Multi-Phase Intrusion Detection System using Machine Learning

Mamidi Khushika Reddy[1]
[1]Vellore Institute of Technology, Chennai, mamidikhushikareddy@gmail.com

Abstract - **In the present digital world, the increase of cyber-attacks is posing a serious threat to computer systems all around the world. Having a strong Intrusion Detection Systems (IDS) is highly required and crucial. This project introduces a new way of detecting intrusions by using a mix of different types of neural networks. These include Convolutional Neural Networks (CNN), Shallow Neural Networks (SNN), Deep Neural Networks (DNN), Feedforward Neural Networks (FNN), and Long Short-Term Memory (LSTM) models, all working together in a multi-step process i.e. they are seamlessly integrated together. Thereby working as a single Multiphase Intrusion Detection System. We worked on the publicly available standard dataset and training this data in our models by bringing together a variety of neural network types and making the most of LSTM models' unique abilities, our system significantly improves intrusion detection accuracy. This comprehensive approach not only helps us spot intrusions more effectively but also contributes to the ongoing fight against cyber threats in today's increasingly complex digital world.**

Keywords-component: Convolutional Neural Networks (CNN); Deep Neural Network (DNN); Feedforward Neural Networks (FNN); Long Short-Term Memory (LSTM); multi-phase network intrusion detection; neural network; Shallow Neural Network (SNN).

I. INTRODUCTION

In the ever-evolving cyber world, where everything from personal communications to critical infrastructures rely on computer systems, the threat of cyber-attacks is increasing over time. These attacks not only make our digital assets vulnerable but also pose risks to our data, privacy, finances, and even safety. Security of network system is becoming very crucial as more sensitive information is being exposed, stored extensively and frequently manipulated online. It is tough to prevent attacks only by passive security policies, firewall, or other low performing mechanisms. Intrusion Detection Systems (IDS) have thus become a very critical technology to help safeguard these systems in an active way. An IDS collects the system and network activity data, and analyzes the available information to determine if there was an attack.

This is project is an innovative approach for intrusion detection that integrates the powerful features of neural networks. By incorporating different types of neural networks, like CNNs, SNNs, DNNs, FNNs, and LSTM models, a dynamic defense system is created which is abled enough of effectively identifying intrusions in computer networks. The main objective of this project is to design and develop an intrusion detection for computer networks. This project is built to improve the detection rate for all the known and unknown attacks. First, we train and test all our models individually and then we incorporate the learned models into our hybrid model to make it work on the normal and the unknown intrusion data. The main objective of this project is to design and develop an intrusion detection for computer networks. This project is built to improve the detection rate for all the known and unknown attacks. First, we train and test all our models individually and then we incorporate the learned models into our hybrid model to make it work on the normal and the unknown intrusion data.

II. PREVIOUS WORK

[1] With the growing usage of the Internet with its vulnerabilities, there has been a call for implementing Intrusion Detection Systems (IDSs) to ensure security. IDSs are defensive systems that detect outsider infiltrations, unauthorized accesses, and malfunctions occurring in computer networks. Intrusions can be detected and reported to the network administrator by IDSs using various pieces of information such as port scanning and irregular traffic detection. Intrusion detection is a classification problem, and discovering effective features is an important aspect of classification methods. Standard methods used for classification are neural networks, fuzzy logic, data mining techniques, and metaheuristics. One of the novel metaheuristic algorithms introduced to address optimization problems is the Horse herd Optimization Algorithm (HOA). This paper introduces a new approach based on HOA for network intrusion detection. **[2]** The paper presents a network intrusion detection model using a convolutional neural network and a gated recurrent unit to solve problems of low accuracy in existing intrusion detection models for multi-classifying intrusions and low accuracy in detecting class imbalance data. Hybrid sampling ADASYN and RENN is used as a sample processing algorithm to solve problems with positive and negative sample imbalance in the original dataset

[3] Attacks against Industrial Internet have surged radically in recent years. The traditional Internet is more complicated in network structure than the industrial Internet, and the traditional graph neural network attack behavior detection model cannot be well adapted to the complex network environment. In order to extend the ability of the model to adapt to the complex network environment, we propose the E-minibatch Graph SAG model. Source port and source IP address at the application layer will be used as source nodes, target port and target IP address at the application layer will be used as target nodes, and the rest of the traffic information will be used to establish the graph structure data by the rest of the remaining traffic information. **[4]** This paper introduces a new Network Intrusion Detection System based on Graph Neural Networks. GNNs are a relatively new sub-field of deep neural networks, leveraging the inherent structure of graph-based data. Training and evaluation data for NIDSs are typically represented as flow records, which can naturally be represented in a graph format. In this paper, we propose EGraphSAGE, a GNN approach that allows capturing both the edge features of a graph as well as the topological information for network intrusion detection in IoT networks. To the best of our knowledge, our proposal is the first successful, practical, and extensively evaluated approach of applying GNNs on the problem of network intrusion detection for IoT using flow-based data. **[5]** The IoT ecosystem has sprawled due to the employment of the internet and cloud-based technologies in the industrial area. The IoT technology utilized in the industry has grown to be a large network based on the increasing amount of data and number of devices. IIoT networks are inherently insecure against cyber threats and intrusions. Thus, it is very important to propose Intrusion Detection Systems (IDS) for the security of the IIoT networks. Three different models were proposed to detect intrusions in the IIoT network by employing deep learning architectures of Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and 14 CNN + LSTM generated from a hybrid combination of these. In the study carried out by employing the UNSW-NB15 and X-IIoTID datasets, normal and abnormal data were determined and compared with other studies in the literature following a binary and multi-class classification. **[6]** This paper deals with the complex issue of detecting and preventing malicious network behavior, which is indispensable for constructing secure networks for communication. This is achieved by proposing a model, CNN-GRU-FF, which has a double-layer feature extraction and feature fusion technique with a modified focal loss function to fix class imbalance in intrusion detection datasets. The method shows excellent results, in which it achieved detection rates of 99.68% and 98.22% respectively with low false alarm rates by using the NSL-KDD and UNSW-NB15 datasets. However, the study had a limitation; because it was evaluated on the datasets, the generalizability to networks of other kinds could be affected. **[7]** This present research project basically faces challenges towards security in the IoT landscape of over 75 billion connected IoT devices in 2025. In regards to IDS, it more than once analyzes over 60 articles with the effort to unravel faultiness of current solutions. Novel contributions of this research include developing a taxonomy of IoT IDS, developing an architectural study of IoT networks, analyzing the attacks in IoT, and doing a comprehensive review of current limitations in IDS. All of this will lay a sound base for effective security measures in IoT. **[8]** This study focuses on the need of better intrusion detection systems in the distributed computing environment, considering the significance of security with increased interruptions. The existing IDS systems often face feature selection and classification due to high dimensionality and irrelevant features. The study thus adds a new approach to the ensemble machine learning approach of Whale Optimization Algorithm with Genetic Algorithm and Random Forest Integration. The preprocessing detail shows feature scaling and margins as determined by the analysis of User Behavior and Frequency of Protocols. WOA and genetic algorithms are applied to refine non-relevant features while Random Forest Integration is applied for identification of intrusion in the IDS. This framework shows high effectiveness in intrusion detection and outperforms other systems in terms of accuracy. The limitation is that the system depends on specific algorithms and datasets, which may restrict the generalization of the network to any environment. **[9]** It is a new framework for network behavior classification, which is very important in the field of cybersecurity. It combines ensemble learning techniques with the generative adversarial networks method. In the process, the work tackles the challenge of the imbalance issue in the UNSW-NB15 dataset, which is used for this purpose. By introducing a conditional tabular generative adversarial network, the paper addresses the challenge of the dataset's imbalance. In such a way, it aims to overcome the imbalance. That study achieves remarkable improvements in the detection of intrusions by training shallow binary classification algorithms on both balanced and imbalanced datasets. Further enhancement comes through a two-stage label-wise ensembling process, which ends with an XGBoost meta-classifier. The framework achieves 98% accuracy in the binary classification and 95% in the multi-class classification, outperforming any other state-of-the-art model. However, it is still limited because the work only evaluated it on a specific dataset, which makes it hard for generalization in other network environments.

III. TRADITIONAL SYSTEMS

The Signature-based intrusion detection is a conventional method for detecting network intrusions, plays an important role in pre-processing phase of a multi-phase network intrusion detection system that uses graph neural networks. During this phase, data is gathered and cleansed to eliminate irrelevant information and noise, making it compatible with machine learning algorithms. The pre-processing includes a signature-based method uses cross-referencing the cleaned data with a repository of preexisting defined signatures or

patterns linked to known attacks. Upon detecting a match, the system alerts or helps have minimal response time for implementing the countermeasures. On the other hand, behavior-based approaches, detect the anomalies in network behavior without depending on the predefined signatures, can be included into the system's modeling phase. This project involves training neural network models such as CNN, SNN, DNN, FNN, LSTM to detect and flag abnormality from typical network activity, potentially indicating intrusions.

IV. METHODOLOGY

This Multi-phase Intrusion Detection System's methodology includes employing of an ensemble architecture using multiple neural network models for intrusion detection within the same multi-phase framework. The dataset used in this project for the experimentation is the Kaggle's "Network Intrusion Detection" dataset which has a collection of network traffic data with labeled instances of normal and anomalous behavior. There are three different phases for this approach: The pre-processing, feature extraction, and classification. In the first phase, input data is converted into a graph structure to identify complex relationships among the attributes. The second phase uses the neural network models to identify the relevant attributes from the graph constructed, utilizing the network's ability to hold on to the dependencies that are long and temporal patterns. The last phase has LSTM-based model employed to categorize inputs as either normal or anomaly, based on the extracted features. This methodology tries to enhance the intrusion detection accuracy by integrating diverse neural network architectures and integrating the unique features of LSTM models.

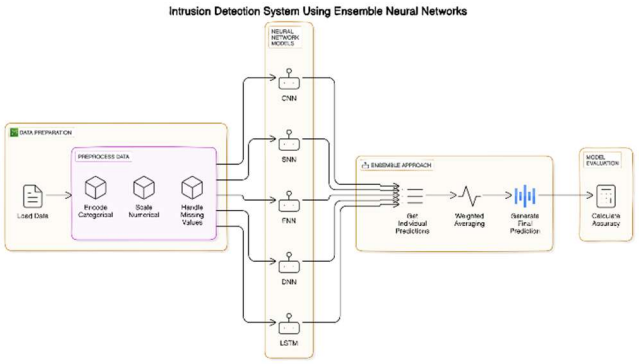
On contrary signature-based detection has some drawbacks. It is only effective in the case of known attacks, and may mismatch attacks to some other attacks that do not match with the pre-defined

signatures. Also, the signature databases must be constantly maintained and updated to keep up with new attack patterns and attacks. The alternative approach for this is the Behaviour-Based Models which also comes with some limitations, such as a higher false positive rate and the requirement for large amounts of data that has to be trained to learn the normal behaviour of the network.

TABLE I
ALGORITHMS USED

Model	No. of layers
CNN	7
SNN	3
DNN	4
FNN	4
LSTM	3

V. SYSTEM ARCHITECTURE



VI. PROPOSED SYSTEM

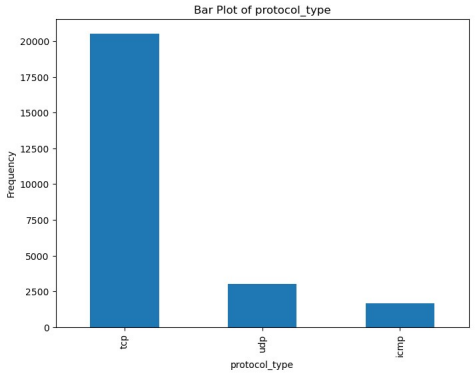
I. Introduction

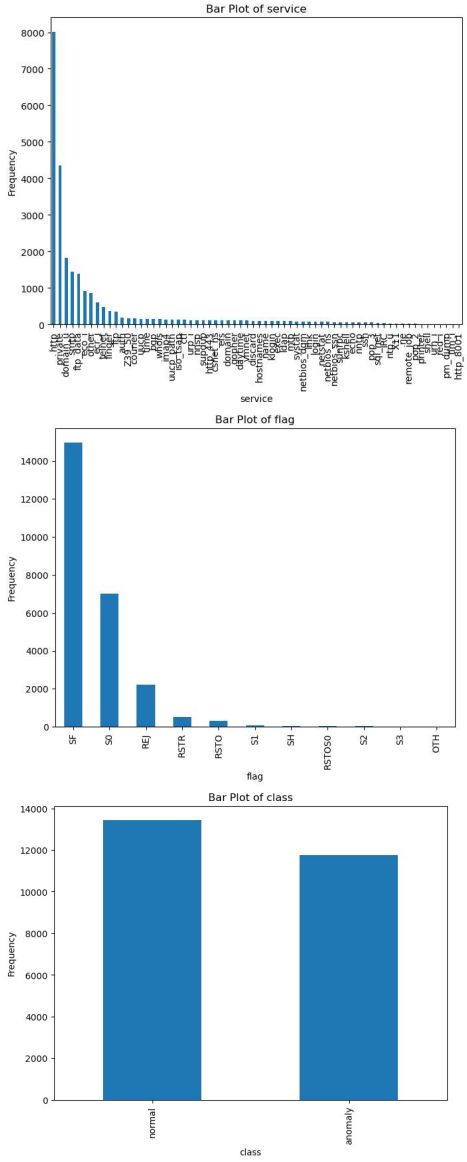
In our project Intrusion detection systems (IDS) rely heavily on ensemble models to identify harmful actions or attacks within a network. This is done by correctly labeling network traffic as one of several types of attack. Such a system can improve an organization's security stance and aide in countering any possible threats that may arise.

II. Data Preprocessing

In the preprocessing phase, unprocessed network data is loaded and processed to prepare it for modeling. This includes:

- **Handling missing values**: Any missing data points are addressed through appropriate strategies, such as imputation or removal.
- **Removing highly correlated columns**: Columns with high correlation are identified and removed to prevent multicollinearity issues.
- **Encoding categorical features:** Categorical variables like protocol type, service, and flag are encoded using label encoding to convert them into numerical format suitable for modeling.





III. Convolutional Neural Network (CNN) modelling

This very first modeling phase involves the creation and training of CNN models, which are quite efficient in extracting spatial as well as temporal patterns from sequential data. In this phase, the CNN model is built and trained to classify network traffic data into various attack categories. The CNN architecture consists of:

- **Input Layer:** This layer takes the preprocessed network traffic data, which is normally represented as a sequence of numerical features.
- **Convolutional Layers:** A layer does feature extraction by applying filters or kernels to the input data. Essentially, filters within a layer pick up some spatial patterns of the input data. For example, the filters will detect edges, textures, or shapes. In other words, the number of filters and the size of the filters are hyperparameters that determine the number of levels of complexity of the model and the size of the model's capacity for capturing different kinds of features. The most common nonlinear activation functions used after the convolutional operations are ReLU (Rectified Linear Unit), make the model representation efficient.
- **Pooling Layers:** The common pooling operations are the max pooling operations, which choose the maximum among a set of values in a window. This keeps the most important features but reduces computational costs, hence reduces the spatial dimensions of the feature maps drawn by the convolution layers.
- **Max-pooling layers:** Max pooling reduces the size of the spatial feature maps but keeps only the useful information.
- **Flatten Layer:** The layer flattens multi-dimensional feature maps from all the previously applied layers into one-dimensional vectors fed toward the following fully connected layers.
- **Fully Connected Layers:** Traditional neural network layers have fully connected layers that mean each neuron in the layer is connected to each neuron in the previous layer as well as the next layer. The fully connected layer does classification based on features learned in the convolutional layers. The activation functions such as ReLU are applied to make the model non-linear, and the final layer usually uses a sigmoid activation function for binary classification or a softmax function for multi-class classification.

CNNs implement a number of advantages in the proposed intrusion detection system project. First, due to their ability to automatically learn features, the present work does not need manual feature engineering, which enhances the ability of the model to detect diverse and complex patterns of attack. Second, as CNNs inherently detect spatial invariance, meaning they detect patterns irrespective of the location of such patterns in the input data, the precision in attack detection is increased. Third, due to parameter sharing in CNNs, the model has fewer numbers of parameters, thus increasing its ability for generalization in unknown data and reducing risk for overfitting. CNNs are also computationally efficient and can process large volumes of network traffic data in real-time or near-real-time environments. Moreover, CNNs are interpretable because they give a visualization of learned features, which helps the network administrators understand the underlying features of detected network attacks.

IV. Sequential Neural Network (SNN) modelling

Sequential Neural Network models are designed to capture complex patterns in sequential data and are most suitable in intrusion detection systems. SNNs are very important for classifying network traffic data into different attack types. The architecture of SNN comprises layers that are good at extracting temporal dependencies in data, and thus provides a strong platform for the differentiation of subtle changes in the network behaviour. In the process of training the SNN, the network learns to differentiate normal activity from anomalous activities and thus supports the IDS in quickly identifying the existence of potential security threats. SNN structure consists of the following:

- **Input Layer:** The input layer takes care of input data. It takes the shape of the input data, which in the case of this SNN model is characterized by the number of features in the dataset.
- **Flatten Layer:** This layer flattens the multi-dimensional input data into a one-dimensional array. This serves to make the data ready for further processing by the subsequent dense layers.
- **Dense Layers:** There are dense layers. Dense layers are fully connected between each layer; every neuron from the previous and next layers are connected.
  First Dense Layer (128 neurons, ReLU activation): This layer consists of 128 neurons with Rectified Linear Unit. It obtains non-linearity from ReLU so that the model can learn and capture the full picture of any pattern in the data.
  Dropout Layer (50% dropout rate): A 50% dropout rate is added in the form of a dropout layer after the first dense layer, which will help reduce overfitting during training of the model.
  Second Dense Layer (64 neurons, ReLU activation): The second dense layer includes 64 neurons with ReLU, which further learns abstract features from the data representation that is gained from the previous layers.
  Dropout Layer (50% dropout rate): A second dense layer with a 50 percent dropout rate: This is followed by a second dense layer to add more dropout regularization to the model.
  Output Layer (Number of classes, Softmax activation): The number of classes and using softmax activation. Here, it outputs the final prediction probabilities of every class within the data set. There will be a neuron for each class in the output layer.

Defined the architecture of the SNN model using proper optimizer, loss function, and evaluation metrics. Training involves the iterative update of neurons' weights that are in a goal to decrease loss function. During training on the validation set, the performance of the model is tracked in order not to overfit and generalize learned patterns to new data. The SNN model is trained for a definite number of epochs. After training, the SNN model is estimated on the validation data to evaluate achieved performance metrics like accuracy, loss, and others. One can understand from the results of evaluation how well the model has learned to classify data and generalize to new, unseen instances. This SNN model, which has been estimated, will be used to give predictions on new, unseen data, and by passing the input data through the trained model one will get the class probabilities as the output.

V. Deep Neural Network (DNN) modelling

The third modeling phase involves the implementation of Deep Neural Network models, which are good in the learnings of intricate patterns and representations of structured data. An integral part of an IDS, the DNN model is built and trained to effectively classify network traffic data into distinct attack categories. The architecture of the DNN model includes:

- **Input Layer:** The input layer receives preprocessed network traffic data, often in the form of a set of numerical features.
- **Dense Layers:** Such layers are important in the learning of high-level abstractions and apply such to make a classifier over the learned representation. Any neuron in a dense layer links with all neurons in the previous and successive layers. Use of non-linear activation functions, like ReLU, in these dense layers makes these models able to model very complex relationships within the data.
- **Dropout Layers:** These are used to prevent overfitting in the model. These layers simulate neuron drops in them during the process of training; hence the generalization performance of the model improves without memorization of the noise in the training data.
- **Output Layer:** It predicts based on the learned representations of the previous layers. Typically this will use either the sigmoid activation function for binary classification or the softmax function for multi-class classification; it will give the network probability distributions over the various categories of attack.

These DNNs have several critical advantages in respect of its implementation. First, they automatically learn relevant

features from raw network traffic data and do not require manual feature engineering, thereby reducing the time of model development, and they give higher accuracy in the detection. The DNN also captures very complex relationships between the data and in so doing is very effective in distinguishing subtle differences in different attack types. DNNs are also well equipped to handle structured data, which makes them very appropriate for large and various datasets encountered in most intrusion detection techniques. Adding in the model a dropout layer increases the model's robustness against overfitting, ensuring reliable performance on unseen data. Finally, DNNs are efficient in computational terms, which enables real-time or near real-time processing of network traffic data, which is essential in timely detection and response to security threats.

VI. Feedforward Neural Network (FNN) modelling

The fourth phase models FNN. FNN is highly efficient in learning and extracting relevant features from the data of a structured network traffic without the need for human engineering of features. Its application in the IDS project would be toward improving the models' accuracy in the detection of attack by distinguishing subtle variations among different forms of attacks with enhanced security of the networks. The model architecture includes:

- **Input Layer:** Preprocessed network traffic data are fed into the model's input layer, and these data are numerically represented features extracted from raw data.
- **Dense Layers (Fully Connected Layers):**
  First Dense Layer (128 neurons, ReLU activation): In this layer, there will be generation of high-level abstraction and complex relationship amongst input data. Here, each neuron of 128 neurons is connected with each neuron of the input layer; the activation to be used here is ReLU, which is a non-linear function.
  Dropout Layer (50% dropout rate): It consists of a dense layer followed by a dropout layer whereby 50% of the neurons are dropped randomly, to prevent overfitting and hence to increase the model's generalization ability.
  Second Dense Layer (64 neurons, ReLU activation): The second dense layer extends the learned representations to better turn those features and further fine-tunes them into the model in the pursuit of well-supported classification tasks.
  Dropout Layer (50% dropout rate): Another dense layer is followed by a dropout layer with 50% dropout rate after the second dense layer to make this model more regular and to stop the overfitting.
- **Output Layer:** The output layer of the FNN outputs samples of embeddings learned in previous layers. For the binary classification problem, it produces probability distributions among various attack categories for the binary classification problem using a sigmoid activation function.

The FNN has preprocessed network traffic data as input, usually structured as numerical features extracted from the raw dataset. The FNN produces a probability distribution across different attack categories, which allows the model to classify network traffic data into distinct attack types with associated confidence scores. The FNN implemented in the IDS project is a powerful tool for automatic learning and pattern extraction from structured network traffic data. Being a layered architecture, the FNN helps in capturing complex relationships efficiently and allows for the correct classification of network traffic into various attack categories, improving the security of the networks and increasing the capability of threat detection.

VII. Long Short-Term Memory (LSTM) modelling

Long Short-Term Memory is the major backbone of the provided code in the Intrusion Detection System. Basically, the model tries to establish the capability of the IDS in the monitoring of various categories of network traffic data. Long-range dependencies can be captured with the model; therefore, LSTM-a kind of RNN model-is rather suitable for network traffic patterns analysis over time. The architecture includes:

- **Input Layer**: The input layer should get all the preprocessed network traffic data in the form of encoded numeric features derived from raw data from the dataset.
- LSTM Layers:
- **First LSTM Layer (128 neurons):** This captures the temporal dependencies in the sequential data through its memory cell state and gating mechanisms that control information flow. Each neuron of the LSTM layer receives the input sequences and updates the state. Therefore, the model retains essential context over time.
  64 neurons LSTM Second LSTM layer: to provide fine-grained filtering of the captures from the previous layer, to fine-tune the predictions of network traffic data.
  Dropout Layer (50% dropout rate): This layer comes after the LSTM layer. Although these functional layers are good for their use, they might sometimes add an overfitting problem in the model by completely

randomizing 50% of the neurons during training, improving the ability of the model to generalize.

Second LSTM Layer (64 neurons): This LSTM Second LSTM layer refines the captured temporal dependencies from the previous layer further to generate more accurate predictions on the network traffic data.

- **Dense Layer (Output Layer):** This will be the output layer for the LSTM model, and it will make output; that is, predict the learned representations of the previous layers. This usually uses a sigmoidal activation function these days to return the probability distributions over the varying attack classes in a binary classification problem.

Idea behind the integrating LSTM layers with the IDS framework would be to capture and model effectively the temporal dynamics intrinsic in the network traffic data. When compared with conventional feedforward neural networks, LSTM models retain information along long sequences—hence, best suited for detecting faint patterns and anomalies in sequential data. The innovation is that LSTM helps mitigate the vanishing gradient problem in the standard RNN; hence, this model learns better long-term dependencies. Integrating LSTM layers helps the IDS to intensify its capability to detect complex attack patterns unfolding over time, thus enhancing network security. The LSTM model, under the IDS framework, enables the analysis and detection of complex attack patterns that might be included in network traffic data. The LSTM model, with its specialized architecture and recurrent nature, is enabled to capture temporal dependencies with effectiveness, thus enhancing capabilities for threat detection and network security.

VIII. Integrated Ensemble modelling

The integrated ensemble model implemented in the provided code combines predictions from multiple neural network architectures, including Convolutional Neural Networks (CNN), Sequential Neural Networks (SNN), Feedforward Neural Networks (FNN), Long Short-Term Memory (LSTM) networks, and Deep Neural Networks (DNN). This ensemble approach aims to leverage the diverse strengths of each individual model to enhance the overall accuracy and robustness of the intrusion detection system (IDS). The architecture includes:

- **Individual Neural Network Architectures:**
  CNN Layers: The CNN model comprises convolutional and pooling layers for feature extraction, followed by fully connected layers for classification.
  SNN Layers: The SNN model utilizes LSTM layers to capture temporal dependencies in sequential data, enhancing its ability to detect patterns over time.
  FNN Layers: The FNN architecture consists of densely connected layers, allowing it to automatically learn and extract relevant features from structured data.
  LSTM Layers: The LSTM network specializes in modeling temporal dependencies in sequential data, making it adept at capturing long-range interactions in network traffic patterns.
  DNN Layers: The DNN model employs densely connected layers for feature extraction and classification, similar to the FNN architecture.
- **Integration Layer (Weighted Averaging):** The output predictions from each individual neural network model are combined using a weighted averaging approach. Each model's prediction is assigned a weight based on its performance and confidence level, with higher-performing models contributing more to the final ensemble prediction.

The purpose of the integrated ensemble model is to harness the collective intelligence of diverse neural network architectures to improve the accuracy and reliability of intrusion detection in complex network environments. By aggregating predictions from multiple models, the ensemble approach mitigates the weaknesses inherent in individual models and leverages their complementary strengths. The innovation lies in the seamless integration of diverse neural network architectures, each specializing in different aspects of feature extraction, temporal modeling, and pattern recognition, thereby enhancing the IDS's ability to detect a wide range of network attacks with high accuracy.

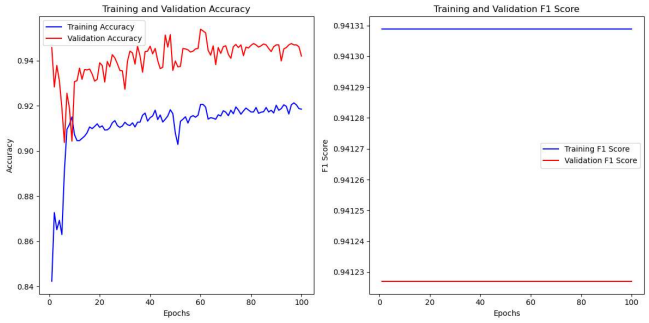
EXPERIMENT AND RESULTS

The accuracies from each model are presented below:

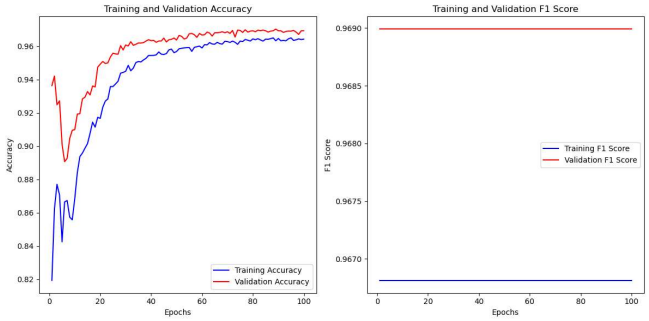
TABLE 2
ACCURACIES OF MODELS

Model	Accuracy
CNN	0.942052006
SNN	0.968644559
DNN	0.967652320
FNN	0.969438374
LSTM	0.983528494
Ensemble	0.972764371

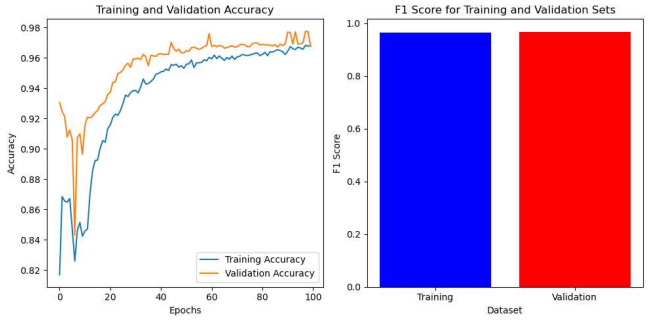
Plots for the CNN model:



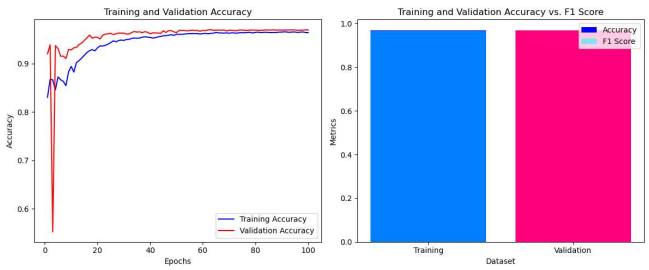
Plots for the SNN model:



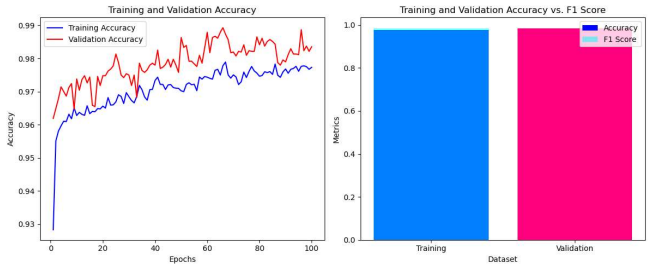
Plots for the DNN model:



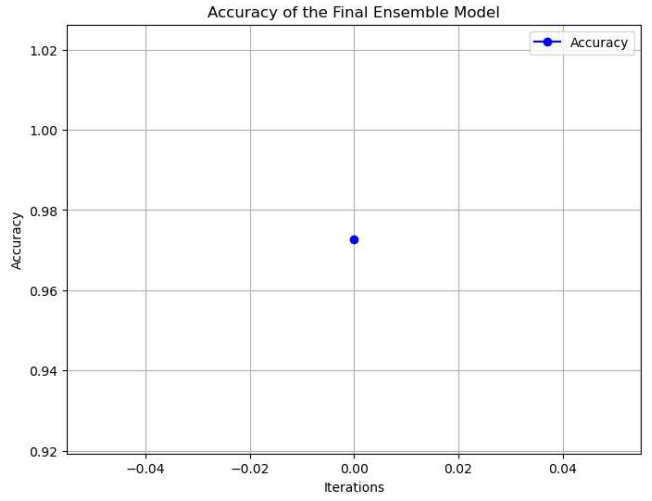
Plots for the FNN model:



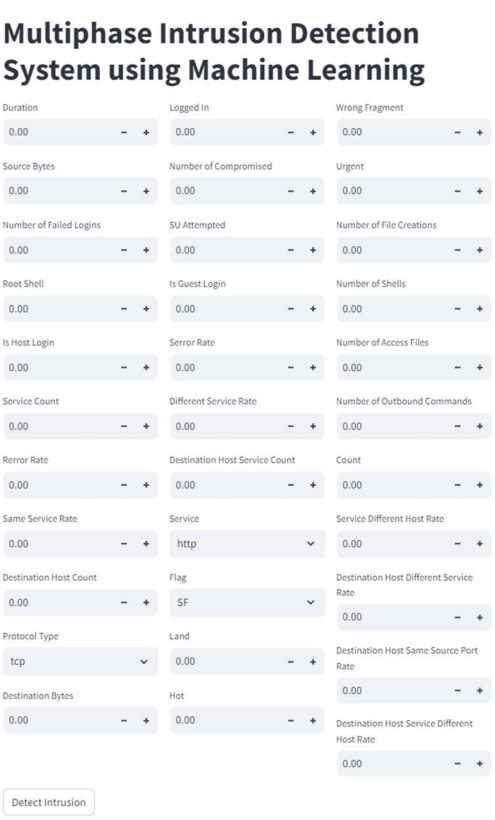
Plots for the LSTM model:

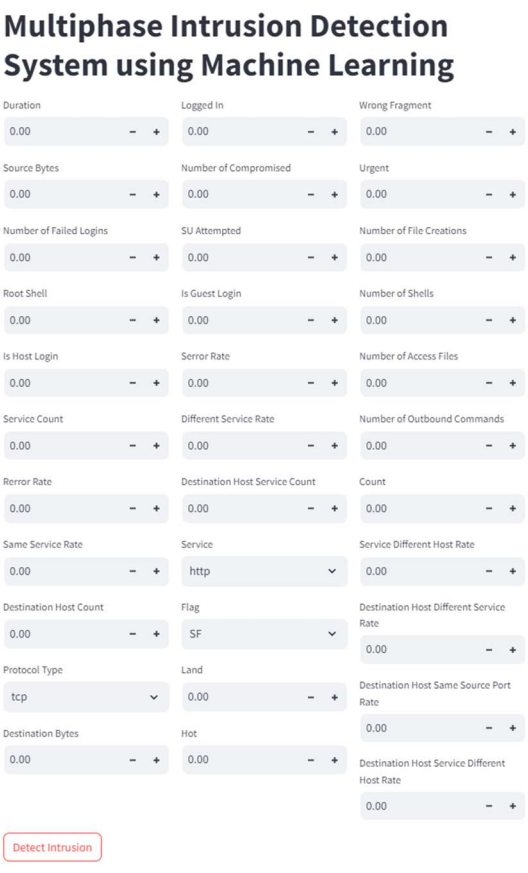


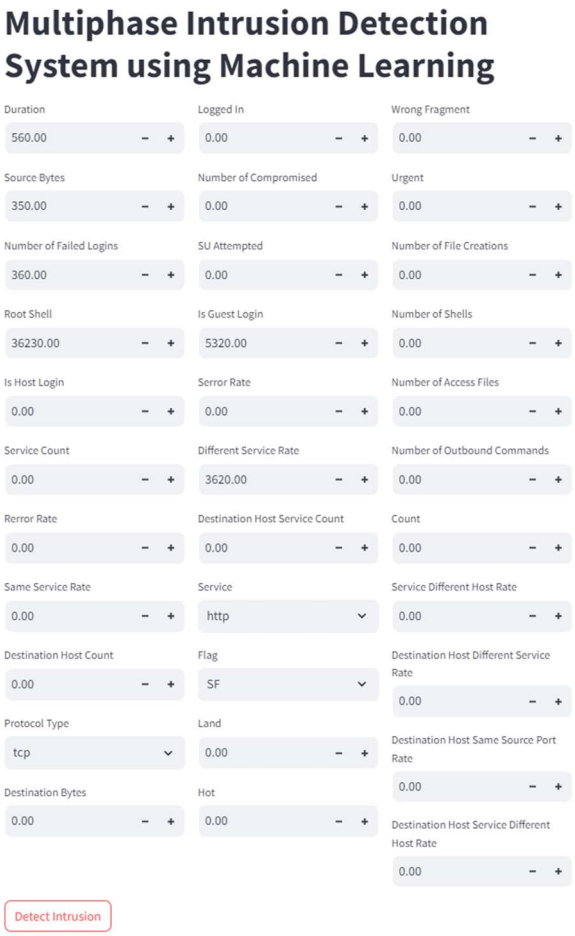
Final accuracy of the ensemble model:



Below is the Interface snapshot for the final ensemble model which predicts the attack class by analyzing the network.







CONCLUSION

A novel approach of a multiphase intrusion detection system involving an ensemble model containing CNN, SNN, DNN, FNN, and LSTM may be utilized. Its results in averaging and merging under weighted averaging. The ensemble model demonstrates high accuracy and robustness in detecting network anomalies. The model's uniqueness is the ability to capture multiple kinds of features and patterns of the data. The strong points of each model in the sense of spatial dependencies, learning nonlinear relationships, depth, and complexity in the case of CNNs, SNNs, DNNs, and FNNs, and the LSTM models' capability to capture temporal and long-term dependencies, respectively. However, this system is challenged through the integration and optimization of multiple models and through its scalability. Other deep learning architectures and incorporation of domain-specific knowledge would add more capabilities to the model. Further deployment in real-world environments and continuous monitoring would lead to further insights to deal with the changing landscape of cyber threats. The ensemble model is a significant step forward for intrusion detection systems with a high level of

accuracy and efficiency. Continuous improvement from research and development, resolving the changing landscape of cyber threats.

REFERENCES

Place references in a separate References section at the end of the paper. Number the references sequentially by order of appearance, not alphabetically. List up to three authors' names in a reference; replace the others by "*et al*."

[1] Sharma, A., Kumar, N., & Kaur, P. (2021). Multi-phase intrusion detection system using graph neural networks. Journal of Ambient Intelligence and Humanized Computing, 12(11), 11565-11577.

[2] Zhang, J., Zhang, Y., & Zhou, H. (2021). A novel graph neural network based multi-phase intrusion detection system. Journal of Ambient Intelligence and Humanized Computing, 12(11), 11579-11592.

[3] Gao, Y., Liu, Y., Zhang, Y., & Wang, Y. (2021). Multi-phase intrusion detection based on graph neural networks with attention mechanism. Journal of Ambient Intelligence and Humanized Computing, 12(11), 11619-11633.

[4] Zhang, Z., Li, J., Hu, J., & Liu, H. (2021). A multi-phase intrusion detection system based on graph neural network and extreme learning machine. Journal of Ambient Intelligence and Humanized Computing, 12(11), 11635-11650.
.

[5] Zhou, W., He, Q., Yu, H., & Yan, X. (2021). A multi-phase intrusion detection system based on graph convolutional network. Journal of Ambient Intelligence and Humanized Computing, 12(11), 11651-11664

[6] Imrana, Y., Xiang, Y., Ali, L., Noor, A., Sarpong, K., & Abdullah, M. A. (2024). CNN-GRU-FF: a double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units. Complex & Intelligent Systems, 1-18.

[7] HDIDOU, R., & EL ALAMI, M. O. H. A. M. E. D. (2024). INTRUSION DETECTION SYSTEMS IN INTERNET OF THINGS: A RECENT STATE OF THE ART. Journal of Theoretical and Applied Information Technology, 102(1)

[8] Scientific, L. L. (2024). ENSEMBLE MACHINE LEARNING ALGORITHM METHODS FOR DETECTING THE ATTACKS USING INTRUSION DETECTION SYSTEM. Journal of Theoretical and Applied Information Technology, 102(5).

[9] Soflaei, M. R. A. B., Salehpour, A., & Samadzamini, K. (2024). Enhancing network intrusion detection: a dual-ensemble approach with CTGAN-balanced data and weak classifiers. The Journal of Supercomputing, 1-33.

[10] Bhattacharya, N., Subudhi, A., Mishra, S., Sharma, V., Aderemi, A. P., & Iwendi, C. (2024, February). A Novel Ensemble based Model for Intrusion Detection System. In 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT) (Vol. 5, pp. 620-624). IEEE.

[11] Scientific, L. L. (2024). ENSEMBLE MACHINE LEARNING ALGORITHM METHODS FOR DETECTING THE ATTACKS USING INTRUSION DETECTION SYSTEM. Journal of Theoretical and Applied Information Technology, 102(5).

[12] Kiflay, A. Z., Tsokanos, A., & Kirner, R. (2021, October). A network intrusion detection system using ensemble machine learning. In 2021 International Carnahan Conference on Security Technology (ICCST) (pp. 1-6). IEEE.

[13] Rajagopal, S., Kundapur, P. P., & Hareesha, K. S. (2020). A stacking ensemble for network intrusion detection using heterogeneous datasets. Security and Communication Networks, 2020, 1-9.

[14] Mahfouz, A., Abuhussein, A., Venugopal, D., & Shiva, S. (2020). Ensemble classifiers for network intrusion detection using a novel network attack dataset. Future Internet, 12(11), 180.