A Comprehensive Study of Advanced Deep and Recurrent Neural Networks for Anomaly-based Intrusion Detection Systems

Nafiz Mahamud[1] and Minhee Jun[1]

[1]Electrical Engineering and Computer Science, The Catholic University of America, Washington, DC, USA
Email: {mahamad, junm}@cua.edu

***Abstract*—The increasing prevalence of IoT devices, alongside AI agents, emphasizes their essential role in shaping the future. However, the limited capacity of IoT devices makes them particularly vulnerable to cyber-security threats. To address this, efforts have been made to protect IoT devices from cyber threats by developing robust Intrusion Detection Systems (IDS). Although earlier IDS models demonstrated reasonable detection accuracy, they were impractical due to high false rates, leading to manual or semi-automatized management. To address these limitations, this paper focuses on an anomaly-based IDS designed to reduce specifically false negatives and improve detection accuracy. This paper also provides a comprehensive analysis of advanced deep learning models utilized for anomaly-based IDS, including Deep Neural Networks (DNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Autoencoders (AE). Using the IoT-23 dataset for a real-world environment, simulation results show that GRU and LSTM outperform other models achieving low false positive rates and nearly zero false negatives while maintaining high accuracy. Additionally, integrating GRU and LSTM into the AE architecture significantly improves the reliability and robustness of intrusion detection in IoT networks. Furthermore, the DNN-Random forest hybrid model proved effective for feature extraction and classification. In conclusion, our study presents an anomaly-based IDS utilizing the proposed AI models of deep learning, making it a practical solution for fully automatized IDS for the security of future IoT devices and networks.**

***Index Terms*—Cyber security, Internet of Things (IoT), Intrusion detection systems (IDS), Machine learning (ML), Deep learning (DL), Anomaly detection, Autoencoders (AE).**

I. INTRODUCTION

ADVANCEMENTS in wireless communications are rapidly evolving, enhancing global connectivity each year. Modern industries such as automotive transport, smart agriculture, and public safety are adopting new technologies and standards, including Internet-of-Things (IoT) devices as given in Figure 1. However, IoT-based system faces security risks and challenges at every architectural level of the sensing layer, the network layer, the data processing layer, and the application layer [1] as given in Figure 2. For instance, the sensing layer is exposed to threats such as malicious code injection, eavesdropping, and interference [2], [3]. The network layer is vulnerable to spoofing, denial of service, man-in-the-middle, and routing information attacks [3]. The application layer is susceptible to viruses, worms, and phishing attempts. Particularly, a botnet represents a targeted form of cyber-attack leveraging IoT devices. Angrishi's study in [4] defines a botnet as a large collection of internet-connected devices manipulated to inundate a specified server (or servers) with simultaneous requests, rendering it incapable of responding to genuine requests, effectively halting its operation. This assault constitutes a distributed denial of service (DDoS) attack as given in Figure 3. These attacks have become increasingly sophisticated, making detection challenging [4]. IoT botnets pose a threat not only to the owners of IoT devices but also to all internet users. Because DDoS attacks require substantial network traffic to disrupt services, IoT devices are ideal hosts due to their vast numbers and generally weak security measures, rendering them easy targets [5]. In their research, Das et al. [6] illustrate Mirai, a recent example of such a botnet, wherein a virus seeks out susceptible devices and links them to Command-and-Control Servers (C&C servers). Das et al. [6] and Tushir et al. [7] observed that IoT devices linked to Mirai Botnets are primarily utilized to execute DDoS assaults on targeted devices. Tushir et al. [8] investigated the impact of Mirai attacks on IoT devices, revealing a 40% surge in energy consumption and a 50% increase in storage usage. Subsequently, numerous iterations and variants of the Mirai botnet emerged, such as Persirai, Hajime, and BrickerBot [8]. Targets of DDoS attacks encompassed websites, cloud providers, individuals, educational institutions, telecommunication companies, and DNS providers (Dyn), which serviced multiple websites including Reddit, Amazon, Spotify, and Airbnb, among others [5]. As per Statista [9], the global count of IoT-connected devices reached nearly 8.74 billion in 2021, with a Cisco white paper [10] projecting a rise to approximately 30 billion by 2023, compared to roughly 18 billion in 2018. By 2024, it's estimated that there will be 83 billion devices connected to the Internet of Things (IoT) [11] as shown in Figure 4. Moreover, the same paper [10] anticipates a surge in Distributed Denial of Service (DDoS) attacks to around 15 million by 2023, contrasting with 7 million recorded in 2018 [12]. According to statistics, the frequency of attacks is doubling annually, resulting in significant financial losses, amounting to tens of millions of dollars specifically from ransomware attacks [13].

One effective strategy for thwarting such assaults involves implementing a robust Intrusion Detection System (IDS) [13] capable of identifying various forms of intrusion as shown in Figure 5. Presently, IDSs employ two main detection approaches: signature-based and anomaly-based. Signature-based detection systems are hindered in their effectiveness by their incapacity to recognize emerging cyber threats and their reliance on manual updates to the signature database which can be laborious. Conversely, anomaly-based methods analyze data, relying on the system's comprehension of typical behavior to flag any incoming connections that appear aberrant.

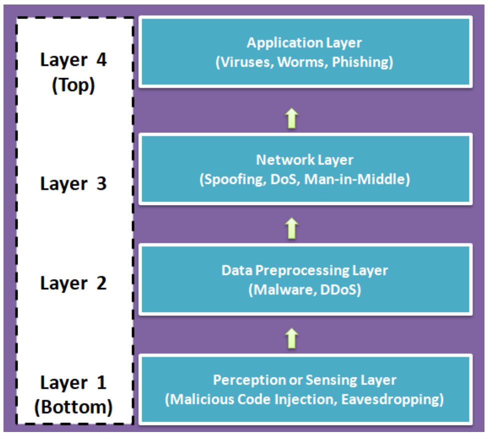

Figure 1. IoT Arch.

However, previous approaches to anomaly detection in security systems were either manual or semi-automated, requiring significant time and expertise. To improve accuracy and efficiency, integrating AI into the process offers great potential [14]. Network intrusion detection detection seeks to assess diverse network data using different behavioral analyses to uphold its security.

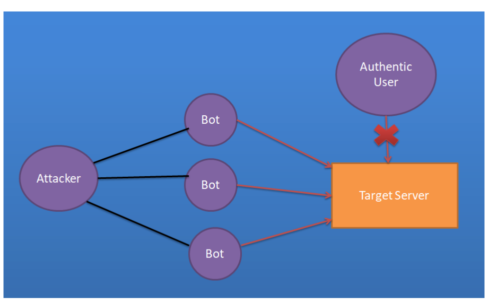

Figure 2. DDoS Attack

Numerous methods exist for detecting anomalies in networks. Although machine learning has proven indispensable and efficient in promptly identifying cyber-attacks, Deep learning using extensive datasets for training can potentially avoid overfitting issues, as it possesses greater capacity for generalization compared to conventional learning models [12]. To train an AI model effectively, high-quality, large-scale data from IoT devices are essential. However, IoT devices are vulnerable to cybersecurity threats. By combining IoT's real-time data collection with AI's data analysis and decision-making capabilities, organizations can develop more responsive, adaptive, and efficient systems across various sectors. Handling real-time data in security AI systems requires not only accuracy but also a low false rate. However, previous studies on anomaly detection have not fully met this criterion [15]. Although several anomaly detection techniques are employed, there have been fewer comparative studies of different deep learning models for anomaly detection. Since intrusions entail a sequence of linked malevolent actions executed by an internal or external perpetrator to compromise the security of the designated system [16], our attention will be directed toward Recurrent Neural Networks, primarily tailored for sequential data processing. During training, the RNN is fed sequences of data where, at each time step, it learns to predict the next item in the sequence and its hidden state acts as a form of memory, enabling the RNN to capture patterns and dependencies over time.

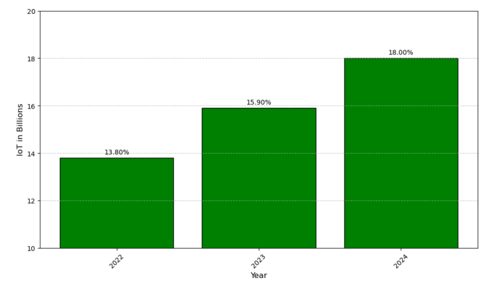

Figure 3. IoT Device Numbers

Constructing an IDS for automatic cyber-attack detection necessitates a suitable dataset for training. We are going to explore the IoT-23 dataset by Garcia et al. [17] a recent release specifically tailored to address cyber-attacks involving IoT devices, introduced in early 2020. Our proposed IDS uses several deep learning-based models for binary classification. There are several factors for selecting binary classification. First, binary classification simplifies the problem of distinguishing between normal (benign) and abnormal (malicious) activities. This can make the system easier to design, implement, and maintain.

Second, since the model only needs to learn two classes, training and prediction can be faster and require fewer computational resources. Finally, by focusing on identifying anomalies as a whole, the IDS can be more robust in catching new or unknown types of attacks that deviate significantly from normal behavior. Therefore, this article makes the following significant contributions:

Particularly, this paper presents a comprehensive analysis of First, this paper provides a comprehensive comparative analysis of advanced deep learning models, including Deep Neural Networks (DNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) for intrusion detection using the IoT-23 dataset. This dataset captures real-world IoT traffic, which makes our study highly relevant to practical applications. Second, this paper proposes a novel hybrid model that combines a DNN for feature extraction and an RF classifier for binary classification of benign and malicious activities.

Third, this paper addresses the issue of dataset imbalance, which can significantly skew model performance in IDS applications. Our novel balancing technique generates additional benign data using normal distribution-based sampling and ensures that the model can effectively learn to distinguish between benign and malicious activities.

In recognition of the computational constraints of IoT devices, we developed lightweight GRU and LSTM architectures to achieve high detection accuracy while minimizing computational overhead.

Finally, this paper shows the substantial improvements in accuracy and false rates of the final AI models for Anomaly-based IDS.

The rest of the paper proceeds as follows: Literature Review discusses the related works. The proposed models are presented in the Methodology section. The evaluation results are presented in the Experiments Section, with comparison results. Finally, the Discussion section discusses the results and limitations and the Conclusion concludes the paper and offers ideas for future work.

II. Literature Review

Li et al. [18] used convolutional neural networks (CNNs) for intrusion classification, dividing the dataset into four segments based on feature correlations. They transformed one-dimensional features into grayscale graphs and trained four CNNs (CNN1, CNN2, CNN3, CNN4) individually for binary classification. A combined model, CNN0, integrated these outputs and trained on the entire dataset. Using the NSL-KDD dataset, CNN1 achieved 82.62% and 67.22% accuracy on KDDTest+ and KDDTest-21, respectively. The ensemble model achieved 86.95% and 76.67% on the same test sets. Martin et al. [19] proposed a model that uses a linear classifier based on a Neural Network (NN) with linear activations. This model incorporates feature transformations using kernel approximation algorithms such as Nystrom, Random Fourier Features, and Fastfood transformation, which add the necessary complexity and non-linear characteristics to the model. To test their model, they chose three datasets but only performed binary classification on the NSL-KDD dataset. The highest accuracy achieved in this binary classification was 80%. As evident, the results were not particularly promising. Kim et al. [20] developed a hybrid model integrating a convolutional neural network (CNN) and a long short-term memory network (LSTM) for the binary classification. They tested this model on two publicly available datasets, CSIC-2010 and CICIDS2017, achieving accuracies of 91.5% and 93.0%, respectively. Susilo et al. [21] utilized three algorithms—Random Forests, Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN)—to identify network intrusions. They employed the Bot-Iot dataset, created by UNSW Canberra [22] for multi-class classification. The CNN algorithm achieved the highest accuracy, reaching 91.25%. Yin et al. [23] introduced a deep learning method for intrusion detection using recurrent neural networks (RNN-IDS) and assessed its performance in both binary and multiclass classification tasks. They trained their model using the KDDTrain+ dataset and tested it with the KDDTest+ and KDDTest-21 datasets, the latter being a subset of the former. By experimenting with different hyperparameters (such as the number of nodes and learning rate), they achieved the highest accuracy of 83.28% on the KDDTest+ dataset and 68.55% on the KDDTest-21 dataset, with the optimal configuration being 80 hidden nodes and a learning rate of 0.1. Sokolov et al. [24] explored the use of Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), for intrusion detection for the binary classification in Industrial Control Systems (ICS), due to their ability to handle sequential data. The study emphasized the importance of considering both network traffic and the state of industrial processes for effective intrusion detection. The experiments compared the performance of LSTM and GRU networks in detecting intrusions using the Gas Pipeline dataset. GRU networks showed slightly better performance with an accuracy of 91.70% compared to LSTM's 90.68%. The study found that GRU networks learn faster and are computationally more efficient than LSTMs. None of the papers didn't address dataset balancing or adequately reported false rates. When one class significantly outnumbers another, models tend to become biased toward the majority class, leading to poor performance in detecting the minority class (anomalies). Thus, balancing the dataset helps the model generalize better to new, unseen data. High accuracy on an imbalanced dataset can be misleading. For example, if 95% of the data is normal and 5% is anomalous, a model predicting every instance as normal would achieve 95% accuracy but fail to detect intrusions. Balancing the dataset ensures that accuracy and other metrics truly reflect the model's performance and lead to better training dynamics, as gradient-based learning algorithms benefit from more stable gradients. Therefore, we balanced our dataset and included metrics such as False Positive and False Negative rates for comprehensive evaluation. Balancing false negatives and false positives is often the goal, but the operational context also influences prioritization. In stable environments, reducing false positives is crucial for operational efficiency. In dynamic environments, a low false negative rate is preferred to adapt to emerging threats. In high-security settings, minimizing false negatives is paramount, even at the cost of more false positives, to avoid missed detections leading to successful attacks. Hence, our primary aim is to minimize false negatives while maintaining an acceptable false positive rate.

III. Methodology

A. DNNs

DNNs (Deep Neural Networks) are suitable for anomaly-based intrusion detection systems because they can learn complex patterns and representations from large volumes of network traffic data, enabling them to identify subtle deviations from normal behavior that may indicate potential threats. Their ability to capture intricate features and relationships helps to improve the detection of previously unseen or sophisticated anomalies. Utilizing the Rectified Linear Unit (ReLU) activation function across all hidden layers, our Deep Neural Networks (DNN) model is optimized using the "adam" optimizer, a widely used and optimized gradient descent algorithm, with the "binary_crossentropy" serving as the loss function.

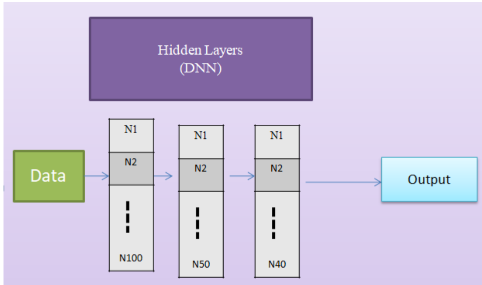

Figure 4. Arch. of DNN

Table 1. Details of DNN

Layer(Type)	# Nodes	# Parameters
Dense	100	1500
Dense	50	1500
Dense	40	1500
Dense	1	41

B. LSTM

LSTM [25] networks are designed with memory cells that can retain information over extended periods, allowing them to capture long-term dependencies in data. This is particularly important in IoT environments, where the sequence of events leading to an intrusion may span multiple time steps. LSTM's ability to maintain and update memory selectively enables it to detect complex patterns in network traffic, improving the model's capability to identify subtle anomalies. We developed a very lightweight sequential LSTM model comprising just three hidden (LSTM) layers with limited nodes. In the final hidden layer, we allocated only five nodes and excluded sequence return, opting instead for the dense layer as the output layer (Output at Final Time Step).

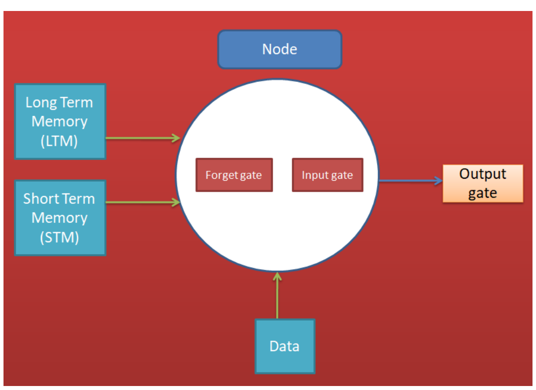

Figure 5. Arch. of LSTM; Forget Gate: Decides what information to forget from the cell state; Input Gate: Updates cell state with new information; Output Gate: Outputs final result after updating the cell state. Data is processed in each gate

Table 2. Details of LSTM

Layer(Type)	# Nodes	# Parameters
LSTM	20	1760
LSTM	10	1240
Dense	5	320
Dense	1	6

C. GRU

GRU [26] is a more computationally efficient variant of LSTM, with a simpler architecture that often provides similar performance while reducing the computational overhead. GRUs are particularly advantageous in IoT scenarios, where devices often have limited processing power and memory.

By leveraging GRU's efficiency, we aimed to achieve high detection accuracy without imposing excessive computational demands, making it a practical choice for real-world IoT deployments. To improve outcomes, we adopted a Gated Recurrent Unit (GRU) model featuring three layers of hidden units. In the initial layer, we incorporated a one-dimensional Convolutional layer to compress the data, employing a filter count of five, a kernel size of two, a stride of two, and valid padding. For the second layer, we utilized just five nodes and ensured sequence retention. In the last hidden layer, we employed only three nodes without sequence retention.

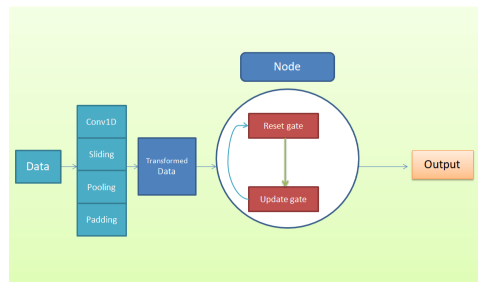

Figure 6. Arch. of GRU; Update Gate: Determines the balance between the old and new information; Reset Gate: Decides how much past information to forget; Output: weighted sum of the previous hidden state and the candidate hidden state, using the update gate; Data is processed in each gate

Table 3. Details of GRU

Layer(Type)	# Nodes	# Parameters
LSTM	20	1760
LSTM	10	1240
Dense	5	320
Dense	1	6

By training the autoencoder to reconstruct the input data, the model can identify anomalies based on the reconstruction error. Higher reconstruction errors often signify deviations from the learned normal patterns, making autoencoders a valuable tool for detecting previously unseen attacks or rare events in IoT networks. In our stacked auto-encoding setup, we utilized LSTM layers as the hidden layers due to their effectiveness in sequence learning. During the encoding phase, the model compressed the data from ten dimensions down to three (with three representing the latent representation). In the decoding phase, two hidden layers were employed to restore the dimensions from three back to ten.

Likewise, we employed the GRU layer for autoencoding purposes as well.

D. Bi-LSTM

A Bi-directional LSTM (Bi-LSTM) [27] consists of two separate LSTMs: one that processes the sequence forward

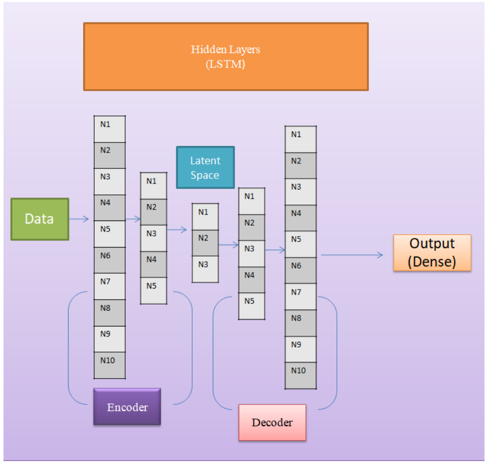

Figure 7. Arch. of AE(LSTM)

Table 4. Details of AE(LSTM)

Layer(Type)	# Nodes	# Parameters
LSTM	10	None
LSTM	5	None
Dense	3	None
LSTM	5	800
LSTM	10	966
Dense	1	None

(from the start to the end) and another that processes it backward (from the end to the start). Bi-LSTMs can leverage their bidirectional processing to recognize anomalies that might not be apparent when only considering past events like RNNs do. For instance, an unusual pattern might become clear when examining how it fits within the broader sequence of events, both before and after it. In the initial hidden layer, we utilized a basic RNN layer, while in the subsequent hidden layer, we employed an LSTM layer for bidirectional processing to address the previously mentioned challenges associated with RNNs.

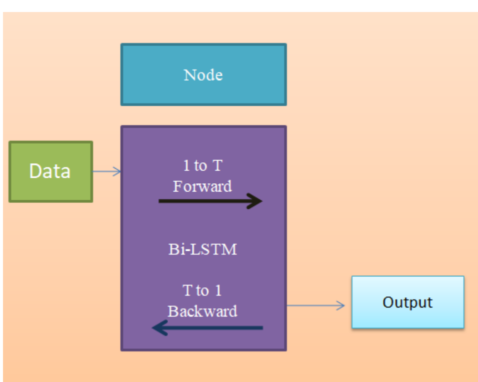

Figure 8. Arch. of Bi-LSTM; Forward (Hidden State): Processes the sequence from 1 to T; Backward (Hidden State): Processes the sequence from T to 1.

E. DNN-RF

We developed an innovative architecture where a Deep Neural Network (DNN) was used for feature extraction, and a Random Forest (RF) was employed for binary classification. The layered structure of DNNs allows them to capture hierarchical features in the data, ranging from low-level attributes to more abstract representations. This capability is crucial for intrusion detection in IoT networks, where raw data is often high-dimensional and heterogeneous. By using DNNs for feature extraction, we aimed to reduce the dimensionality of the data while preserving the most informative characteristics, thereby enhancing the performance of Random Forest classifiers.

Table 5. Details of Bi-LSTM

Layer(Type)	# Nodes	# Parameters
LSTM	20	None
LSTM	20	None
Dense	1	None

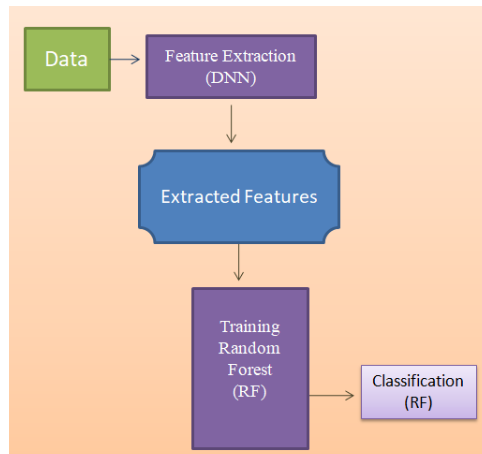

Figure 9. Arch. of DNN-RF

IV. Simulation Setting

A. Simulation Environments

In this study, we used Keras on the backend Tensorflow (Version: 2.10.0), known for its speed, simplicity, and popularity in deep learning. The experiment was conducted using Jupyter Notebook (Version: 6.4.12) on an HP Pavilion personal computer equipped with an Intel Core i7-1195G7 CPU @ 2.90GHz and 16 GB of RAM.

B. Data Set and Preprocessing

This study utilized the IoT-23 dataset, which comprises network traffic collected from Internet of Things (IoT) devices. It includes 20 instances of malware and 3 instances of benign activity. This dataset aims to provide a substantial collection of real-world labeled IoT malware infections and benign IoT traffic, facilitating the development of machine learning algorithms for researchers. It consists of 23 captures, referred to as scenarios, wherein 20 involve malicious activity and 3 involve benign activity. Each capture from infected devices may include the name of the potential malware sample executed in that scenario.

1) IoT-23 dataset: The IoT-23 dataset includes various labels representing threats such as Attack (exploits vulnerabilities), Command and Control (C&C), Distributed Denial of Service (DDoS), and botnets like Mirai and Okiru. Furthermore, Zeek functions as software for conducting network analysis. The IoT-23 dataset utilized it in the conn.log.labeled format, derived from the original pcap file through the Zeek network analyzer. Specific attributes of the features, such as conn_state and history, hold significant value with characters that carry particular implications. For a detailed explanation of these specific values and their meanings, readers are referred to [28].

We encountered a minimal number of missing values, which we addressed by substituting them with zeros. Features 'local_orig', 'local_resp' which were empty for all the files (scenarios) and hence they were dropped. Based on previous work on pre-processing of intrusion detection datasets like NSL-KDD, CICIDS2017, features id.orig_h, id.orig_p, id.resp_h and id.resp_p contained IP addresses and port numbers were dropped. Additionally, the 'history' attribute, representing a sequence detailing the connection history, was initially dropped.

2) Pre-Processing: Since our focus was on binary classification, we assigned a value of '1' for all attack instances and '0' for benign instances in the "label" column using Python's 'map' function. Following the encoding of object type features, we examined each column and observed that the majority of values in certain features were zero. Consequently, we decided to drop those features from consideration as well. After removing all those columns, we now have a total of 15 columns, including the label column.

To balance our dataset, we developed a novel approach. Initially, we divided the dataset into two segments: training set (75%) and test set (25%). Subsequently, we constructed a sequential deep neural network (DNN) comprising four layers, culminating in an output layer with a single node for binary classification. Training the network on the training set yielded an accuracy of 89.3%.

Our objective now is to generate some randomly distributed normal data. We first constructed a dataset comprising only benignnormal flows, identified by a label column value of '0'. Next, we calculated the standard deviations of all 14 columns using the 'std()' function. We then scaled down each standard deviation by a factor of 0.1 to reduce and normalize the deviation values, converting them into a 'numpy' array. A function named 'random_val()' was created to generate random values using a normal distribution via the 'tf.random.normal' function, with parameters including a seed value range of 1 to 256, a shape of 148534 rows and 14 columns, a mean of 0, and standard deviations derived from each column of the normal data. Finally, we employed our trained model to predict benign data from the generated random values. By iterating through a loop, we obtained the desired quantity of benign data and appended it to the original benign data. Following this procedure, we eliminated the duplicated entries and ultimately obtained a reasonably balanced dataset, as depicted in Figure 5(Right).

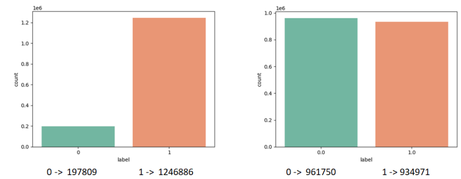

Figure 10. Dataset Balancing

C. Training and Validation Setting

The entire dataset was divided into a training portion comprising 75% and a testing portion making up 25% for all the models mentioned above. To implement this, we first divided our dataset into two parts: 75% for training and 25% for testing. The training set was further split into 60% for actual training and 40% for validation. The DNN was trained using the 60% training data and validated with the remaining 40%. The DNN outputted extracted features as ten-dimensional vectors. These features, along with their corresponding labels, were then used to train the RF classifier. Finally, the classifier was tested on the 25% testing data, leading to superior results.

V. Results

To comprehensively assess the models, the following tables provide details on the Accuracy, Precision, Recall, F1-score, False Positive (F/P), and False Negative (F/N).

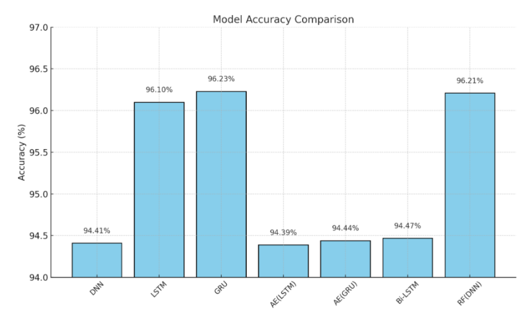

Figure 11. Accuracy

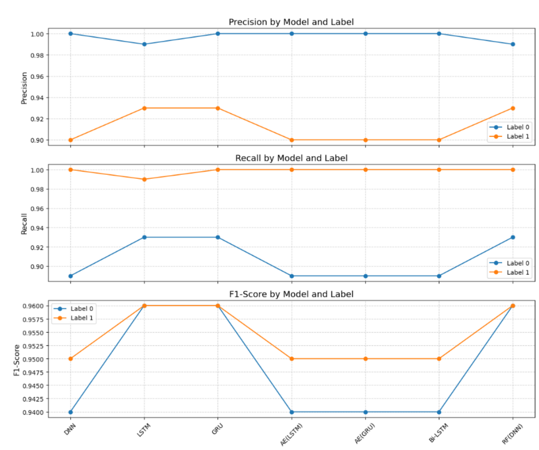

Figure 12. Precision, Recall and F1

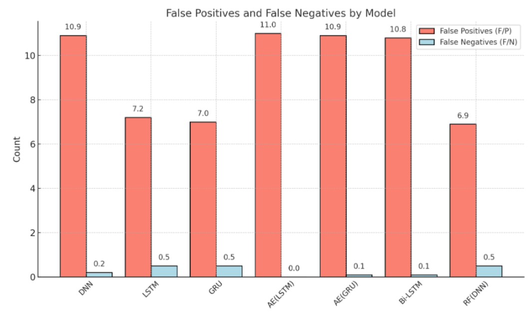

Figure 13. F/P and F/N

We achieved an exceptional improvement in F/N, reducing the rate to 0.0%. Additionally, we observed a notable enhancement in F/P, achieving a rate of 6.9%, which compares favorably to other studies, such as [18], which reported an F/P rate of 13.5%.

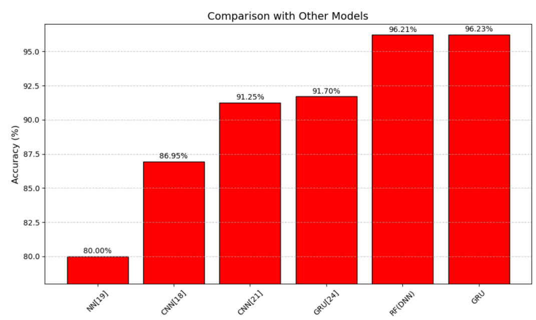

Figure 14. Comparison with Others

While previous studies [15], [16], [18], and [21] implemented Neural Networks (NN), CNN, and GRU models, achieving accuracies of 80.00%, 86.95%, 91.25%, and 91.70% respectively, our simulation results demonstrate a significant improvement, attaining accuracies of 96.21% and 96.23%, as illustrated in Figure 14. This highlights the effectiveness of our proposed methodology in outperforming existing approaches.

VI. Discussion

Our analysis shows that GRU and LSTM outperform other models, with accuracy rates exceeding 96%. This detailed comparison fills a gap in the literature, where fewer studies have investigated the performance of different deep-learning models for anomaly detection in IoT environments. By leveraging the feature extraction capabilities of DNNs and the classification strength of Random Forests, our hybrid model achieves 96.21% accuracy, outperforming standalone models and reducing false positive and false negative rates. This hybrid approach contributes to more reliable and accurate intrusion detection in IoT networks, a domain where balancing accuracy and computational efficiency is critical. Our models, especially GRU and LSTM, demonstrate false positive rates as low as 7.0% and false negative rates approaching 0%. This substantial improvement in performance metrics ensures that our IDS solutions can detect a wide range of cyber threats without overwhelming the system with false alarms or missing critical intrusions. These contributions are significant in both academic and practical contexts. By improving the accuracy, efficiency, and reliability of IDS systems for IoT networks, our work provides a foundation for future research in the field and addresses key challenges posed by the rapid growth of IoT devices. With billions of IoT devices expected to be deployed globally, the ability to detect and mitigate cyber threats in real time will become increasingly critical. Our models offer practical solutions that can be integrated into existing security frameworks, contributing to more secure IoT networks.

VII. Conclusion

By presenting novel architectures, balancing techniques, and thorough performance comparisons, this paper highlights significant advancements in AI-driven intrusion detection systems. The combination of high accuracy, low false positive/negative rates, and resource-efficient models makes our contributions highly relevant for real-time IoT security applications. Our work not only surpasses existing methodologies but also opens new pathways for future research, making it a critical addition to the field of cybersecurity.

References

[1] C. Vorakulpipat, E. Rattanalerdnusorn, P. Thaenkaew, and H. Dang Hai, "Recent challenges, trends, and concerns related to iot security: An evolutionary study," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 2018, pp. 405–410.

[2] S. Fenanir, F. Semchedine, S. Harous, and A. Baadache, "A semi-supervised deep auto-encoder based intrusion detection for iot." *Ingenierie des Systemes d'Information*, vol. 25, no. 5, 2020.

[3] M. Burhan, R. A. Rehman, B. Khan, and B.-S. Kim, "Iot elements, layered architectures and security issues: A comprehensive survey," *sensors*, vol. 18, no. 9, p. 2796, 2018.

[4] K. Angrishi, "Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets," *arXiv preprint arXiv:1702.03681*, 2017.

[5] R. Ahmad and I. Alsmadi, "Machine learning approaches to iot security: A systematic literature review," *Internet of Things*, vol. 14, p. 100365, 2021.

[6] S. Das, P. Amritha, and K. Praveen, "Detection and prevention of mirai attack," in *Soft Computing and Signal Processing: Proceedings of 3rd ICSCSP 2020, Volume 1*. Springer, 2021, pp. 79–88.

[7] B. Tushir, H. Sehgal, R. Nair, B. Dezfouli, and Y. Liu, "The impact of dos attacks onresource-constrained iot devices: A study on the mirai attack," *arXiv preprint arXiv:2104.09041*, 2021.

[8] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[9] Statista. (June 2024) Number of internet of things (iot) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2033. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[10] Cisco. (June 2024) Cisco annual internet report (2018–2023) white paper. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[11] S. Smith. (2020) Iot connections to reach 83 billion by 2024, driven by maturing industrial use cases. [Online]. Available: https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024/

[12] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the internet of things: A comprehensive investigation," *Computer Networks*, vol. 160, pp. 165–191, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128619306267

[13] K. Al Jallad, M. Aljnidi, and M. S. Desouki, "Anomaly detection optimization using big data and deep learning to reduce false-positive," *Journal of Big Data*, vol. 7, pp. 1–12, 2020.

[14] A. Bécue, I. Praça, and J. Gama, "Artificial intelligence, cyber-threats and industry 4.0: Challenges and opportunities," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3849–3886, 2021.

[15] K. Oosthoek and C. Doerr, "Cyber threat intelligence: A product without a process?" *International Journal of Intelligence and CounterIntelligence*, vol. 34, no. 2, pp. 300–315, 2021.
[16] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, p. 105124, 2020.
[17] S. Garcia, A. Parmisano, and M. J. E. Erquiaga, "Iot-23: A labeled dataset with malicious and benign iot network traffic (version 1.0.0)," *CyberSANE H2020 Project*, vol. 20, p. 15, Jan,2020.
[18] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, and L. Cui, "Robust detection for network intrusion of industrial iot based on multi-cnn fusion," *Measurement*, vol. 154, p. 107450, 2020.
[19] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Shallow neural network with kernel approximation for prediction problems in highly demanding data networks," *Expert Systems with Applications*, vol. 124, pp. 196–208, 2019.
[20] A. Kim, M. Park, and D. H. Lee, "Ai-ids: Application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70 245–70 261, 2020.
[21] B. Susilo and R. F. Sari, "Intrusion detection in iot networks using deep learning algorithm," *Information*, vol. 11, no. 5, p. 279, 2020.
[22] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.
[23] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, vol. 5, pp. 21 954–21 961, 2017.
[24] A. N. Sokolov, S. K. Alabugin, and I. A. Pyatnitsky, "Traffic modeling by recurrent neural networks for intrusion detection in industrial control systems," in *2019 international conference on industrial engineering, applications and manufacturing (ICIEAM)*. IEEE, 2019, pp. 1–5.
[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
[27] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
[28] (Nov 2024) Base/protocols/conn/main.zeek-book of zeek (v4.0.1). [Online]. Available: https://docs.zeek.org/en/lts/scripts/base/protocols/conn/main.zeek.html