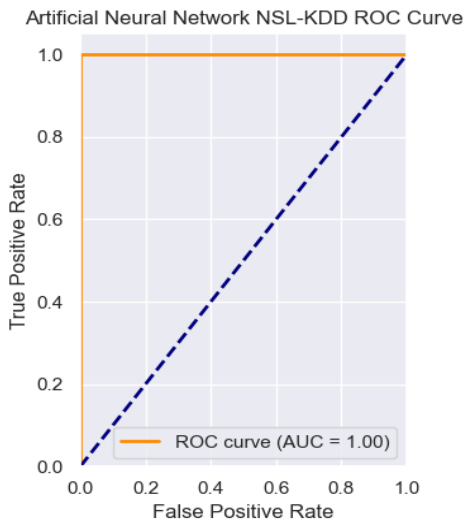


Figure 8 (a): ANN (NSL-KDD) **Figure 8 (b)**: ANN (WUSTL) ROC and AUC Report

State-of-the-Art Comparison

The proposed model based on ANN and LSTM as shown in table 4 has been compared with various recent research articles related to intrusion detection. The proposed model based on ANN has achieved better performance compared to other recent research articles. The proposed model has achieved near-perfect classification metrics with the NSL-KDD dataset. The proposed model based on deep learning has achieved better performance compared to traditional machine learning and ensemble methods.

Table 4. State-of-the-Art Comparison

Study	Dataset	Model	Accuracy (%)
Nazir & Khan (2021)	UNSW-NB15	TS-RF	83.12
Srivastava & Sinha (2024)	NSL-KDD	XGBoost	90.38
Zhang et al. (2024)	NSL-KDD	BO-KNN-Bagging	82.48
Maddireddy & Maddireddy (2024)	NSL-KDD	SVM	95.00
Current Study (LSTM)	NSL-KDD	LSTM	99.95
Current Study (ANN)	NSL-KDD	ANN	99.99
Current Study (ANN)	WUSTL	ANN	99.96

While the proposed ANN model achieved a near-perfect accuracy of 99.99% on the NSL-KDD dataset, this result warrants careful interpretation to rule out potential overfitting or data leakage. It is important to note that the NSL-KDD dataset provides predefined and strictly independent training and testing splits, which were adhered to in this study to ensure that the test data was not exposed during model training. Therefore, the likelihood of data leakage is

minimized. The high accuracy can be attributed to several factors, including the effectiveness of the Chi-Square feature selection method in eliminating irrelevant and redundant features, thereby enhancing the discriminative power of the input data, as well as the relatively structured and less noisy nature of the NSL-KDD dataset.

However, to provide a more realistic assessment of model generalizability, the performance of the models was also evaluated on the WUSTL dataset, which represents a more complex and heterogeneous industrial IoT environment. Unlike NSL-KDD, the WUSTL dataset contains a larger volume of records and more diverse traffic patterns, making intrusion detection more challenging. The comparatively lower performance observed on this dataset, particularly for the LSTM model (96.5% accuracy), highlights the practical limitations of the models when applied to real-world scenarios. This contrast suggests that while the proposed models perform exceptionally well on benchmark datasets, their performance may vary depending on data complexity and distribution. Therefore, the dual-dataset evaluation demonstrates that the proposed approach maintains high accuracy while exhibiting reasonable generalization capability. The slight performance variation across datasets provides a more nuanced understanding of the model's strengths and limitations, reinforcing its applicability for real-world intrusion detection while acknowledging the need for further optimization in highly dynamic network environments.

CONCLUSION AND RECOMMENDATIONS

This study proposed and evaluated deep learning-based Network Intrusion Detection Models (NIDMs) using Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) architectures integrated with Chi-Square feature selection. The results demonstrated that the proposed approach achieved high detection accuracy across both NSL-KDD and WUSTL datasets, with the ANN model outperforming the LSTM model in overall classification performance. The application of Chi-Square feature selection significantly improved model efficiency by reducing feature dimensionality while retaining the most informative attributes. While the findings confirm the effectiveness of the proposed models in detecting network intrusions, this study also highlights the importance of evaluating such models across heterogeneous datasets to ensure robustness and generalizability. The variation in performance between NSL-KDD and WUSTL datasets underscores the need for practical deployment considerations in dynamic and real-world network environments.

To enhance the practical applicability of this research, the following recommendations are proposed:

i. **Deployment in Edge Computing Environments:** The proposed models can be adapted for deployment in edge-based intrusion detection systems, particularly in Industrial Internet of Things (IIoT) environments where real-time threat detection is critical. Lightweight optimization of the ANN model, in particular, makes it suitable for resource-constrained edge devices.

ii. **Integration into Real-Time Network Monitoring Systems:** Future implementations should focus on integrating the proposed models into real-time network traffic monitoring frameworks to evaluate their performance under live data streams and dynamic attack scenarios.

iii. **Adversarial Robustness Evaluation:** Further research should investigate the resilience of the proposed models against adversarial machine learning attacks, such as evasion and poisoning attacks, which can compromise IDS performance in practical deployments.

iv. **Cross-Dataset and Cross-Domain Validation:** To strengthen generalizability, future studies should evaluate the models across additional datasets and diverse network environments, including cloud, SDN, and IoT-based infrastructures.

These practical directions bridge the gap between experimental evaluation and real-world deployment, positioning the proposed framework as a viable solution for next-generation intelligent intrusion detection systems.

DECLARATIONS

Availability of Dataset:

https://www.kaggle.com/datasets/mostafa2372/wustl-iiot-2021
https://www.kaggle.com/datasets/hassan06/nslkdd

Competing Interests: The authors declare that they have no competing interests.

Authors' Contributions: The manuscript was written by Patrick Moses, Dr. Agu Edward Onyebueke, and Madugu Jeremiah Omanga, while the code was written by Andrew Ishaku Wreford, and the entire research was carried out under the supervision of Dr. OladunJoye John Abiodun. It should be noted that all authors contributed substantially to the study.

Funding: No funding was received for this research.

Acknowledgement: This study was supported by the Department of Computer Science, Federal University Wukari, Taraba State.

REFERENCES

Abdiyeva-Aliyeva, G., Hematyar, M., and Bakan, S. (2021). Development of a System for Detection and Prevention of Cyber Attacks Using Artificial Intelligence Methods. 2021 2nd Global Conference for Advancement in Technology (GCAT), 2021, 1-5, DOI: 10.1109/GCAT52182.2021.9587584.

Ali, L. (2019). Cybercrimes-A constant threat to the business sectors and their growth (A study of the online banking sectors in GCC). *The Journal of Developing Areas*, 53(1).

Arafat, M. Y. (2025). Implementation of IDS and IPS for Network Security in IT Infrastructure: Design, Configuration, and Evaluation of Intrusion Detection and Prevention Mechanisms.

Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H., and He, Y.L., (2017). Fuzziness-based semi-supervised learning approach for the intrusion detection system. *Journal of Information Science,* 378, 484–497.

Li, J., Qu, Y., Chao, F., Shum, H.P., Ho, E.S., and Yang, L. (2019). Machine learning algorithms for network intrusion detection. *AI in Cybersecurity*, 3 (2), 151-179.

Maddireddy, B. R., & Maddireddy, B. R. (2024). A Comprehensive Analysis of Machine Learning Algorithms in Intrusion Detection Systems. Journal Environmental Sciences And Technology, 3(1), 877-891.

Nazir, A. and Khan, R.A. Khan. (2021). A novel combinatorial optimization-based feature selection method for network intrusion detection. Comput. Secur., 102.

Prasad R, and Rohokale V. (2020). Artificial intelligence and machine learning in cyber security. *Cyber Security: The Lifeline of Information and Communication Technology*, 2(4), 231-247.

Rakesh, G. (2015). The theory of Intrusion Detection and Prevention Systems (IDS/IPS). https://www.linkedin.com/pulse/theory-intrusion-prevention-detection-systems-idsips-rakesh-gogane/

Sharma, M., & Kaur, S. (2019). Cyber-crimes becoming a threat to cyber security. *Academic Journal of Forensic Sciences,* 6(2), 171-179, ISSN, 2581, 4273.

Srivastava, A., & Sinha, D. (2024). PSO-ACO-based bi-phase lightweight intrusion detection system combined with GA optimized ensemble classifiers. Cluster Computing, 1-56.

Tarter A. (2017). Importance of cyber security. Community Policing-A European Perspective: Strategies, Best Practices, and Guidelines. *Springer Journals*, 2(4), 213-230.

Vasilomanolakis, E., Karuppayah, S., Mühlhäuser, M., and Fischer, M. (2015). Taxonomy and survey of collaborative intrusion detection. *ACM Comput Surv*., 47(4), 1-33. https://doi.org/10.1145/2716260.

Yang, Y., Zheng, K., Wu, B., Yang, Y., and Wang, X. (2020). Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. *IEEE (*8),42169-42184. https://doi.org/10.1109/ACCESS.2020.2977007.

Yao, H., Fu, D., Zhang, P., Li, M., and Liu, Y. (2018). A novel multilevel semi-supervised machine learning framework for the intrusion detection system. *Internet of Things Journal,* 6(2),1949-1959. https://doi.org/10.1109/JIOT.2018.2873125.

Zhang, Z., Kong, S., Xiao, T., & Yang, A. (2024). A Network Intrusion Detection Method Based on Bagging Ensemble. Symmetry, 16(7), 850.

Zhen, L., Kamarudin, N. H., Kok, V. J., & Qamar, F. (2025). Anomaly detection model in network security situational awareness based on machine learning: Limitation, techniques, future trends. IEEE Access.