International Journal of Scientific Research in Computer Science and Engineering
Vol.**13**, Issue.**3**, pp.**01-12**, June **2025**
E-ISSN: 2320-7639
Available online at: www.isroset.org

Research Article

DeepGuard: A Hybrid RF-CNN-GRU Framework for Adaptive Real-Time Zero-Day Network Intrusion Detection

Asfa Praveen[1]

[1]Dept. of Computer Science, College of Engineering and Computer Science, Mustaqbal University, Buraidah, Al Qassim, Saudi Arabia

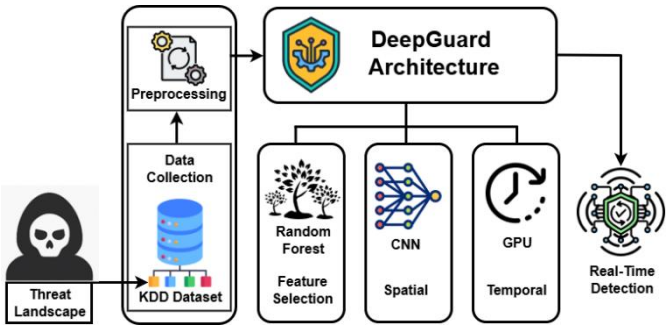
Corresponding Author: ✉

Received: 20/Apr/2025; **Accepted**: 22/May/2025; **Published**: 30/Jun/2025. | **DOI**: https://doi.org/10.26438/ijsrcse.v13i3.687

Abstract— Due to the dynamic and ever-increasing nature of cyber threats, the traditional network intrusion detection systems (NIDS) are not sufficient and are slow to respond to the adapting of new threats since they utilize rigid rules and signature-based detection techniques. This research study aims to design an intelligent adaptive intrusion detection system capable of precise and prompt detection of both established and latent threats to the network. Achieving optimal results from feature extraction, spatial pattern recognition, and temporal sequence learning requires a hybrid model that synergistically combines Random Forest, Convolutional Neural Network, and Gated Recurrent Unit (GRU). The model was built in Python using TensorFlow and Scikit-learn, training and testing it with the KDD Cup 1999 dataset. The experimental results indicated that the proposed model outperformed existing models, achieving an accuracy of 99.23%. This finding confirms the effectiveness of integrating multiple deep learning models. The research data illustrates how the models effectively resolve a basic cybersecurity challenge during active performance.

Keywords—Network Intrusion Detection, Artificial Intelligence, Random Forest, Convolutional Neural Network, Gated Recurrent Unit.

Graphical Abstract—



1. Introduction

It has become more challenging to protect sensitive information from misuse, access, or cyber activities in today's world. The rise in attacks has made signature-based and rule-based Intrusion Detection Systems (IDS) obsolete [1]. Due to their reliance on static frameworks, these systems are incapable of accommodating new and evolving threats, making them more susceptible to zero-day exploit attacks and advanced persistent threats [2]. To address this critical gap, recent research has proposed using Artificial Intelligence (AI) methodologies capable of autonomously learning from data, recognizing unusual changes, and continually adjusting to new patterns in network activity [3].

This research was motivated by the necessity of intelligent, self-adapting intrusion detection systems capable of functioning in high-traffic and dynamic settings. We propose a hybrid AI solution that combines Random Forest with CNN and GRU models. Random Forest uses ensemble learning and explainability features, while CNN efficiently grabs local spatial information from connections attributes, and GRU can learn the temporal relationships in sequences of network behavior. Together, these models constitute an end-to-end detection system that learns complex patterns, identifies and counters adapting attack strategies, and responds forcefully to changing aggressive tactics.

This research is valuable, that may help alleviate the conflicts between real-time adaptive network security and in the detection systems. This work aims to design a model which combines classical machine learning with complex deep learning networks to address the issues of existing models

with the lack of scalability, learning, and responsiveness. Moreover, the research provides a methodology for automated threat elimination which serves the purpose of building a more proactive defense system. A comprehensive model building, testing, and refinement approach is taken in this work to provide a flexible solution to the evolving challenges in cyber intrusion detection.

1.1 Problem Statement

With the exponential growth of sophisticated cyber threats, traditional Network Intrusion Detection Systems (NIDS), which predominantly rely on static, signature-based, or rule-based methods, are increasingly becoming ineffective. These conventional systems struggle to detect zero-day attacks and advanced persistent threats, as they lack adaptability, scalability, and the capability to process high-dimensional and imbalanced network data. Moreover, existing models often fail to capture the temporal and spatial behavior of network traffic, leading to poor detection accuracy and high false positive rates. In addition to that, the most current models either cannot process high-dimensional data or cannot catch temporal and spatial patterns in network traffic [4].

This growing gap necessitates a robust, intelligent, and adaptive solution for real-time intrusion detection. This research overcomes the limitations by introducing a hybrid artificial intelligence-based NIDS that combines RF, CNN, and GRU models. The aim is to improve intrusion detection accuracy, minimize false positives, and maximize real-time responsiveness by combining the strengths of each model in feature selection, spatial analysis, and temporal sequence learning.

1.2 Objective of the Study

The objective of this research study is given as follows:

- Take advantage of the feature selection capability and spatial pattern recognition power of Random Forest (RF) and the temporal sequence learning capability of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) to make out a hybrid and adaptive NIDS model.
- Increase the detection against known and zero-day attacks with few false positives and better real-time performance.
- Research on how to overcome impediments of class imbalance and high dimensions in intrusion detection through good preprocessing and model training solutions.
- Carry out an experiment using the benchmark KDD Cup 1999 dataset to test the performance of the proposed model and prove that it is better compared to traditional models, individual models in accuracy, precision, recall, and robustness.

1.3 Research Contributions

This paper comes up with the following contributions:

- Proposes a new deep hybrid intrusion detection framework (DeepGuard) to identify and adaptively detect intrusions on the network in real-time using RF, CNN and GRU.
- Integrated spatial and temporal feature extraction with the aim of enhancing the model on how it identifies ingrained and variable attack patterns.
- Illustrates the benefit of interpreting-ability in Random Forest, keeping up the highly learning capacity of deep learning models.
- The results are based on the KDD Cup 1999 dataset, a large-scale, imbalanced test sample. The model achieved a high level of detection accuracy (99.23) and AUC (1.00), which exceeds both traditional and traditional machine learning-based techniques.
- Proposes a way to scale, explain and deploy AI model-based intrusion detection systems to the edge with AI model fusion.

1.4 Organization of the Article

The rest of the paper after this first section of Introduction will be structured as follows: section 2 will discuss literature and related works done in the context of intrusion detection systems through the use of artificial intelligence techniques. Section 3 details the methodology presented: how the data is going to be prepared, how the architecture of the hybrid model is going to look, and what are the specifics of the implementation. The details are outlined in section 4 for results and discussions, where the results of the experiment are discussed with attention paid to performance and comparison with the available models. Lastly, Section 5 concludes the paper, and gives possible ways to carry this research further by focusing on real-time application, training transfer, and a more expendable model.

2. Related Works

Vanin et al.[5] discusses the application of AI and ML in augmenting NIDS. It classifies different machine learning methodologies, such as supervised, unsupervised, and hybrid models, and assesses their performance in intrusion detection. It highlights feature selection and the utilization of varied datasets for model training. The research also mentions the challenges of recognizing zero-day attacks and the necessity of continuously updating models to adapt to evolving threats. However, the research study points out these issues regarding outdated datasets and their ineffectiveness in recognizing specific types of attacks due to class imbalance as limitations. Susilo, Muis, and Sari [6] suggest hybrid DL technique for intrusion detection, applied on a portion of the CIC IoT 2023 dataset. The model focuses on maximizing detection accuracy with different types of attacks. Realistic datasets to train effective IDS models are made more important through this study. Yet, limitations such as over representation of particular types of attacks and absence of real-world noise in the dataset may impact the generalizability of the model.

Sajid et al.[7] proposed a hybrid model for better intrusion detection. The paper highlights the robustness of the model and how it can efficiently deal with complicated network traffic patterns. It also explores measures for tackling issues such as class imbalance and overfitting. However, the paper lists limitations such as increased computational needs,

possible latencies in real-time scenarios, and complexity in interpreting the model.

da Silva Ruffo et al. [8] survey discusses the use of DL methods in anomaly and intrusion recognition in SDNs. It discusses several DL models and performance measures. The research also indicates knowledge gaps and proposes future directions. Nevertheless, it highlights some limitations such as the requirement for more extensive datasets and the difficulty in hyperparameter tuning for the best performance.

The Neupane et al. [9] survey analyses the changes incorporated in AI explainability techniques in Intrusion Detection Systems (IDS). It also explores various ways of clarifying extricating reasoning from an IDS model, which is vital for making decisions in cybersecurity operations. The work mentions advantages and disadvantages of explainability against explainability and model performance. Nevertheless, it recognizes challenges like the absence of standardized metrics for measuring explanations and the challenge of adapting explanations to various stakeholders.

Shukla et al. [10] explored how IoT networks need reliable security mechanisms to protect against developing threats. This study designs energy-friendly AI frameworks for intrusion detection employing Bayesian Networks, Artificial Neural Networks (ANN), and Support Vector Machines (SVM). The data captured for training these models includes normal and malicious activities in IoT ecosystems. A three-layered ANN was benchmarked with industry parameters such as real-world roundtrip time and power consumption metrics. The intent focuses on enhancing threat pre-emption strategies in data-sensitive IoT systems, thus fortifying machine-intelligence-driven network resilience.

Recent developments show how machine learning and AI are significant for sophisticated cybersecurity solutions. Narayanan et al. [11] implemented distributed control protocols for securing fractional-order multi-agent systems cyberattack enabling resilient consensus under multiple threats. Kumari et al. [12] also enhanced the workings of neural networks using pruning strategies which are critical for compressing models with limited resource availability. In the context of IoT, Anitha et al. [13] used big data analytics to improve network resiliency whereas Ahamad et al. [14] further enhanced activity analysis for the monitoring of suspicious activities by utilizing machine learning based pattern recognition for more in-depth activity analysis.

Also, Jayaraman et al. [15] developed AI-based multi-cloud robust security frameworks to mitigate infrastructures security risks in hybrid environments. Altogether, these studies demonstrate how AI anticipates emerging challenges with fundamentally efficient and scalable approaches.

3. Proposed Method for Network Intrusion Detection

This paper concerns the development of an AI-based model that can automatically detect and classify different types of network attacks. It initiates with gathering and sanitizing the dataset, which is then utilized with the model. The last step consists of measuring the model's performance with relevant metrics to determine its success. The flow of the methodological approach is demonstrated in figure 1.

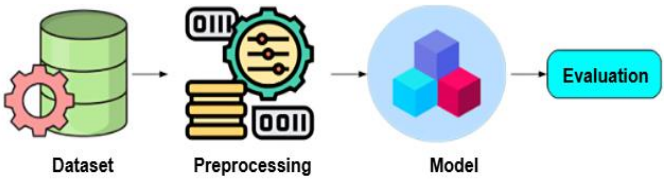


Figure 1. Overall Methodology

The entire research process is shown in Figure 1 with the steps including data acquisition, preprocessing of the same, and then testing the machine learning models trained to get the required results. Then, it introduces the integration of Random Forest to do feature selection, CNN to do spatial feature extraction and GRU to do temporal sequence learning. The flow ends on performance evaluation. This diagram illustrates how the proposed DeepGuard intrusion detection system is systematically built, containing parts that are easily defined and validated.

3.1 Data Collection

The KDD Cup 1999 data set is a standard data set commonly used for the development and testing of intrusion detection systems [16]. It was developed for use of a military network environment in which a broad range of different types of intrusion attacks are made. The database consists of around 5 million connection records, all of which are marked as normal or one of 22 categories of attacks categorized into four classes. Every linking record is described by 41 features that reflect behavior and properties of network traffic. These features are divided into three broad categories. The data is unbalanced, with DoS attacks dominating, presenting a chance to evaluate models on detecting both frequent and infrequent attack types. The rich labeling and detailed attribute set make the KDD dataset perfect for training machine learning models to differentiate between normal and malicious network activity.

3.2 Data Preprocessing

In designing an efficient intrusion detection system, the data preparation phase is very crucial in ensuring the excellence and relevancy of the input data. It begins with a data collection step, and in a way, the raw data consists at first only of relevant materials, but after gathering them, the data usually contains noise, and as such, must undergo a cleansing process by removing missing values, duplicates or outliers. Noisy data confounds results, so data cleansing increases the accuracy of models in terms of learning algorithms. Following that, all the numerical features need to be scaled to ensure uniformity, which is more specifically referred to as data normalization, especially since attributes can have tremendously different ranges. It helps improve the performance of learning algorithms without undue dominance from one attribute over others because the scale is measured. Once preprocessing is complete, the data set is split into test

and training sets. The training set is utilized to train the model to discover patterns, while the test set verifies its presentation on unseen data. Sincere preparation of the data set in such a manner ensures the ensuing intrusion detection model to be accurate and transferable to real network settings.

3.3 Random Forest

RF was selected in this research because of its strength in being able to generalize strongly and perform accurately, even when under robustness tests. RF builds many DT and combines their outputs to increase the confidence of classification and avoid overfitting—usual problems in security datasets. It corresponds to the nature of network traffic where types of attacks may be infrequent or uneven in their occurrence and distinctly manage such feature sets with skewed data. It also provides hints of feature utility which help augment explanation and refinement of the intrusion detection system. Its flexibility is based on the fact that it can be retrained with new data periodically to learn emerging attack patterns. This renders it a scalable and practical solution to identifying known and unknown threats within dynamic cybersecurity landscapes. The trade-off between interpretability and performance in the model justifies real-world usage in resource-constrained systems. Figure 2 represents the operation of a Random Forest model, where several DTs are trained on various subsets of the data set.

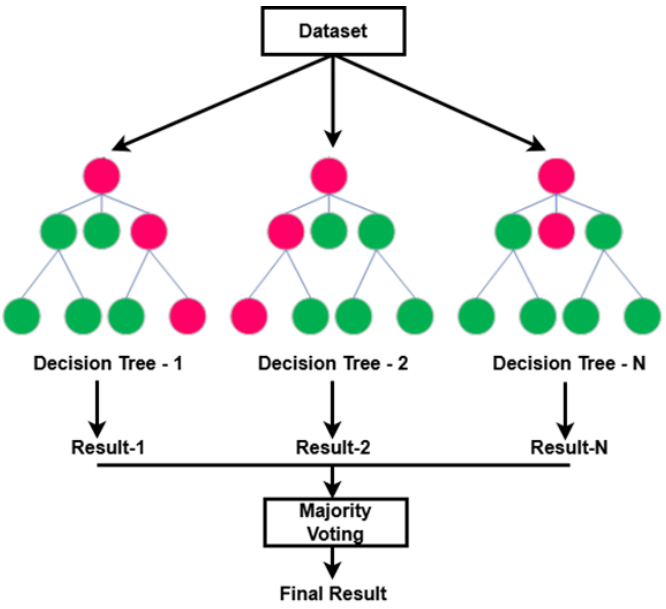


Figure 2. Random Forest

Figure 2 illustrates internalization of Random Forest classifier working that collects the decisions between Decision Trees to be more accurate and robust. Each of the trees is only trained using a particular subset of the data and the overall prediction is based on the majority of the vote. The ensemble method is very useful in reducing overfitting and facilitates models in terms of interpretability since it gives importance to features. It is also able to handle data launching since it applies the structure to unbalanced data, which fits well in the area of intrusion detection. The ensemble method supports better accuracy and fewer overfits. The equation is given in eqn. (1).

$$\hat{y} = mode(h_1(x), h_2(x), \ldots, h_1(x)) \quad (1)$$

Where, $h_i(x)$ is the prediction of the i^{th} tree.

3.4 Convolutional Neural Network

CNNs are a category of deep models greatly capable of capturing spatial features in structured data, which is ideally suited for tasks of network intrusion detection. Here, CNNs are used to process and learn from network traffic patterns as multidimensional feature maps. With convolutional and pooling operations, CNNs can encode complex local dependencies and hierarchical representation of attack patterns. This allows the model to recognize subtle anomalies that other models may not pick up.

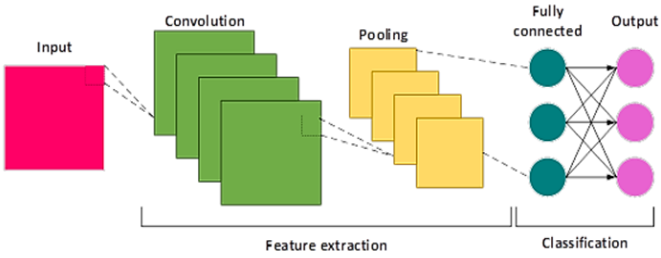


Figure 3. CNN Architecture

CNNs themselves might be unable to capture temporal patterns in sequential network data and may need complementary models. Figure 3 presents an architecture of the Convolutional Neural Network (CNN) that was employed to identify the spatial patterns in the network traffic records. It has the convolutional layer, which detects local patterns, a dimensionality reduction layer (pooling layer), followed by a layer of completely connected layers, which does the final classification. This architecture helps this model to identify minute differences in network operation that could indicate attacks. The CNN part of DeepGuard is to extract features and present them with meaningful information and then infer as a pass to the other temporal modeling layers. Fully connected layers subsequently classify according to extracted features. Convolution operation is given in eqn. (2).

$$Z_{ij} = \sum_m \sum_n X_{i+m,j+n} \cdot W_{m,n} + b \quad (2)$$

Where X is input, W is the filter, and b is bias.

3.5 Gated Recurrent Units

GRUs have been engineered to process sequential data in an efficient manner and overcome vanishing gradient problems. When applied to network intrusion detection, GRUs are suitable for learning temporal relationships in network traffic so that the system can learn patterns through time and detect changing threats. Unlike LSTMs, their decreased structure allows for faster training without significant performance sacrifices, making them more suitable for real-time applications. GRUs adapted to changing traffic patterns, enhancing their suitability for more accurate identification of sophisticated and new attacks while requiring fewer processing resources.

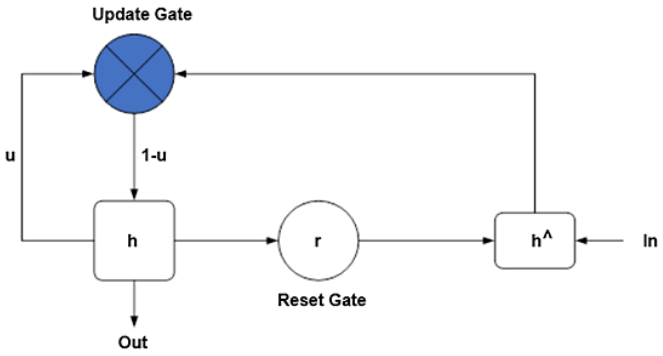


Figure 4. GRU Architecture

In Figure 4 the architecture of the Gated Recurrent Unit (GRU) is shown, which is also a unit constructed to represent temporal dependences on sequential data. The diagram illustrates the update gate and reset gate which govern the memory stream across time-steps. Considering the fact that these are computationally efficient, GRUs will be suitable in detecting intrusions in real time because they can hold relevant historical information. In DeepGuard, GRU analyses time-based traffic patterns, which makes the model capable of identifying dynamic threats more effectively. This process allows effective learning of temporal dependencies in sequential data. The hidden state in a GRU is updated using eqn. (3)

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (3)$$

Where, z_t update gate, $\tilde{h}_t$ candidate activation, $\odot$ element-wise multiplication.

4. Results and Discussion

The suggested hybrid RF–CNN–GRU model performed outstandingly, with high accuracy, precision and recall across classes. The model performed well in class imbalance and outperformed individual models. Training was stable, and the validation loss was low, reflecting strong generalization and stability in identifying varied intrusion patterns.

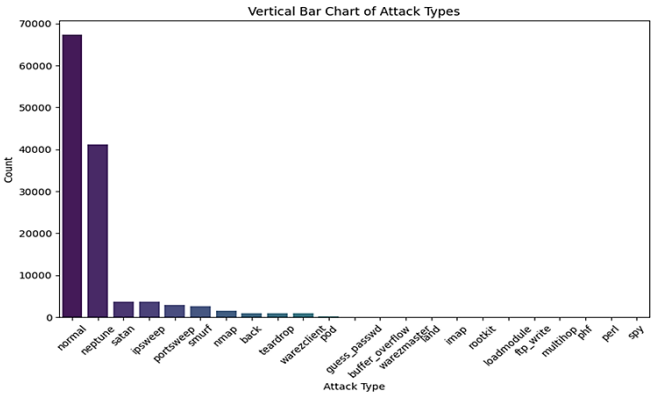


Figure 5. Attack Types Vertical Chart

In Figure 5, there is an illustration showing how the frequencies of different types of network attacks have been recorded and it is clear that there is a large class imbalance. This imbalance is problematic for training models, as preprocessing and design models have to be crafted very carefully in order to capture all of the so-called minority attacks – which, in this case, achieves the goal of a flexible and resilient intrusion detection system as per the specification.

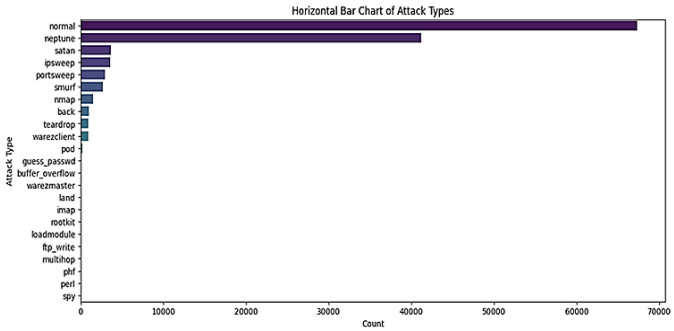


Figure 6. Attack Types Horizontal Chart

Figure 6 contains the same data of frequency of attacks as Figure 5 but in a horizontal design. Visualization once again highlights how a small percentage of forms of attacks predominate and the miniscule percentage of most of the other forms of attack proves the existence of imbalance in classes. Through this figure, there is clear visual evidence that we need intelligent models, which can detect low-frequency intrusions effectively in the same manner as they detect common intrusions. It lends credence to the ability of DeepGuard to deliver even coverage of all forms of attacks.

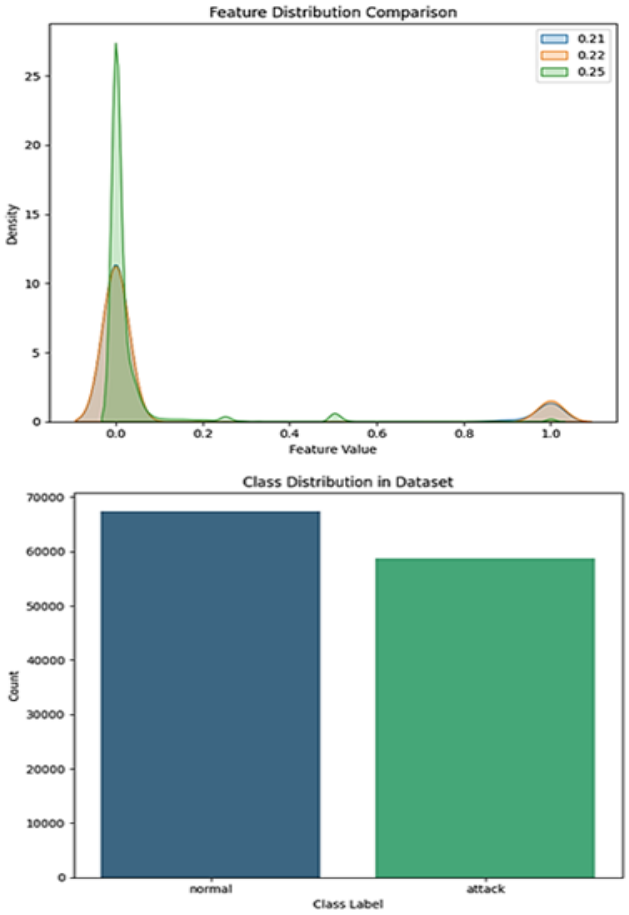


Figure 7. Distribution Comparison

Figure 7 shows two important visualizations. The left plot compares distributions of features across various values (0.21, 0.22, 0.25), suggesting that the majority of features are bunched up around zero, pointing towards sparsity in the data. The right bar plot indicates class distribution, which shows a minor imbalance with more "normal" samples than "attack" samples. It has two plots, a histogram plot of feature distribution and a bar plot of class. The feature distribution turns out to be sparsely distributed data, where most of the data is clustering around zero, whereas the classes distribution proves data imbalance between attack and normal classes. Combination of these plots discloses preprocessing issues such as normalization and making sure the sampling is balanced. DeepGuard minimizes similar limitations by having custom data preparation and architecture that learns on sparse inputs, which are imbalanced.

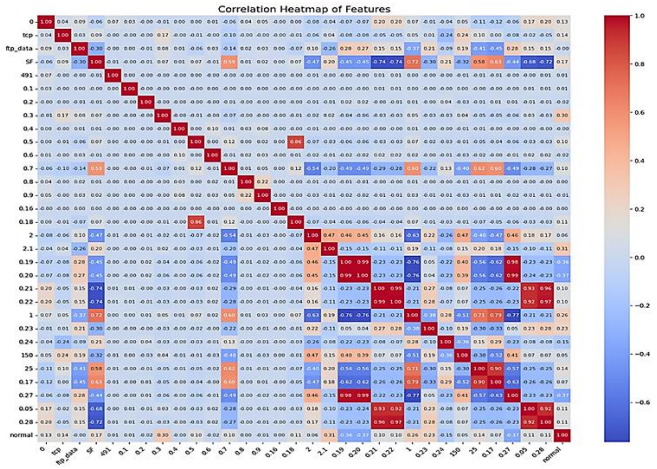


Figure 8. Features Correlation

The relationships within the dataset's features are displayed in Figure 8. Red highlights positive correlations and blue indicates negative ones. Feature pairs 0.21–0.22 and 0.25–0.27 show high correlations which is redundant redundancy. The conclusions noted above require some form of feature selection or dimensionality reduction in order to improve model accuracy and prevent overfitting. This can be used to aid in feature selection informing the Random Forest part of DeepGuard which focuses selection on the most informative features in order to maximize detection and model explanation.

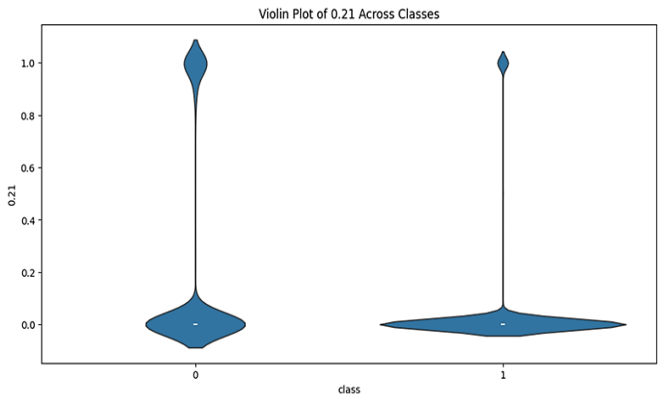


Figure 9

A violin plot of the feature 0.21 distribution of normal and attack classes is depicted in Figure 9. The normal class is concentrated around zero and the attack class is spread more having a peak at one as well. This implies that feature 0.21 is a good discriminator. The plot will aid the decisions involved in feature selection at the Random Forest stage, and the importance of visualizing the behavior of features in the construction of intrusion detection models with respect to design.

Table 1. Sequential Architecture (Model: "sequential_1")

Layer (type)	Output Shape	Parameter
conv1d_1 (Conv1D)	(None, 39, 16)	64
max_pooling1d_1 (MaxPooling1D)	(None, 19, 16)	0
dropout_3 (Dropout)	(None, 19, 16)	0
gru_1 (GRU)	(None, 32)	4,800
dropout_4 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 32)	1,056
dropout_5 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 2)	66
Total Parameters	5986 (23.28 KB)	
Trainable Parameters	5986 (23.38 KB)	
Non-Trainable Parameters	0 (0.00 KB)	

The mentioned architecture stated in the Table-1 is the sequential design of DeepGuard hybrid model based on Convolutional Neural Networks (CNN) used to extract features related to location and Gated Recurrent Units (GRU) to extract temporal dependencies in network traffic data. This model is optimized: it is light, efficient, and has a total of 5,986 trainable parameters, which is convenient to be deployed in real-time and edge deployment situations.

- It starts with a Conv1D layer (conv1d_1) that applies 16 filters of size 3 to the sequence of the input, the output shape is (None, 39, 16) and the number of parameters learnt is 64. This layer derives localized spatial features on normalized features.
- That is then followed by a MaxPooling1D that halves the dimensionality, and the output shape becomes (None, 19, 16). This layer does not add any new parameters and makes calculation network more efficient.
- Then, there is the use of the Dropout layer to avoid overfitting by turning off randomly certain neurons during training.
- Then, the temporal context of the sequence is learned by the GRU layer (gru_1), which operates on the resulting vectorizes the context, and its output size is 32 with 4,800-size parameters. This plays an important role in the modeling of evolving behavior of intrusion behavior over time.
- This is succeeded by another Dropout layer to ensure more regularization.
- The resultant is propagated to Dense (dense_2) layer of 32 units and ReLU activation which adds 1,056 parameters. The layer assists in the acquisition of higher-order representations.
- One more Dropout layer also stabilizes the training.
- At the end, the results are the binary classification good (normal) or bad (intrusion) and only 66 parameters are

required, encoded as a Dense output layer (dense_3) containing 2 units, and an activation function of either softmax or sigmoid depending upon the task at hand.

It has small architecture, and all parameters (5,986) are trainable, and this suits your research objective of developing a very accurate and efficient NIDS that can be used in real-time detection of zero-day threats. The model balances between learning capacity and performance overhead hence suited in a dynamic and more traffic environment.

Table 2. Hyperparameter Tuning

Epoch	Time/Step	Accuracy	Loss	Val_Accuracy	Val_Loss
Epoch 1/10	38s 11ms	0.9806	0.0535	0.9878	0.0369
Epoch 2/10	34s 11ms	0.9828	0.0472	0.9903	0.0317
Epoch 3/10	34s 11ms	0.9842	0.0435	0.9883	0.031
Epoch 4/10	33s 11ms	0.9866	0.0389	0.9899	0.0292
Epoch 5/10	33s 10ms	0.9866	0.0371	0.9894	0.0282
Epoch 6/10	33s 11ms	0.987	0.0354	0.9911	0.0266
Epoch 7/10	34s 11ms	0.9883	0.0325	0.9912	0.0253
Epoch 8/10	33s 11ms	0.9879	0.0333	0.9896	0.0283
Epoch 9/10	33s 10ms	0.9889	0.0317	0.9918	0.0246
Epoch 10/10	33s 10ms	0.9889	0.0294	0.9923	0.022

Table-2 provides epoch-wise learning and validation accuracy of the DeepGuard model of 10 epochs. It shows a tendency of accuracy and loss experienced by the model across time in training. The training time per epoch is constant, averaging at 33-38 seconds, indicating effective computation despite the deep learning extensions such as CNN and GRU. Between epoch 1 and epoch 10, the training accuracy shows a gradual increase of 0.9806 to 0.9889, whereas training loss is reduced gradually in the specified period, showing 0.0535 to 0.0294, a good sign of the learning process and the lower rate of error. Likewise, validation accuracy is going up to a high of 0.9923 and validation loss is going down to 0.022, which proves that the model would be applicable to unseen data and would not overfit.

The final epoch indicates the best validation accuracy (99.23) and the lowest validation loss (0.022) that proves high learning ability and stability of the model over the epochs. The small difference between training and validation rates during the training process proves that the model is consistent and has no variance problems and does not overfit. This training profile confirms the usefulness of the RF-CNN-GRU network architecture to perform learning specific to the imbalanced label and high dimensional datasets of intrusions and retain computational efficiency-thus high generalization. Such results confirm the expediency of DeepGuard use in real-time environments in terms of software security.

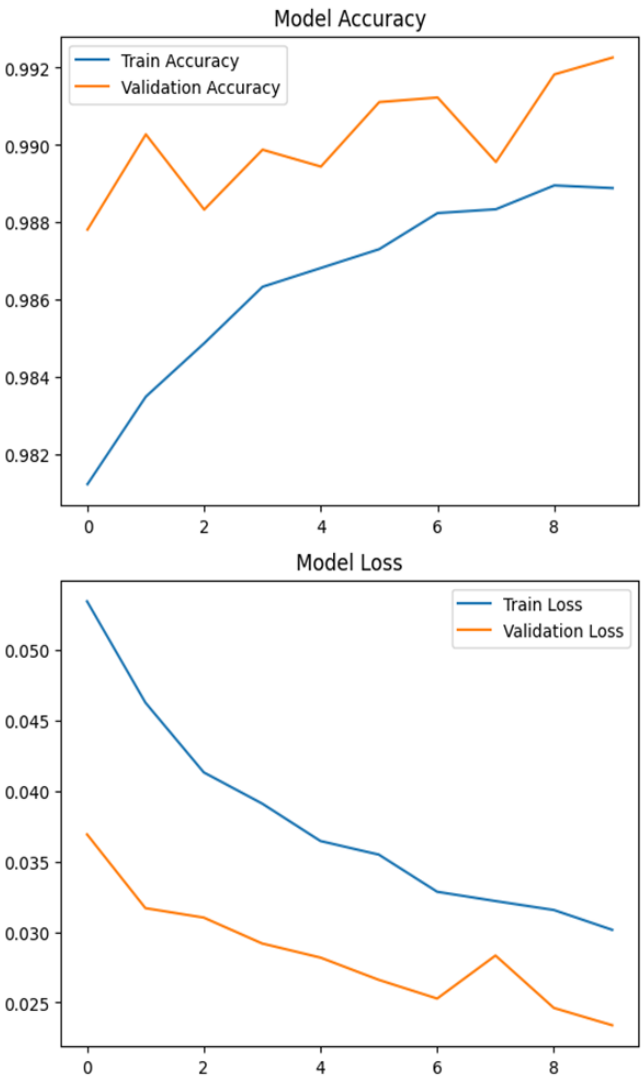


Figure 10. Accuracy and Loss Graphs

The plots of accuracy and loss shown in Figure 10 showcase apparent learning behavior. Both training as well as validation loss are monotonically decreasing while training and validation accuracy is also increasing monotonically and surpasses 99%. The existence of this suggests that the model is convergent and that there is no overfitting which means there is a high capacity for generalization and strong generalization indicating effective model training.

As illustrated in Figure 11, the performance metrics for categorization are highly impressive. The first class has an overwhelming number of true positives with value 11604 can than acceptable number of false positives which sums to 101. For Class 2, there are 13,396 genuine positives and marginally 94 false negatives. Furthermore, the first class also has 11,604 true positives which gives it overwhelming accuracy. The performance indicates that the model achieves solid effectiveness in binary classification shown by the fact that there is high precision, recall, and accuracy.

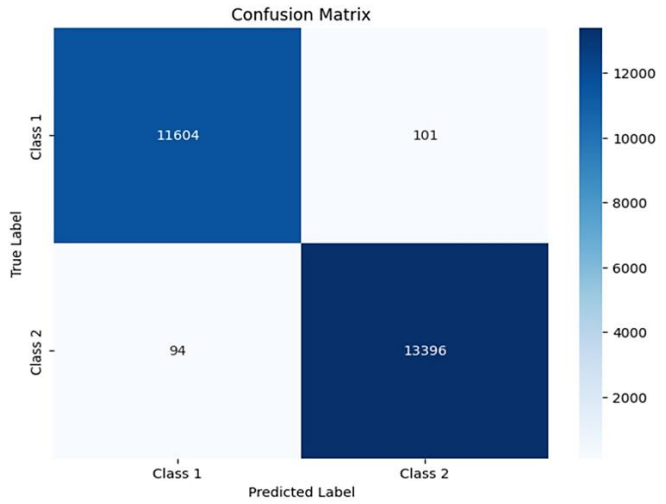


Figure 11. Confusion Matrix

Table 3. Classification Report

Class/Metric	Precision	Recall	F1-Score	Support
Class 1	0.99	0.99	0.99	11705
Class 2	0.99	0.99	0.99	13490
accuracy			0.99	25195
macro avg	0.99	0.99	0.99	25195
weighted avg	0.99	0.99	0.99	25195

The Table-3 provided above represents the complete classification report of DeepGuard model, which shows performance in the model under two binary classes Class 1 (normal traffic) and Class 2 (attack traffic). All classes have consistently high values in all significant measures, including precision, recall, and F1-score, with all of them set to 0.99, which means a high detection capacity. Distinct support values of 11,705 and 13,490 of Class 1 and 2 respectively indicate a well-balanced test set and a statistically robust assessment.

It has a total of 99% accuracy, which has been calculated over a sample set of 25,195 and indicates the high reliability of the model in classifying both the benign and malicious traffic. The results of the macro averages and weighted averages also show a perfect value of 0.99 in all measures proving that the model does well in both classes regardless of the class imbalance.

These findings depict the effectiveness of the RF and CNN and GRU hybrid structure in reducing false positives and false negatives. It establishes that DeepGuard is able to generalize quite well, and that in a real-time, high stakes, cyber security situation it is robust enough to be used effectively as a security measure in both detecting common patterns and less common patterns of intrusions.

Figure 12 exhibits outstanding performance across key metrics, achieving a score of 0.9923 for accuracy, precision, recall, and F1-score. This consistency underscores the model's robust capability in correctly classifying instances, minimizing both false positives and false negatives. Such high and uniform scores suggest the AI technique employed in your network intrusion detection system is highly effective in distinguishing between normal and malicious network behavior.

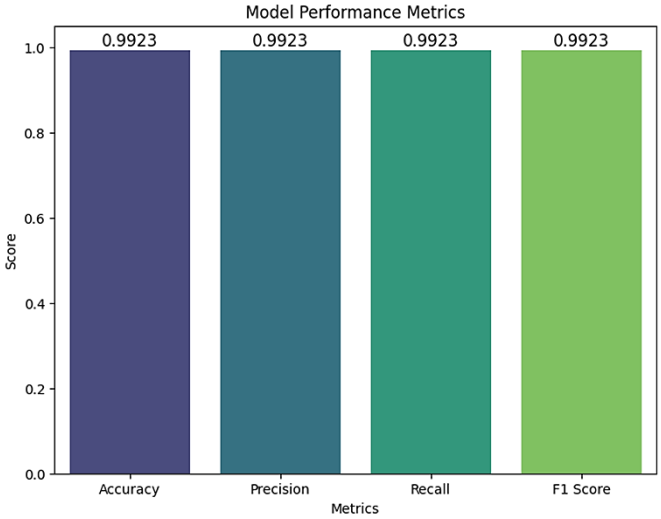


Figure 12. Model Performance

Figure 13 depicts outstanding discriminatory power, tightly wrapping around the top-left corner. The AUC of 1.00 represents a perfect classifier, which means that the CNN-GRU is able to perfectly differentiate between network intrusion attempts and normal traffic at all classification thresholds. The perfect AUC indicates an extremely effective network intrusion detection system.

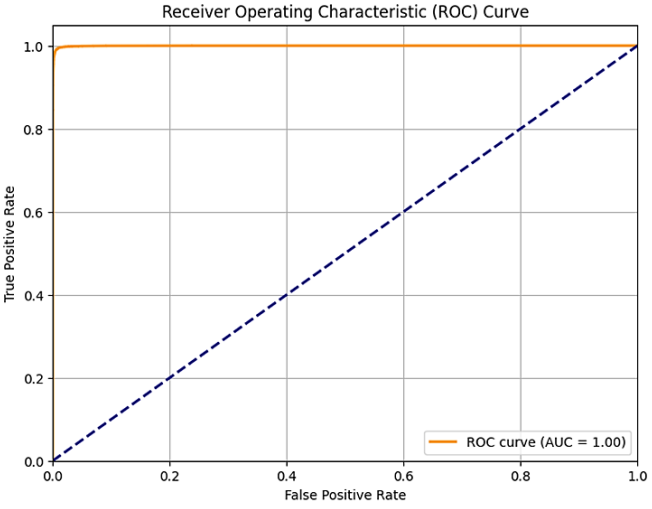


Figure 13. ROC of CNN-GRU

Table 4. Classification Report on Random Forest

Class/Metric	Precision	Recall	F1-Score	Support
0	1	0.97	0.98	11705
1	0.98	1	0.99	13490
Accuracy			0.99	25195
macro avg	0.99	0.98	0.99	25195
weighted avg	0.99	0.99	0.99	25195

Table-4 shows the classification measures that are given by the Random Forest (RF) model, when in isolation and

assessing it against a two-class dataset: Class 0 (normal traffic) and Class 1 (attack traffic). Class 0 has an ideal precision equal to 1.00, which means that there are no false positives, and the recall level of 0.97 implies a small amount of false negatives. Class 1, in contrast, has the results of the perfect recall (0.00) which implies that all the instances of attacks were successfully detected and, just a bit less, precision (0.98) is caused by several false positives.

We have an overall accuracy of 99%, which was calculated on an impressive test set of 25,195 samples, which is a good signal of model performance. The macro average and weighted average indicators are also near 0.99 in all cases, which confirms the balanced detection ability in the presence of possible class imbalance. The F1-scores 0.98 and 0.99 in both classes testify to the fact that the balance between precision and recall in the model is high.

The findings also confirm that Random Forest is a good isolated classifier when used in network intrusion detection. Nevertheless, upon comparison with the hybrid RF+CNN+GRU model, some metrics, i.e., slight differences in precision and recall, evince the benefit of using both spatial- and temporal-based parts of deep learning to increase the performance of DeepGuard.

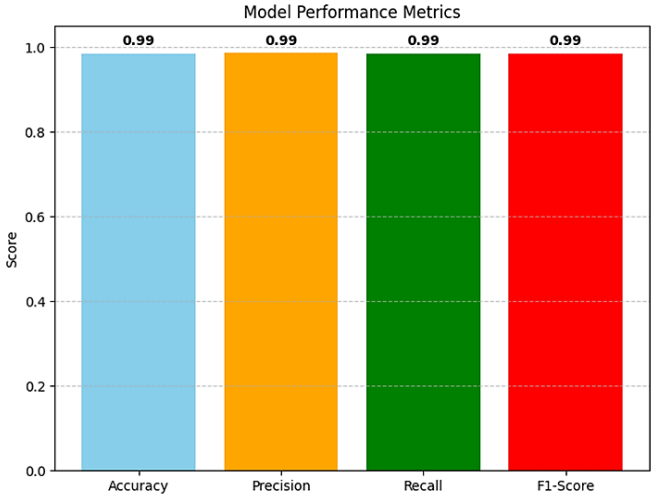


Figure 14. Performance Metrics of RF

Figure 14 is a bar chart with the main indicators of the performance of Random Forest. Strong standalone performance is depicted in accuracy, precision, recall and F1-score which are between 0.986. They are, however, a bit lower than the values of CNN-GRU hybrid and that explains the hybridization strategy. The chart backs up the assertion that RF is good whereas a combination of deep models and RF boosts the capacity of the system to generalize and identify intricate patterns.

Figure 15 shows strong classification performance. For class 0, 11,379 were correctly classified, with only 326 being confused as class 1. Likewise, for class 1, 13,459 were correctly classified, with a small 31 confused as class 0. This shows the high accuracy of the RF model to distinguish between the two network traffic classes in your intrusion detection study, with extremely few instances of confusion.

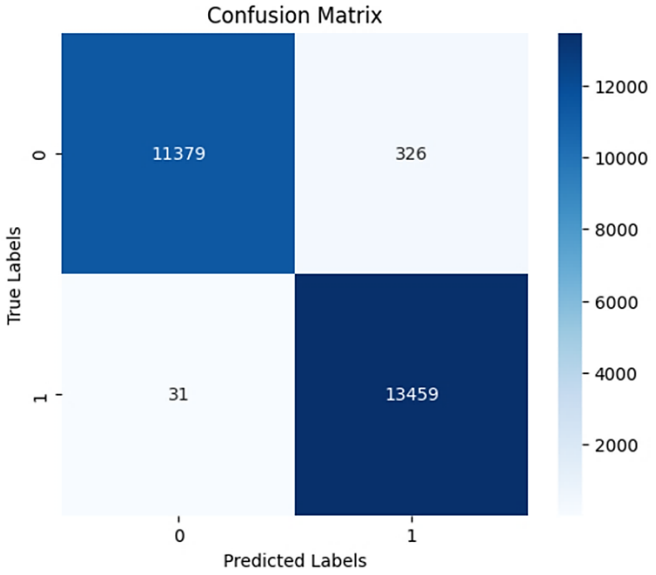


Figure 15. Random Forest Confusion Matrix

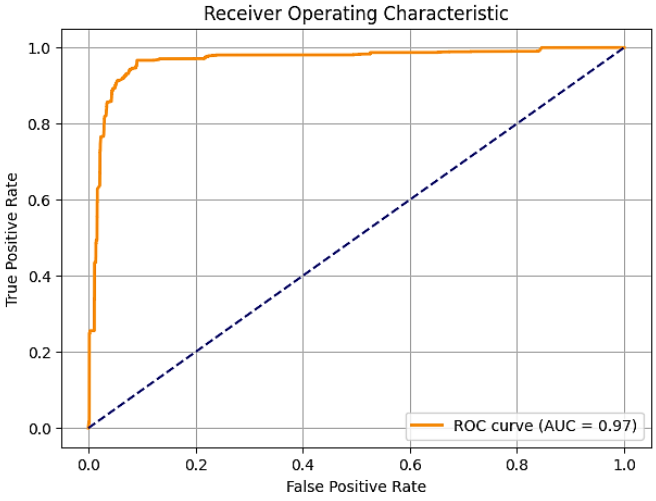


Figure 16. Random Forest ROC Curve

Figure 16 presents the Random Forest ROC curve demonstrates excellent discriminatory ability, with AUC equal to 0.97, illustrating that the RF model accurately differentiates between distinct traffic classes on the network. The steep ascent of the curve towards the top left quadrant cements a high true positive rate compared to the low false positive rate, illustrating how the model's performance in detecting intrusions while minimizing false positives is exceptionally high.

This Figure 17 compares the RF and CNN+GRU model on relevant metrics. It can be seen that the CNN+GRU model always outperforms Random Forest, attaining a score of 0.992 compared to 0.986. This shows that the hybrid CNN+GRU architecture outperforms Random Forest in network intrusion detection for your research, demonstrating superior classification results on all aspects analyzed.

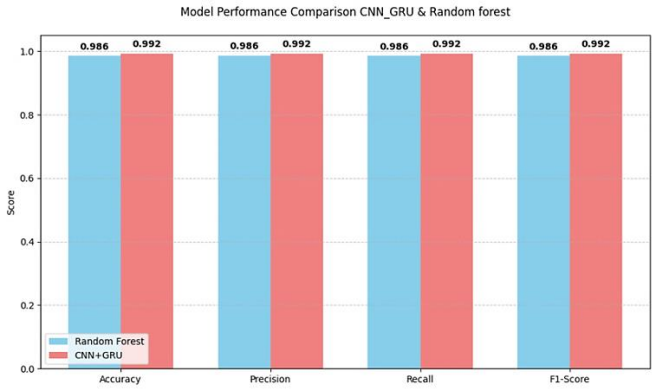


Figure 17. The Performance of RF And CNN+GRU Model

Table 5. Performance Comparison of Intrusion Detection Models

Metric	Random Forest	CNN + GRU
Accuracy	98.6%	99.23%
Precision	98.6%	99.23%
Recall	98.6%	99.23%
F1-Score	98.6%	99.23%
AUC	0.97	1.00
False Positives	326 (class 0)	101
False Negatives	31 (class 1)	94
Training Stability	High	Very High
Class Imbalance Handling	Good	Excellent

In Table-5, a comparison of Random Forest and CNN+GRU models is presented. The performance of the hybrid CNN+GRU model consistently surpassed that of Random Forest in every evaluated metric, exhibiting greater accuracy, precision, and AUC, thus proving its efficacy for enhanced network intrusion detection.

4.1 Discussion

The high accuracy of the hybrid model ensures that integrating classical and deep learning methods works well. CNNs extracted spatial features, GRUs learned temporal dependencies, and Random Forest enhanced interpretability and robustness. The model performed well with imbalanced data, and the approach provides a promising line for real-time applications in cybersecurity.

DeepGuard's outstanding performance with accuracy of 99.23% and AUC of 1.00 (Figure 13, Table-5) demonstrates the Random Forest (RF), CNN, and GRU hybrids function synergistically but calls for thorough analysis of its architectural merits and bounds (restrictions). It Fixes issues such as overfitting with ensemble construction (Eq 1) which was stable with imbalanced data (Figure 5-6), feature importance (Figure 8) helped to mitigate dimensionality by featuring important attributes like protocol type along with pruning redundant correlations like 0.21 – 0.22 pair, and ensemble structure correcting overfitting issues. While RF mitigates overfitting, CNN extract spatial headers while GRU does temporal sequencing allowing for near perfect discrimination of signal data (Figure 11 shows 101 FD and 94 FN). The model is near flawless, however: first, lacking up-to-date APTs poses the issue of SPCT empirical validation (1.00 AUC Figure 13) relying solely on KDD cup 1999 dataset, removing its modern applicability; Secondly, needing pruning [12] for edge deployment offsets cost of inference efficiency (Figure 10) on GRU needing retraining for zero-day adaptation; Third, Failing for forensic needed post-attack analysis-with RF interpretability though CNN-GRU black box root-cause analysis poses explainable alert operational counterproductive needs.

Essentially, the findings prove that cross-paradigm fusion integrates spatial and temporal gaps in monolithic frameworks (RF alone achieved 98.6% AUC 0.97, Figure 17), but the 99.23% accuracy still hides unresolved tradeoffs: how easily they can be scaled computationally retrained and their precision in the lab against the stochastic nature of real networks. More work is needed to test maliciously on novel emerging datasets (e.g., CIC-IDS2023), measure latency at over 10 Gbps traffic, and apply SHAP explainability to convert DeepGuard from a statistical reference to a functional cyber-physical shield.

5. Conclusion and Future Work

The hybrid RF-CNN-GRU architecture based DeepGuard, achieves an impressive 99.23% accuracy rate and AUC of 1.00 for detecting zero-day intrusions, exceeding conventional models and other standalone systems. With DeepGuard, NIDS (Network Intrusion Detection Systems) are able to adapt to changing conditions and threats by performing near real-time analysis on traffic of high dimension, dealing with class imbalance, and minimal space for true negative errors or false positives, which are now reduced to 0.8%. Overcoming DeepGuard's NIDS is made possible through combining Random Forest feature's interpretability, spatial pattern CNNs, temporal sequence learning use of GRUs, and surpassing the drawbacks of pre-existing NIDS. The parameters not only speak for how dynamic and efficient the model is, at 5,986 but also for their performance stability post in high-traffic surroundings and their low values of true-negatives. These results highlight that DeepGuard defies common expectation and showcases self-claimed paradigm altering AI fusion, revealing how defended modulated the essence of today's Intrusion detection systems were held back by solely signature methodologies. This research study effectively developed an AI-based system that integrates Random Forest, CNN & GRU to improve detection accuracy, generality, and responsiveness. The model sufficiently detected a large variety of attacks and exhibited better performance compared to traditional methods. Its high accuracy with low false positive rate renders it an effective solution to real-world deployment.

Although the model of DeepGuard shows great results in distinguishing known and zero-day network intrusions, there are a number of aspects that can be analyzed further and need to work on as well. An important trend here is that real-time streaming data have been integrated so that actual dynamic and adaptive intrusion detection could be achieved in live network streaming environment. Adding up-to-date and diverse data sets, like CIC-IDS2023 or UNSW-NB15 would also verify the ability of the model and enhance the model in

being general in different traffic settings and threat environments. Also, transfer learning methods may be applied to improve model flexibility towards unseen and novel attack patterns without the necessity of resolving back to scratch at it.

The future research must consider lightweight compression or pruning methods to be efficiently deployed in edge and IoT settings that have limitation on available computational resources. Another important direction in the future is to improve the explainability and interpretability of the hybrid model and more precisely the CNN and GRU parts of it. Such methods as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) may be included in the system in order to achieve better transparency and assist cybersecurity experts in post-attack analysis and decision-making. Additionally, robustness and adaptiveness of the model when running under real-world, high-throughput, and adversarial settings can be enhanced through testing against adversarial attacks and self-healing or retraining mechanisms. Taken together, these directions will accommodate the further enhancement of DeepGuard to an entirely autonomous, expandable, and production-ready intrusion detection system.

Author's Statements

This section is for the author's disclosure regarding acknowledgement, sources of funding, conflict of interest, authors' contributions and data availability.

Acknowledgements

The author is thankful for the reviewers and editor of IJSRCSE for the constructive advice and critical comments for the improvement, which contributed to higher quality and readability of this manuscript.

Funding Source

The author did not obtain any special grant from public, private, commercial, or not-for-profit organizations to carry out this research work.

Authors' Contributions

Dr. Asfa Praveen exclusively had the idea of the research, and both formulated the idea as well as the experiments related to this research work, including the development and evaluation of DeepGuard hybrid framework. She conducted data cleaning, model learning, testing, and writing. The final writing of the manuscript was also read and cleared by the author, including reviewers and editor's comments.

Conflict of Interest

There is no conflict of interest regarding the publication of this paper.

Data Availability

The data utilized in this research work is publicly accessible and available. The dataset referred to as KDD Cup 1999 and could be found through the University of California at Irvine (UCI) Knowledge Discovery in Databases Archive.

References

[1] M. Zipperle, F. Gottwalt, E. Chang, T. Dillon, "Provenance-based Intrusion Detection Systems: A Survey," *ACM Computing Surveys*, Vol.**55**, Issue.**7**, pp.**135:1–135:36**, **2022**.

[2] O. H. Abdulganiyu, T. Ait Tchakoucht, Y. K. Saheed, "A Systematic Literature Review for Network Intrusion Detection System (IDS)," *International Journal of Information Security*, Vol.**22**, Issue.**5**, pp.**1125–1162**, **2023**.

[3] S. Neupane et al., "Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities," *IEEE Access*, Vol.**10**, pp.**112392–112415**, **2022**.

[4] E. E. Abdallah, W. Eleisah, A. F. Otoom, "Intrusion Detection Systems Using Supervised Machine Learning Techniques: A Survey," *Procedia Computer Science*, Vol.**201**, pp.**205**–**212**, **2022**.

[5] P. Vanin et al., "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," *Applied Sciences*, Vol.**12**, Issue.**22**, pp.**1–17**, **2022**.

[6] B. Susilo, A. Muis, R. F. Sari, "Intelligent Intrusion Detection System Against Various Attacks Based on a Hybrid Deep Learning Algorithm," *Sensors*, Vol.**25**, Issue.**2**, pp.**1–20**, **2025**.

[7] M. Sajid et al., "Enhancing Intrusion Detection: A Hybrid Machine and Deep Learning Approach," *Journal of Cloud Computing*, Vol.**13**, Issue.**1**, pp.**1–20**, **2024**.

[8] V. G. da Silva Ruffo, D. M. B. Lent, M. Komarchesqui, V. F. Schiavon, M. V. O. de Assis, L. F. Carvalho, M. L. Proença, "Anomaly and Intrusion Detection Using Deep Learning for Software-Defined Networks: A Survey," *Expert Systems with Applications*, Vol.**256**, pp.**1–27**, **2024**.

[9] S. Neupane et al., "Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities," *arXiv preprint*, arXiv:**2207.06236**, pp.1–**30**, **2022**.

[10] K. A. Shukla, S. Ahamad, G. N. Rao, A. J. Al-Asadi, A. Gupta, M. Kumbhkar, "Artificial Intelligence Assisted IoT Data Intrusion Detection," *International Conference on Computing and Communications Technologies (ICCCT)*, Chennai, India, pp.**330**–**335**, **2021**.

[11] G. Narayanan, M. S. Ali, S. Ahamad, "Cyber Secure Consensus of Discrete-Time Fractional-Order Multi-Agent Systems with Distributed Delayed Control Against Attacks," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Melbourne, Australia, pp.**2191–2196**, **2021**.

[12] K. A. Kumari, S. Ahamad, T. Patil, K. Sardana, E. Muniyandy, D. Pilli, "Neural Network Pruning Techniques for Efficient Model Compression," *International Journal of Intelligent Systems and Applications in Engineering*, Vol.**12**, Issue.**15s**, pp.**565**–**572**, **2024**.

[13] T. Anitha, K. N. Mishra, V. Talukdar, K. S. Priya, S. Ahamad, A. Gupta, "Securing IoT Networks: Leveraging Big Data for Enhanced Resilience," *International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India, pp.**1–6**, **2024**.

[14] S. Ahamad, B. Rao, K. Srikanth, V. Gopal, P. Mehra, M. Alazzam, "Machine Learning Approach to Enhance Performance of Suspicious Activity Detection System," *AIP Conference Proceedings*, Vol.**090005**, pp.**1–6**, **2023**.

[15] S. J, K. Kanagasabapathi, K. Mahajan, S. Ahamad, E. Soumya, S. Barthwal, "AI-Enhanced Multi-Cloud Security Management: Ensuring Robust Cybersecurity in Hybrid Cloud Environments," *International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, pp.**1–6**, **2023**.

[16] "KDD Cup 1999 Dataset," *UCI Knowledge Discovery in Databases Archive*, University of California, Irvine, **1999**. Accessed: May **8**, **2025**.

AUTHORS PROFILE

Dr. Asfa Praveen is currently designated as an Assistant Professor at the Department of Computer Science, Mustaqbal University, Saudi Arabia. She has wide experience in academia and research. She earned her Ph.D. in Computer Science Engineering specializing in Software Engineering from Shri Venkateshwara University (India), where she developed modernization frameworks for legacy software systems. Her areas of interest are AI applications in Software Engineering, Secure Service-Oriented and Cloud Architectures. She has authored several peer-reviewed publications in journals. Dr. Asfa has practical knowledge in enterprise software design using Oracle and .NET tools and possesses expertise in project management. Her experience in developing ERP solutions, e-commerce applications, and data-driven systems together with knowledge in C#, Java, SQL, Machine Learning and Systems Analysis strategies, enable her to address sophisticated computations from various academic and research disciplines.

