ISSN 2348-0319 International Journal of Innovative and Applied Research [2025]
(Volume 13, Issue 07)
16-26

Journal home page: http://www.journalijiar.com

INTERNATIONAL JOURNAL
OF INNOVATIVE AND APPLIED RESEARCH

RESEARCH ARTICLE

Article DOI: 10.58538/IJIAR/2136
DOI URL: *http://dx.doi.org/10.58538/IJIAR/2136*

DEVEOPMENT OF A DEEP LEARNING BASED FRAMEWORK FOR SECURITY OF IOT NETWORK PROTOCOL

***Osita Miracle Nwakeze[1] and Naveed Uddin Mohammed[2]**

1. Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra Nigeria.

2. Department of Computer Science, Lindsey Wilson University, Columbia, Kentucky, USA.

Manuscript Info

Manuscript History
Received: 19 June 2025
Final Accepted: 23 July 2025
Published: July 2025

Keywords:
Internet of Things (IoT), Feed-Forward Neural Network (FFNN), SMOTE, PCA, Deep Learning

Abstract

This study presents the development of a Feed-Forward Neural Network (FFNN)-based model for security of Internet of Things (IoT) network protocols. The proposed method applied in the execution of the study involves data collection, preprocessing, feature selection, and model training using the CIC-IoT 2022 dataset, which includes normal and attack traffic from various IoT devices. In the study, Synthetic Minority Oversampling Technique (SMOTE) was used for data balancing, Principal Component Analysis (PCA) technique was used for feature selection and hyperparameter optimization were employed to enhance the performance of the model during training. The system was simulated in the NS-3 environment to replicate real-world IoT network conditions, and its effectiveness was evaluated using metrics such as accuracy, precision, recall, and F1-score. The results demonstrated that the FFNN-based model achieves an average validation accuracy of 89.7%, precision of 88.9%, recall of 86.9%, and an F1-score of 87.9%. The system results showcased robustness in detecting various attacks, including DoS, brute force and RTSP attacks in mixed traffic scenarios, meanwhile this study serves as a strong foundation for leveraging deep learning techniques to enhance IoT network security protocols.

****Corresponding Author:-*** **Osita Miracle Nwakeze,** *Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli.*

Introduction:-

Over the past ten years, the number of wireless devices and their uses has increased due to widespread demand for the Internet of Things (IoT). Over 10 billion mobile devices were in use globally in 2020, and this figure is predicted to rise as these networks are used more often (Cisco, 2019). Mobile Ad Hoc Networks (MANETs)-based decentralised wireless communications are acknowledged as the key communication technology for implementing extensive Internet of Things systems for uses like healthcare management (Dey et al., 2017) and agricultural

monitoring (Sahitya et al., 2017). Wireless devices may connect with one another without a Base Station (BS) thanks to MANETs.

By establishing a communication channel between different objects, the Internet of Things (IoT) enhances everyday life. Through this connection, we are able to keep an eye on those items in real time and take the appropriate steps to make the process better. Using multiple levels, an IoT reference model (Bendavid et al., 2018) began with devices/sensors for various purposes and progressed from various technologies like RFID and Bluetooth Low Energy (BLE) to represent empowered individuals and corporate processes that use IoT-enabled data to drive action. However, data and communication at every level of an IoT system's conceptualisation, as well as the links between them, can be impacted by information security threats. Accordingly, every gadget or subsystem needs to be protected (Hosseinzadeh et al., 2023).

Numerous autonomous applications in the fields of healthcare, business solutions, smart cities, home automation, industry automation, and intelligent transportation systems have been made possible by the Internet of Things. The success of the Internet of Things depends on distributed data gathering, aggregation, processing, and analytics, which are usually done through cloud services. As data moves from sensors to services that process and analyse it before delivering the findings to clients or companies that use the analytics data, IoT systems evolve (Cynthia et al., 2019).

Among the new obstacles created by the massive scale of IoT networks are the management of these devices, the total amount of data, connectivity, storage, processing, security, and privacy concerns. Numerous aspects of the Internet of Things, including architecture, communication, applications, protocols, security, and privacy, have been extensively studied. The foundation of IoT technology commercialisation is the assurance of security, privacy, and user pleasure. The IoT increases the amount of threats that attackers may face by utilising enabling technologies like edge computing, software-defined networking, and cloud computing (CC). As a result, it is now difficult and complicated to monitor security as IoT infrastructure develops. To meet the security issues, solutions must take a broad range of factors into account (Jin et al., 2017; Bharati and Podder, 2022). Because nodes join and exit the network in real time, IoT networks are open and dynamic in their topology. Their vulnerability to security risks stems from the absence of centralised network management solutions. According to Liu and Xu (2018), low memory capacity, limited data storage, limited power supply, and connection bandwidth are some of the unique features of IoT devices. These restrictions have a big influence on how well security procedures work for IoT infrastructures in terms of expansion and functionality. Therefore, the increasing overhead that demands compute resources makes it difficult to create an intrusion detection system that works for an IoT network (Bakhsh et al., 2023). As hackers use advanced methods to steal confidential information while avoiding detection by intrusion detection systems, cyberattacks are becoming more complex and challenging to detect. Additionally, there are cybersecurity hazards associated with inter-network connectivity. Innovative techniques are therefore essential for prompt intrusion detection and assault prevention strategies. Lately, intrusion detection, network anomaly detection, and network protocol protection have been accomplished via the use of Machine Learning (ML) and Deep Learning (DL) algorithms (Khan et al., 2022).

DL frameworks have gained popularity as a means of detecting network breaches. The literature needs an objective comparison of various deep learning models, particularly in light of new intrusion detection datasets, even if several surveys cover the emerging field of study on this subject (Awajao, 2023). In the modern world, cybersecurity is a major issue. For instance, IDS search for signs of malicious activity, whereas firewalls are used to safeguard critical data. The fast proliferation of Artificial Intelligence (AI) research has led to significant advancements in approaches like anomaly detection and pattern recognition (Yadav et al., 2022). A viable tactic for mitigating cybersecurity risks and guaranteeing security is artificial intelligence (Abdullahi et al., 2022).

In order to analyse vast volumes of data, identify patterns and correlations, and categorise the data based on predetermined standards, Deep Neural Networks (DNN) employ computer resources. A promising technique for anomaly-based intrusion detection in IoT security is the use of DNNs to recognise and categorise data in properly trained DNNs. This study proposes an intrusion detection system for the IoT that uses deep neural networks to detect anomalies in real-time data. The goal of this study is to use Feed Forward Neural Networks (FFNN) models to create an anomaly detection system for the IoT network protocol. In order to obtain successful intrusion detection performance, the research presents a framework for implementing a Deep Learning approach for intrusion detection within IoT networks.

The Proposed Research Method

In order to detect and categorise any weaknesses and assaults in IoT networks, this section offers a framework for deep learning-based intrusion detection in IoT protocols. The Feed-Forward Neural Network (FFNN) model's proposed framework is introduced, and then each step's important contributions are thoroughly examined and the tasks completed at each stage are thoroughly evaluated. The intrusion detection dataset, software, and libraries used in this study are thoroughly explained in the discussion that follows. After enough training, the suggested system can detect intrusions and recover packets from the data generated by the underlying IoT infrastructure. In order to gather the traffic flow of IoT devices, the suggested architecture calls for using a traffic capture technique from an IoT network, as suggested in Dadkhah et al. (2022). The following steps make up the process: data collecting, data preparation, which is crucial to data analysis since it includes feature scaling, encoding, cleaning, and dataset extraction. Class balance is accomplished by data augmentation approaches, and the best features are found through feature selection. The first stage of feature preprocessing involves splitting the dataset, after which FFNN models are trained, validated, and tested.

Data Collection:-

The CIC-IOT 2022 dataset was utilised in this study to detect intrusions. Using other protocols, the dataset and testbed configuration described in (Dadkhah et al., 2022) may be utilised for IoT device vulnerability evaluation, behavioural analysis, and profiling. IoT device network traffic from several domains was used to collect the dataset. Devices from the Canadian Institute for Cybersecurity (CIC) lab included cameras, audio equipment, and home automation systems. Profiling 60 IoT devices linked to the IoT network was necessary to create the dataset. The dataset is made up of packet headers that were gathered from every device in a variety of conditions, such as when it was turned on, idle, active, and during interactions, using a series of IoT traffic capture experiments over a 30-day period. IoT devices were subjected to a denial-of-service (DoS) attack via HTTP, UDP, and TCP flooding using the Low Orbit Ion Cannon (LOIC) tool. To assess the behaviour of IoT devices, brute-force and Camera Real Time Streaming Protocol (RTSP) URL assaults were conducted using the Hydra and Nmap tools.

Data Cleaning

Finding and fixing mistakes, insufficient organisation, duplication, or missing values in a dataset is known as data cleaning. When integrating several data sources, there are several ways that data duplication or inaccurate categorisation might happen (Kosongo and Sun, 2020). Anomalies in datasets might result from the absence of a generally recognised approach for specifying each step of the data cleaning procedure. Even when findings and algorithms are reliable, the existence of erroneous data might jeopardise their dependability. In order to guarantee accuracy and consistency across all iterations, a standardised structure for the data cleaning process must be established. Duplicate packets are dropped, and missing and infinite values are eliminated from the CIC IOT 2022 dataset utilised in this study.

Categorical Encoding

Label encoding is a commonly utilised approach in DL that transforms categorical input into numerical data. It entails giving each unique category variable a numerical value so that algorithms may handle the data effectively. The values of the dataset's absolute features, for instance, are “HTTPFlood”, “UDPFlood”, and “TCPFlood”. Label encoding is used in this stage to give the features the appropriate numerical values. One-Hot encoding, which represents categorical variables as binary vectors, is a further technique for converting categorical data into numerical data (Hancock and Khoshgoftaar, 2020). This method creates a binary vector for each category in the variable, as was previously explained. All values in the vector are zeros, with the exception of the index value, which is set to one for the matching class.

Feature scaling

The process of altering a collection of features' value range in accordance with a predetermined range is known as feature scaling. When a feature has a high value but has little effect on other features, this approach must be applied (Kosongo and Sun, 2020). Normalisation and standardisation are part of this (Kunang et al., 2021). To lessen the problem of notable variations throughout the merging process, normalisation is an essential step. In order to enable optimisation, all features are scaled to a standard range. Column values can be expressed on a consistent scale by normalising them using Min-Max normalisation with a chosen range of [-1,1].

Data Augmentation and Balancing

When samples are not evenly distributed among classes, the issue of unbalanced data arises. One class becomes unbalanced, causing a skewed dataset that might lead to a biassed model. Prior to classification, the training data is often re-sampled, increasing the number of samples in the minority class, and under-sampled, decreasing the number of samples in the majority class. Because the minority class typically has inadequate data, imbalanced classification problems are commonly encountered in a variety of datasets. The synthesis of new data from the minority class is one way to address class imbalance and mitigate the problem of restricted data availability.

One popular method for creating extra samples in the dataset is the Synthetic Minority Oversampling Technique (SMOTE). According to Elreedy and Atiya (2019), the approach is based on creating data packets that link a certain location with its K-Nearest Neighbouring (KNN) points. To achieve a balanced dataset, the SMOTE approach creates new samples from the current dataset. Increasing the number of cases of the minority groups is its purpose. Additionally, an analysis is carried out to assess how several aspects, such as dimensionality reduction, training set size, and number of neighbours (K), affect the system's accuracy. A qualitative study also evaluates the factors influencing the outcomes.

Because of the dataset's imbalance, a resampling strategy is used before to classification in order to lessen the possibility of problems resulting from class imbalance. By oversampling the minority class and under-sampling the majority class, the dataset is balanced. SMOTE analysis of the dataset resulted in resampled data for the X and Y variables. Eight classes were used in the experiment: three Normal classes (Power, Idle, and Interaction) and five Benign classes (HTTP Flood, UDP Flood, TCP Flood, RSTP, and Brute Force). SMOTE analysis was performed on 465101 packets from the dataset, which comprises binary and multiclass categories.

Feature Selection

One method for preserving characteristics in data is feature selection, which lowers the dimensionality of the data. It reduces computing complexity and improves storage economy. Low-dimensional features are necessary for both binary and multiclass classification in the Intrusion Detection System (IDS) classifier design process because of the power constraints of IoT devices. This strategy has produced remarkable results, as evidenced by empirical data. By choosing and extracting pertinent features, Principal Component Analysis (PCA) is utilised in this study to transform the original dataset into a lower-dimensional subspace while maintaining the essential properties of the original data.

PCA is a statistical method that uses orthogonal combinations of the original parameters that exhibit high variance in order to minimise the number of features in a dataset (Abdulhammed et al., 2019). Since the principal components are unrelated to one another, correlated elements that have a negligible impact on the decision-making process are eliminated. According to Bhattacharya et al. (2020), the primary phases in PCA include calculating the mean, standard deviation, covariance, cumulative percentage, eigen-vectors, and eigen-values. The dataset's relevance informs the selection of these pairings. The original variables are combined linearly to create the principal components, which are able to capture the greatest amount of volatility in the dataset.

Feed Forward Neural Network Algorithm

A Feed Forward Neural Network is an artificial Neural Network in which the nodes are linked circularly. In contrast to a recurrent neural network, a feed-forward neural network cycles some of its routes. The feed-forward model is the basic type of neural network because the input is only processed in one direction. The data always goes in one direction and never backwards/opposite. A weight is applied to each input to an artificial neuron. Prior to applying a bias to the result, the inputs are multiplied by their respective weights. The weighted total is then sent via a non-linear function called an activation function (Sharma, 2024). The architecture of the FFNN algorithm is shown in Figure 1.

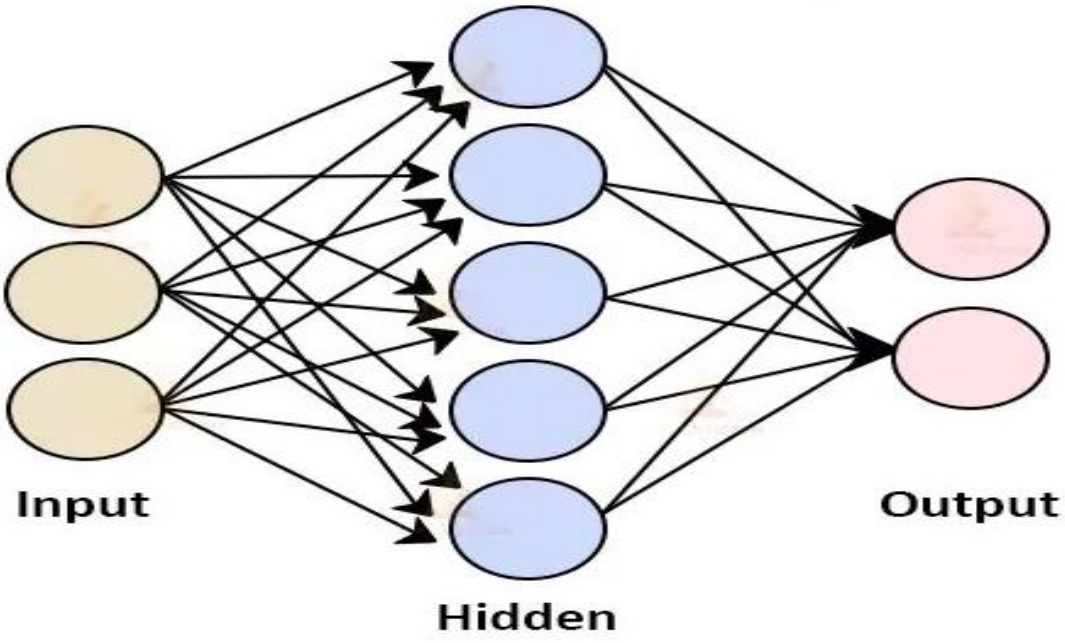

Figure 1:- Simple Architecture of the Feedforward Neural Network (FFNN) (Sharma, 2024).

As seen in Figure 1, the input layer is the initial layer and looks to include six neurones, but it is actually only the data that is sent into the neural network. The last layer is called the output layer. The number of neurones in the first and final layers depends on the task type and the dataset. The number of neurones in the hidden layers and the total number of hidden layers will be decided by trial and error. The first neurone from the first hidden layer will be coupled to every input from the preceding layer. All of the inputs from the previous layer will be linked to the second neurone in the first hidden layer, and the same is true for all of the neurones in the first hidden layer. Neurones of the second hidden layer use the outputs of the first layer as inputs, and each of these neurones is connected to every prior neurone.

Architecture Of The Proposed Ffnn Model

To accomplish hyperparameter optimisation, the FFNN architecture shown in Figure 2 makes use of RandomizedSearchCV, the KerasClassifier wrapper for the Keras library, and Scikit learns. Hyperparameter optimisation is used to determine the best set of hyperparameters given the dataset that is provided. The neural network's design is determined by four hyperparameters: the number of hidden layers, the number of neurones in each hidden layer, the dropout rate to prevent overfitting, and the L2 regularisation coefficient. An adequate selection of hyperparameters is used to modify the model in order to increase classification accuracy and decrease the likelihood of overfitting (Sharma et al., 2023). Following each hidden layer, the dropout regularisation technique is used to reduce overfitting.

There are 67 features for multiclass classification and 65 features for binary class in the input layer. To find the ideal combination of hyperparameters to identify intrusions in the dataset, the algorithm does a randomised search across them. To assess the model's performance, a cross-validation method with three folds and 10 iterations was used. Through a methodical process of iterating through different hyperparameter configurations, the model uses training and validation sets to determine the optimal combination that maximises performance using RandomizedSearchCV in less processing time. Time and effort are saved since human tuning is no longer necessary due to the automatic optimisation of the model's architecture and regularisation parameters. During the search phase, several combinations of hyperparameters were tested for every layer. The optimal model hyperparameter configuration is chosen after the search is complete.

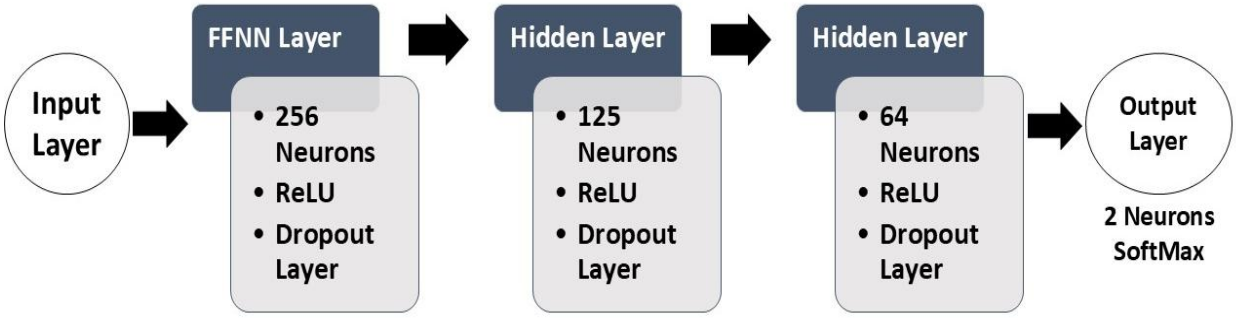

Figure 2:- Architecture of the Proposed FFNN Model.

Three densely linked hidden layers of 256, 125, and 64 neurones make up the proposed FFNN architecture for binary and multiclass classification, which is depicted in Figure 2. Two neurones make up the model output layer, which represents different dataset categories. A popular method for dealing with multiclass issues is to employ the SoftMax activation function in the final layer and the ReLU activation function for hidden layers. The FFNN model facilitates weight updating for training by utilising the categorical cross-entropy loss function and the Adam optimiser. The L2 regularisation approach, a weight loss of 0.001, and a dropout rate of 0.1 for both binary and multiclass classification are used to refine the model. 100 epochs with 32, 64, and 128 batch sizes were used to train the model. Furthermore, several hidden layer and neuronal combinations were investigated.

Training of the FFNN Model

Training a FFNN algorithm in the Google Colab environment starts with setting up the workspace and preparing the dataset to be fed into the model for training. After initiating the necessary libraries, the dataset (CIC-IoT 2022 dataset) is loaded and pre-processed through data cleaning, categorical encoding, feature scaling, data augmentation and balancing and feature selection. The dataset is then split into training and testing sets using train_test_split, then the FFNN model is defined using TensorFlow, with an input layer matching the number of features. The model is compiled with an optimizer like Adam, a loss function such as categorical cross-entropy and metrics like accuracy.

The training process begins by converting labels to categorical format if necessary and training the model on the pre-processed dataset. After training, the model's performance is evaluated on the test set and metrics like performance accuracies for testing and validation sets are displayed. Finally, the trained model is saved in HDF5 format for deployment. This implemented process in Google Colab ensures efficient training and evaluation of the FFNN model, making the model a powerful tool for integration in intrusion detection systems for IoT network protocol.

Integration of the Trained FFNN Model in IoT Network Protocol

Integrating the trainedFFNN-based intrusion detection model into an IoT network protocol involves a structured process to ensure seamless operation, real-time threat detection and minimal disruption to the IoT infrastructure. The architecture of the network includes IoT devices, an edge gateway for data collection and preprocessing and a cloud server for centralized analysis. The FFNN model is deployed in the cloud for large-scale data processing. Network traffic from IoT devices is captured using tools like Wireshark and relevant features such as packet size, protocol type, and source/destination IP are extracted. The data is then pre-processed to clean redundant data, encode categorical features into numerical values and scale features to a standard range. Feature selection techniques (PCA) is applied to reduce dimensionality and retain only the most relevant features. This preprocessing ensures that the data fed into the FFNN model is optimized for accurate anomaly detection.

The FFNN model which has been in the cloud where it can handle massive data volumes and perform complex computations uses the pre-processed data to detect intrusions such as HTTP Flood, UDP Flood, brute force attacks etc. When an intrusion is detected, the system generates alerts and shuts down the IoT network system briefly, then the source IP is blocked and the affected device is isolated to mitigate threats. The process diagram of the integrated system is presented in Figure 3.

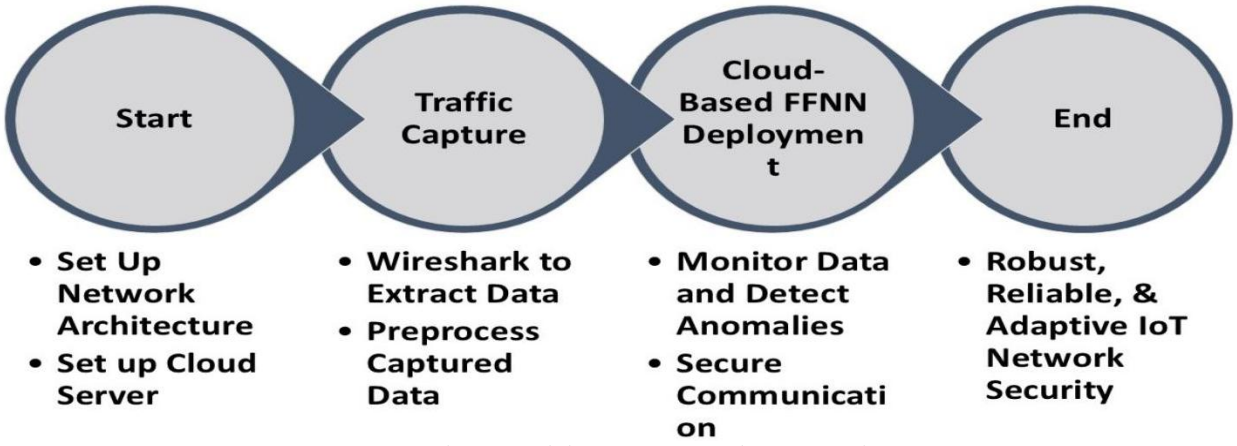

Figure 3:- Process Diagram of the FFNN-Based IoT Security System.

The integration process also includes performance monitoring and model retraining to ensure the system remains effective over time. The FFNN model's performance is continuously evaluated using metrics like accuracy, precision, recall, and F1-score. False positives and false negatives are logged for further analysis, and the model is periodically retrained using new data to adapt to evolving threats. Finally, all communication between IoT devices and the cloud is encrypted using protocols like Secure Socket Layer (SSL) to prevent interception. By addressing challenges such as resource constraints, real-time processing, and scalability, the integrated FFNN-based intrusion detection system provides a robust and efficient solution for securing IoT networks. This process ensures that IoT devices and networks are protected from a wide range of threats while maintaining high performance, reliability and adaptability.

System Implementation

The implementation of the trained FFNN-based model for intrusion detection was implemented in the NS-3 environment which involves creating a simulated IoT network with devices, traffic generators and attack scenarios. The NS-3 application is developed to capture and preprocess network traffic, feed it into the FFNN model, and classify it as normal or malicious. The system's performance is evaluated using metrics like accuracy, precision, recall, and detection time, with results visualized using Matplotlib. The simulation allows for iterative improvement of the intrusion detection by retraining the model and refining the network setup, ensuring robust and accurate intrusion detection before deployment in real-world IoT networks.

System Results:-

The results of the FFNN-based IoT network protocol attack detection implemented in NS-3 environment which demonstrate its effectiveness in securing IoT network protocol. This section presents the training results of implementing the FFNN-based model for intrusion detection over 10 epochs which can be seen in Table 1. The table considers key metrics such as training accuracy, validation accuracy, training loss and validation loss for each epoch. These metrics are commonly used to evaluate the performance of deep learning models during training.

Table 1:- System Implementation Results.

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
1	0.85	0.83	0.45	0.47
2	0.88	0.86	0.38	0.40
3	0.90	0.88	0.32	0.35
4	0.92	0.89	0.28	0.31
5	0.93	0.90	0.25	0.29
6	0.94	0.91	0.22	0.27
7	0.95	0.92	0.20	0.25
8	0.96	0.92	0.18	0.24
9	0.96	0.93	0.16	0.23
10	0.97	0.93	0.15	0.22
Average	**0.926**	**0.897**	**0.259**	**0.303**

The system results in Table 1 achieves average training accuracy of 0.926 and validation accuracy of 0.897 in classifying network traffic as malicious or non-malicious with strong performance across various attack situations such as Denial-of-Service (DoS), brute force, and RTSP attacks. Figure 4 presents the graphical representation of the training and validation accuracy results of the system implementation while Figure 5 presents the Loss graph of the of the system implementation.

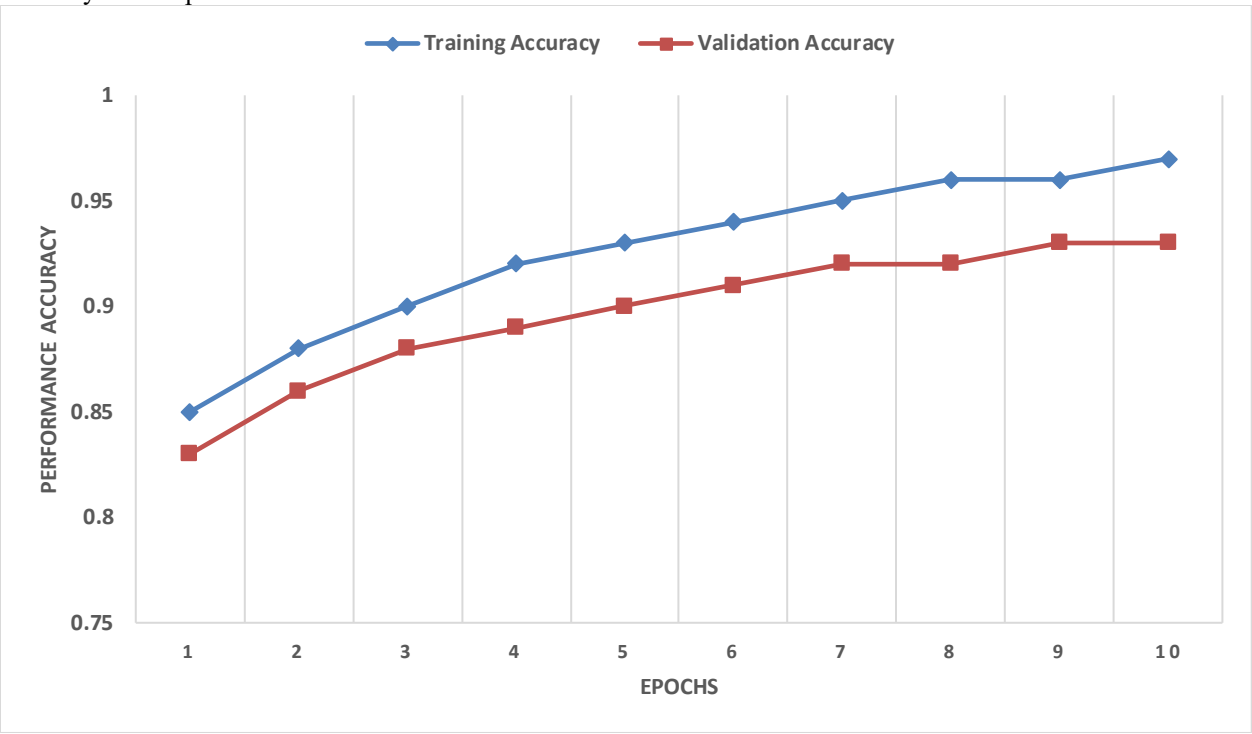

Figure 4:- System Performance Accuracies.

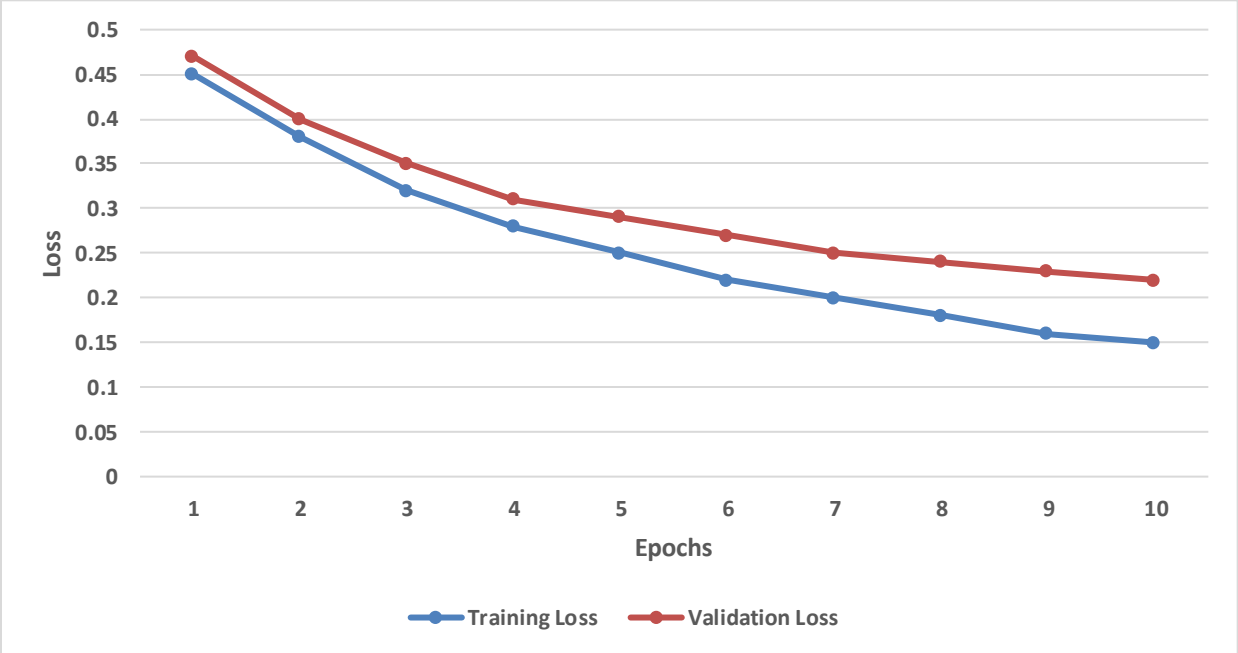

Figure 5:- System Performance Losses.

In addition to accuracy and loss, other importance performance metrics such as recall, precision and F1-score which are essential for evaluating the performance of the FFNN-based model is presented in Table 2. These metrics provide a more comprehensive understanding of the model's ability to detect anomalies. Table 2 is the presentation of these metrics performance in 10 iterations of validation operation.

Results Table for Recall, Precision, and F1-Score

Iteration	Precision	Recall	F1-Score
1	0.82	0.80	0.81
2	0.84	0.82	0.83
3	0.86	0.84	0.85
4	0.88	0.86	0.87
5	0.89	0.87	0.88
6	0.90	0.88	0.89
7	0.91	0.89	0.90
8	0.92	0.90	0.91
9	0.93	0.91	0.92
10	0.94	0.92	0.93
Average	**0.889**	**0.869**	**0.879**

The results attained in Table 2 demonstrates improvement in precision, recall, and F1-score in each of the iterations. The precision results attained increased from 0.82 to 0.94 to make up an average of 0.889 and recall improves from 0.80 to 0.92 arriving at an average of 0.869. Finally, the F1-score which balances precision and recall rose from 0.81 to 0.93 attaining an average of 0.879. This robust performance underscores the model's suitability for real-world IoT environments where both precision and recall are essential for maintaining security and operational efficiency.

Conclusion:-

This study focused on the development Feed-Forward Neural Network (FFNN)-based model for the detection of anomalies in IoT network protocol. The proposed methodology adopted involves the application of data collection, preprocessing, feature selection and model training using acquired CIC-IoT 2022 dataset which is made up of normal and attack traffic from various IoT devices. The FFNN model was fine-tuned using techniques like SMOTE for data balancing, PCA for feature selection and hyperparameter optimization to enhance performance during training. The system was simulated in the NS-3 environment to replicate real-world IoT network conditions and the system's effectiveness was evaluated using metrics such as accuracy, precision, recall and F1-score. The results demonstrated that the FFNN-based model achieves an average validation accuracy of 89.7%, precision of 88.9%, recall of 86.9%, and F1-score 87.9%. The system showcased its robustness in detecting various attacks, including DoS, brute force, and RTSP attacks, even in mixed traffic scenarios.

The high performance across these key metrics combined demonstrates the model's suitability for real-world deployment. However, the performance of the system may degrade under extremely high traffic loads or with highly imbalanced datasets and it will require periodic retraining of the model with different datasets to adapt to evolving threats. Hence, the study recommends that future research work could explore advanced deep learning architectures like Long Short-Term Memory (LSTM) or Convolutional Neural Networks (CNNs) to improve the system's support for additional IoT protocols and devices. Overall, this study provides a strong foundation for leveraging deep learning techniques to enhance IoT network security protocols offering an efficient solution for real-time intrusion detection.

References:-

1. Abdulhammed, R., Faezipour, M., Musafer, H., & Abuzneid, A. (2019). Efficient network intrusion detection using PCA-based dimensionality reduction of features. In 2019 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1–6). IEEE. https://doi.org/10.1109/ISNCC.2019.8909142
2. Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F., & Abdulkadir, S. J. (2022). Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. Electronics, 11(2), 198. https://doi.org/10.3390/electronics11020198
3. Awajan, A. (2023). A novel deep learning-based intrusion detection system for IoT networks. Computers, 12(2), 34. https://doi.org/10.3390/computers12020034.
4. Bakhsh, S. A., Khan, M. A., Ahmed, F., Alshehri, M., Ali, H., & Ahmed, F. (2023). Enhancing IoT network security through deep learning-powered Intrusion Detection System. Internet of Things, 24, 100936. https://doi.org/10.1016/j.iot.2023.100936
5. Bendavid, Y., Bagheri, N., Safkhani, M., & Rostampour, S. (2018). IoT device security: Challenging "A lightweight RFID mutual authentication protocol based on physical unclonable function." Sensors, 18(12), 4444. https://doi.org/10.3390/s18124444
6. Bharati, S., & Podder, P. (2022). Machine and deep learning for IoT security and privacy: Applications, challenges, and future directions. Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka-1205, Bangladesh.
7. Bhattacharya, S., Maddikunta, P. K. R., Kaluri, R., Singh, S., Gadekallu, T. R., Alazab, M., & Tariq, U. (2020). A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU. Electronics, 9(2), 219. https://doi.org/10.3390/electronics9020219
8. Cisco. (2019). Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022. Cisco: San Jose, CA, USA.
9. Cynthia, J., Sultana, H. P., Saroja, M. N., & Senthil, J. (2019). Security protocols for IoT. In Ubiquitous Computing and Computing Security of IoT (pp. 1–20). Springer. https://doi.org/10.1007/978-3-030-01566-4_1
10. Dadkhah, S., Mahdikhani, H., Danso, P. K., Zohourian, A., Truong, K. A., & Ghorbani, A. A. (2022). Towards the development of a realistic multidimensional IoT profiling dataset. In 2022 19th Annual International Conference on Privacy, Security & Trust (PST) (pp. 1–11). IEEE. https://doi.org/10.1109/PST55820.2022.9851966
11. Dey, N., Ashour, A. S., Shi, F., Fong, S. J., & Sherratt, R. S. (2017). Developing residential wireless sensor networks for ECG healthcare monitoring. IEEE Transactions on Consumer Electronics, 63(4), 442–449. https://doi.org/10.1109/TCE.2017.015063
12. Elreedy, D., & Atiya, A. F. (2019). A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. Information Sciences, 505, 32–64. https://doi.org/10.1016/j.ins.2019.07.070
13. Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. Journal of Big Data, 7(1), 1–41. https://doi.org/10.1186/s40537-020-00305-w
14. Hosseinzadeh, M., Hussain Malik, M., Safkhani, M., Bagheri, N., Le, Q. H., Tightiz, L., & Mosavi, A. H. (2023). Toward designing a secure authentication protocol for IoT environments. Sustainability, 15(7), 5934. https://doi.org/10.3390/su15075934
15. Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data, 7(1), 1–20. https://doi.org/10.1186/s40537-020-00379-6
16. Khan, A. R., Kashif, M., Jhaveri, R. H., Raut, R., Saba, T., & Bahaj, S. A. (2022). Deep learning for intrusion detection and security of Internet of things (IoT): Current analysis, challenges, and possible solutions. Security and Communication Networks, 2022, 1–15. https://doi.org/10.1155/2022/1234567
17. Kunang, Y. N., Nurmaini, S., Stiawan, D., & Suprapto, B. Y. (2021). Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. Journal of Information Security and Applications, 58, 102804. https://doi.org/10.1016/j.jisa.2021.102804
18. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet of Things Journal, 4(5), 1125–1142. https://doi.org/10.1109/JIOT.2017.2683200
19. Lu, Y., & Xu, L. D. (2018). Internet of things (IoT) cybersecurity research: A review of current research topics. IEEE Internet of Things Journal, 6(2), 2103–2115. https://doi.org/10.1109/JIOT.2018.2869847

20. Sahitya, G., Balaji, N., Naidu, C. D., & Abinaya, S. (2017). Designing a wireless sensor network for precision agriculture using ZigBee. In Proceedings of the 2017 IEEE 7th International Advance Computing Conference (IACC) (pp. 287–291). Hyderabad, India. https://doi.org/10.1109/IACC.2017.0085
21. Sharma, B., Sharma, L., Lal, C., & Roy, S. (2023). Anomaly based network intrusion detection for IoT attacks using deep learning technique. Computers & Electrical Engineering, 107, 108626. https://doi.org/10.1016/j.compeleceng.2023.108626
22. Sharma, P. (2024). Introduction to feed-forward neural network in deep learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/03/basic-introduction-to-feed-forward-network-in-deep-learning/
23. Yadav, N., Pande, S., Khamparia, A., & Gupta, D. (2022). Intrusion detection system on IoT with 5G network using deep learning. Wireless Communications and Mobile Computing, 2022, 1–13. https://doi.org/10.1155/2022/9876543.