An Efficient Hybrid Method for Coordinated Attacks Detection in NAN of Smart Grid

Vijayanand Radhakrishnan
Department of Computer Science and Engineering
Jain University, India
rkvijayanand@gmail.com

Ulaganathan Meenakshisundaram
Department of Electrical and Electronics Engineering
PSR Engineering College, India
ulagu.er@gmail.com

Ganeshkumar Pugalendhi
Department of Computer Science and Engineering
Anna University, India
ganesh23508@gmail.com

John Basha
Department of Computer Science and Engineering
Jain University, India
jjbasha@gmail.com

Abstract: *Neighborhood area network is a robust communication model essential for disturbance-free power distribution. The reliability of the network depends on the flow-based attack detection model but the lengthy flow completion introduces high latency. This potential delay buys the time for attackers to study and pose subsequent attacks. Thus, the packet-based analysis is utilized in this work to detect the attacks at early stages. The proposed Intrusion Detection System (IDS) is designed with deep learning based Bidirectional Long Short-Term Memory algorithm and attention mechanism. IDS, with multiheaded attention, is executed in the substation to analyze the consolidated collected data of the traffic and detect coordinated attacks in the network. The developed model works effectively in earlier attack detection and the secondary level is used only on requirement. The proposed IDS is evaluated with standard datasets like 5G_NIDS, CICIDS2017 and UNSW-LD. The results proved the efficiency of the proposed method in of Neighborhood Area Network (NAN) coordinated attacks detection of smart grid communication.*

Keywords: *Packet based attack detection, Bi-LSTM, multi-head attention, NAN communication security, NS3 simulation.*

Received February 20, 2025; accepted September 1, 2025
https://doi.org/10.34028/iajit/23/1/11

1. Introduction

The real triumph of the Smart Grid (SG) is providing interruption-free power distribution to the customers. SG is a highly sophisticated electric grid constructed with advancement from various fields. It utilizes various communication systems to collect real-time data from end users. This data is highly sensitive and should reach the substation on time with integrity. The Neighborhood Area Network (NAN) is the most supportive component for this reliable data communication. NAN is derived from the concepts of Local Area Network (LAN) and Metropolitan Area Network (MAN), typically covering the range of 1 kilometer, to satisfy the growing need of connecting various devices for specific purpose in close proximity. It is capable of accommodating multiple intelligent electronic devices, sensors, and meters and also facilitates communication between these devices. The notable features of NAN like rapid speed, reliability, and centralized management help to achieve the availability, integrity, and confidentiality of the transmission. It has the drawback of high latency in data reaching the substation due to the huge number of devices [9].

The performance of NAN depends on the security model adopted for the system. Modern-day attacks are carried out in a highly coordinated manner, challenging to counter even with the most advanced systems.

Intrusion Detection System (IDS) has been a promising technology for more than two decades that detects the attack before causing severe damage to the network [14]. It is placed at various components of the network like routers, gateways, switches, entry and exit points. It continuously monitors the traffic data passively and alerts the user if any attack data is detected. However, the use of data encryption algorithms and the high volume of traffic have increased the complexity of attack detection [2]. Thus, the flow-based analysis at endpoints looks promising solution and is employed widely for this purpose. A network flow represents the entire transfer of a specific message, including details such as starting time, destination time, addresses, hop count, and other relevant information [26, 29]. IDS analyses these collected network parameters for attack detection. It faces np-hard complexity in the finding of attacks from flow data due to incomplete information, data loss, and zero-day attacks. Machine learning algorithm is a promising solution employed to overcome these problems. Recently, deep learning models have dominated this field that output the maximum accuracy by deeply excavating the network packets [20].

These advanced models have waited for a prolonged period to detect the attacks from flow-based analysis due to the occurrence of delay in complete transmission

of data [10]. For example, if the last packet in a sequence fails to reach the receiver, the IDS must wait until the flow terminates or the timeout period is reached. This delay is unnecessary, as the IDS can detect most attacks without needing the complete packet data. Indirectly, it gives additional time to the attackers to analyse the network, potentially enabling them to launch more advanced threats in the future. It further worsens in the case of coordinated attacks where the control centre is difficult to collect multiple network data. This issue significantly impacts the efficiency of IDS and needs to be addressed urgently in highly sensitive systems such as smart grids. Otherwise, it causes severe damage to the life and cost of people. Thus, the authors have come up with the suggestion of detecting attacks at early stages with the help of network packets instead of flow. This recommendation requires the most advanced prediction system for detecting the attacks with few packets. This adds to the complexity of designing IDS and opens new avenues for exploring solutions in this direction.

To address this, the proposed method is designed in this work that utilizes the advantage of transformer-based attention mechanisms and Bidirectional Long Short-Term Memory (Bi-LSTM) neural network. This paper explores the impact of developed method on the early detection of coordinated attacks in smart grid. The key parts of the study are outlined as follows:

1. A novel intrusion detection method by integrating Bi-LSTM with an attention mechanism for coordinated attacks in NAN of smart grid is developed.
2. The efficiency of the proposed method is validated with standard datasets such as CICIDS2017, UNSW-LD and 5G-NIDD.
3. A comprehensive performance comparison is conducted against existing standards and research methods on various metrics, with the detailed discussion of results.

The rest of the paper is structured as follows: Section 2 contains the reviews related to the detection of coordinated attacks and its limitation. Section 3 explains the designing and working of proposed method. Section 4 provides the information about dataset preparation and the steps to execute the proposed model. The results collected from the developed model and the effectiveness by comparison with existing methods are analyzed in section 5. Finally, the paper is outlined and concluded in section 6.

2. Related Work

The power distribution of smart grid depends on the continuous availability of the network between meters and substation. It is difficult to set up uninterrupted services due to wide geographical locations, a vast number of devices incorporated, numerous sensors usage, etc., The incorporation of radio networks into traditional electric grid increases the risk level to the maximum range [23]. The zero-day attacks pose a serious threat to the designed security structures in these highly sensitive networks. It needs to adopt advanced security strategies for its proper functioning. Intrusion detection system is a trustworthy model that detects the threads before causing any serious damage to the network. The advent of machine learning algorithms raises the value of IDS by predicting the attacks accurately. It detects the attacks by matching signatures with regular expressions of attack patterns [19].

Kasongo and Sen [17], an IDS model is designed using XGB algorithm to detect modern-day attacks such as DOS, generic, exploits, etc., It was assessed with the performance of a logistic regression, decision tree, Support Vector Machine (SVM) and perceptron models. The filter-based feature dropping mechanism is employed with that algorithm to improve the efficiency at a certain level. It selects 19 best features to boost detection and rapid execution. The accuracy of the classifier is further improved by hybridizing with one or more optimization methods. The Mutual Information (MI) technique is merged with Genetic Algorithm (GA) to select the most informative features [30]. In that work, the MI algorithm acts on the semi-informative features of GA-selected features to find the best parameter. These concluding features uplift the output of SVM algorithm in wireless mesh network. The performance of the same machine learning algorithms varies across different applications, making it challenging to tailor each algorithm to specific purposes. Thus, the multiple algorithms are ensembled to strengthen the detection and make it suitable for multiple applications. Li *et al*. [21] designed an IDS for providing security to airborne environments. In this method, a tree based multi-layer ensemble model is integrated with the supervised algorithm to detect the attacks. Bayesian optimization tree-structure parzan estimator is used as a hyperparameter to upgrade the classifier performance in attack recognition. Various methods are employed by the researchers for the threat detection of wireless applications [5, 15, 24]. These models have performed well in small networks but suffer to predict the attacks in complex networks due to the poor convergence rate of the chosen algorithms [31]. Assistive techniques, such as optimization algorithms, can enhance performance to some extent but often fail to deliver fully satisfactory results.

Recently, Deep learning algorithms has posed as a promising technology skilfully detect the advanced attacks, including zero-day attacks. It has the powerful mathematical models that solves the feature extraction problems and predict the output with lesser number of features. Kardi *et al*. [16], LSTM based neural network is employed to detect the anomalies in electricity consumption data. It is implemented in two steps where the first LSTM predicts the next hour consumption data which is used as input to the second LSTM integrated with autoencoder mechanisms. Moreover, the detection

of attacks in the time-series environment is one of the most difficult tasks. Guha *et al*. [12] proposed a hybrid model where Bi-LSTM is embedded with autoencoder to analyse time-series power grid network data. The drawback of using Bi-LSTM model is the need of more computational power in the processing of complex datasets. The authors utilized the optimization techniques to mitigate this problem. They have metrics such as precision, accuracy and recall to inspect the developed method.

Modern day attacks are well-planned and are engaged from multiple devices in a distributed fashion. The traditional models have performed well at specific levels but require advanced models to mitigate it. Abid *et al*., proposed a distributed IDS with gradient-boosted trees to examine the data of a cloud environment [1]. They demonstrated the model based on the reasonable response time to represent the effectiveness of the system. Apart from machine learning algorithms, other advanced concepts are also utilized to secure the network. Yakubu *et al*., designed a security model against the colluding attack with the help of blockchain technology ethereum [32]. They employed a single server queuing system and an authentication mechanism to mitigate the attack. They conduct tests on smart contract parameters such as timestamp dependency, assertion failure, etc., to identify the bugs related to the threads. They investigated the efficiency of their developed model on metrics such as message processing time, response time, complexity cost, and accuracy. Anley *et al*. [4] analysed many literatures and concluded that the non-inclusion of distribution environments is a limitation in the existing models. They developed a solution using convolutional neural network and adaptive transfer learning to overcome this issue and evaluate their model performance with combined datasets collected from multiple fields.

The performance of deep learning algorithms might be enhanced with the help of optimization algorithms. Alrayes *et al*. [3] presents a distributed Bidirectional Gated Recurrent Unit (GRU) model improved by the golden jackal method to secure internet-of-things communication. The Chaotic crow search optimization is additionally applied with the developed model to boost the detection ratio. This combination has produced better results compared to Bi-GRU alone. This kind of implementation has some limitations also such as maximizing computational time and complexity. Alternatively, deep learning algorithms are designed in such a way that one algorithm is responsible for extracting optimal features, while another handles the detection part. Diaba and Elmusrati [8], an IDS is proposed to integrate the CNN and GRU models in a smart grid. The CNN layer helps to extract the input features by capturing position-invariant characteristics. On the other hand, The GRU model uses the memory cells to extract the informative features from the previously collected features. The model is constructed with four CNN and three GRU blocks. The concatenation layer is used to combine the outputs of both CNN and GRU layers to predict the categories of the attack. Peng *et al*., use the hybrid model with CNN and RNN algorithms [25]. In that model, CNN algorithm helps to correlate the relation between the network features and RNN is used to mine temporal and spatial features from the traffic matrix that helps to find the intrusions.

All existing Network Intrusion Detection System (NIDS) models detect attacks based on data collected from network traffic flows. It increases the time delay in the detection process. For instance, the standard timeout period of any communication is considered as 60 seconds [6]. Few of the network flows take a long time to complete which slows down the analysis of later traffic data. This time delay is a serious problem in highly sensitive environments like smart grid. In the case of coordinated attacks, intruders capture multiple nodes and launch attacks simultaneously. The network delay provides them the opportunity to analyze and understand the workings of designed security models. Thus, the attacks should be detected at early stage to reduce the maximum damage to the network. Djaidja *et al*. [10] provide a method to handle it. They suggest a framework that explores the network based on the packet analysis model. It is not an easy task, as even machine learning algorithms struggle to predict the outcome. Therefore, the author employed deep learning GRU algorithms with an attention mechanism to forecast the result. The attention mechanism helps to provide additional contextual information by selectively focusing on important points. They simulate the packet analysis model with standard datasets using ScaPy library and demonstrate the integrated model using python language. Their experiments on isolated systems yielded promising results, accurately detecting the majority of attacks by evaluating the first few packets within 5 seconds. However, this approach appears to be a promising method for combating recent attacks. In this paper, the author examines the earlier detection of coordinated attacks using Bi-LSTM and attention mechanism by analyzing the packet headers of network flow analysis. They applied the method to simulate the normal and coordinated attack data generation for evaluating the proposed method.

3. Methodology Overview

Figure 1 shows the working model of proposed algorithm. This work starts from the simulation of packet flow derived from the standard flow analysis dataset. The generated packets exhibit a sequential pattern similar to real data. In this work, the Bi-LSTM is employed with multi-head Attention to analyze the packets in the classification task. The deep learning model has the advantage of processing the packets effectively in the forward and backward directions.

Whereas, the attention mechanism has the capability of tracking selective information of packets received from multiple nodes. This combination helps the early detection of attacks effectively.

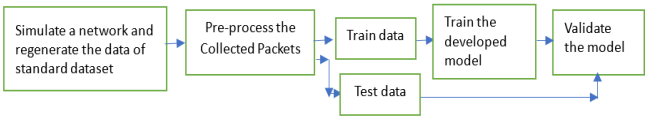

Figure 1. Work flow diagram of the proposed model.

3.1. Data Collection and Processing

In this work, the simulation is designed based on the information given by Djaidja *et al*. [10]. The experiment simulates the standard dataset such as ADFA-LD, CICIDS2017, and KDD Cup 99 using NS3 packet trace in the form of Packet Capture Data (PCAP) files. It generates the normal and attack data based on the metadata information provided by standard datasets including time, types, and starting time of attack. It contains information about individual packets including the headers and payload. Each packet is identified with flow ID, source address, destination address, source port, destination port, etc. The packet is converted into argus file format initially and is then transformed into CSV style for processing. These datasets are used to train the developed classifier in a supervised fashion. The features extracted from the data are:

- Flow id: each flow has a unique number. It represents the particular flow of the packet.
- Packet arrival time: it represents the arrival time of each packet in a flow. It helps to find how long the flow is required to complete. It is measured in seconds because of NS3 simulator has that feature to measure the timing.
- Packet types: in TCP communication, the packet have both data and control packets. The data packets contain the information and the control packets has the commands useful to the network devices.
- Packet payload length: it gives the size of each packet.
- Source port: it shows the port address of starting node.
- Destination port: it indicates the purpose of the packet with the port address of the destination node.
- Protocol: TCP and IP protocols such as HTTP, ICMP, etc., are used in this work.
- Inter arrival time: this is not a field in the packet. It measures the time difference between the current and preceding packets.
- Time to Live: this field is available in each packet to indicate the validity of the packet. If the packet is not received before that time, it will be discarded.
- TCP flags: 8 flags are available in TCP/ IP packets. Each bit occupies 1 bit in the packet but contains valid information about the packet.
- Sequence number: it shows the position of the packet in the flow.

3.2. Deep Learning Concept

3.2.1. Bi-LSTM Algorithm

LSTM is a kind of Recurrent Neural Network (RNN) designed to extend the memory capabilities of previous models, enabling them to effectively learn from long-term input sequences. It employed input (T_t), forget (f_t), and output (O_t) gates for improved performance. The forget gate values are mathematically calculated as in Equation (1):

$$f_t = \sigma(w_f.[h_{t-1}, T_t] + b_f) \quad (1)$$

The Bi-LSTM is a neural network that analyze the systems using two separate LSTM. The first LSTMs analyzes the forward flow and the second is employed for reverse flow. It incorporates two hidden layers to analyze the bidirectional data as represented in Figure 2. Each layer has separate function as in Equations (2) and (3), used to upgrade the performance at each iteration.

$$F_t = Func_{forward}\,(T_t, F_{t-1}) \quad (2)$$

$$B_t = Func_{backward}\,(T_t, B_{t+1}) \quad (3)$$

Both LSTM provide their result to the output layer, it processes further to conclude the decision using Equation (4).

$$O_t = w_O F_t + w_O B_t + BS_O \quad (4)$$

Where O_t-output value at time t, w_O-weight matrix and BS_O-bias value. Bi-LSTM has performed well in the network but has limitations in the selection of parameters like learning rate, total number of layers and training iterations [18]. It affects the convergence rate of the classifier in large and small datasets. In large datasets, it tends to suffer from overfitting, while in smaller datasets, it struggles to converge. Similarly, the use of a large learning rate leads to overshoot of the results and the small value causes slow convergence. Thus, the enhancement models are employed to assist Bi-LSTM to get the expected performance.

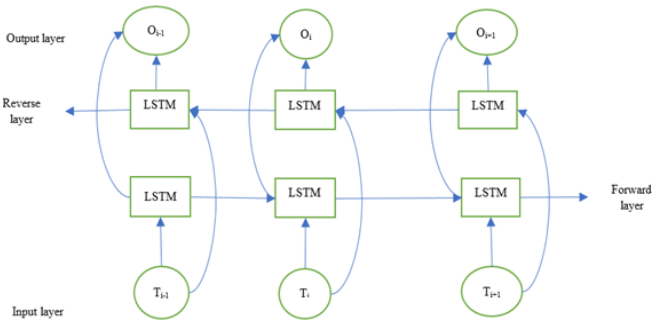

Figure 2. Bi-LSTM architecture [13].

3.2.2. Attention Mechanism

The attention mechanism mimics human vision by dynamically assigning varying weights to different regions of the input data, helping to predict the output more effectively. Multihead Attention (MHA) model is

an improved version that uses multiple headers for processing the data. Each header has an attention function that views the data from different perspectives. In other words, the input space is split into multiple subspaces then the headers are focused on each group and finally, the decision through parallel processing. Three components are required for the implementation of attention module: Query (Q), Key (K) and Value (V) [22]. Query chooses the position of interest in the input data to be processed which is converted into matrix representation. It may be single or multiple features in CSV dataset or particular region of an image. The key is a matrix that is compared with multiple queries to identify the informative features in the input data. The value contains detailed information about the output which is used to predict the remaining data effectively. The attention mechanism uses the scaled dot product to get the reconstructed attention weights and matrix which is given in Equation (5) [11].

$$Head\ (Q,K,V) = softmax\ (\frac{QK^T}{\sqrt{d_k}})V \quad (5)$$

Where d_k is key dimensionality. This (Q, K, V) is the output of single head, similarly each head (h) output is found and is concatenated to get the final output as given in Equation (6).

$$Multihead\ (Q,K,V) = Concat\ (h_1, h_2 \dots \dots h_n)w^0 \quad (6)$$

Where W^0 is weight matrix of linear output function. The final output effectively focusses on the informative areas and can construct the output with a smaller number of input samples.

3.3. Proposed Method

This subsection contains the information about the proposed system which is represented into three sub-modules such as projection layer, encoder and decoder layer. The block diagram of the proposed Bi-LSTM with attention model is shown in Figure 3.

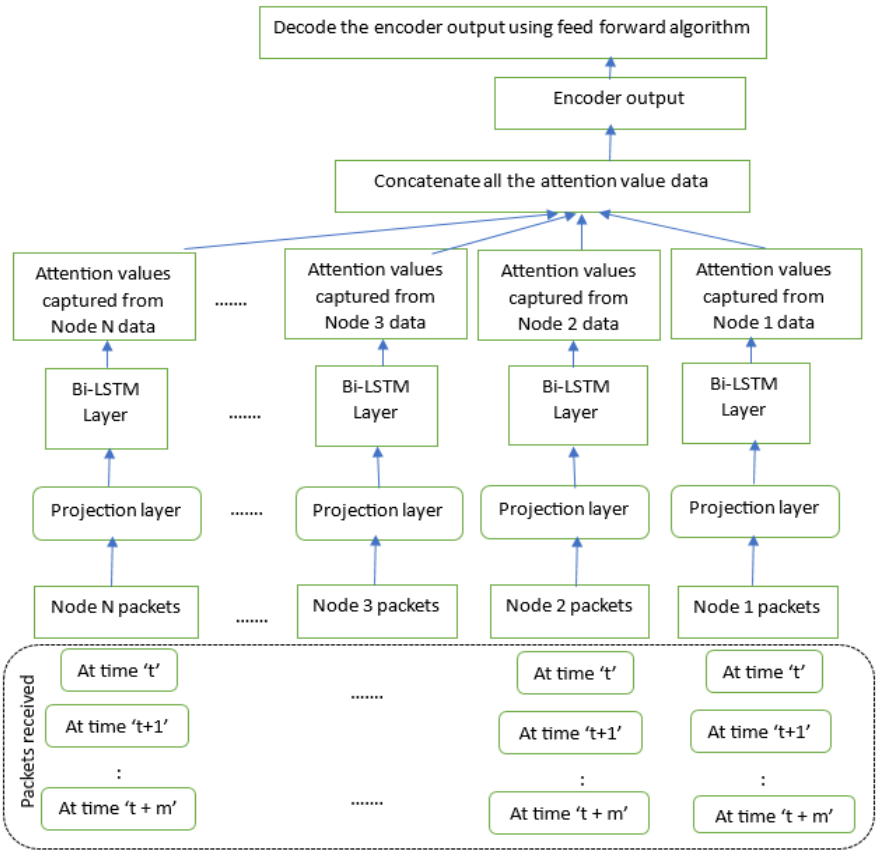

Figure 3. Block diagram of the proposed method.

3.3.1. Projection Layer

In this work, the packets received from multiple nodes are received simultaneously. The packets received in the simulation are not in order and our model processes each packet to find the coordinated attacks in the combined dataset. The raw packet data is not directly applied to the proposed algorithm. Thus, the projection layer is employed to modify the data 'I' in the required format as in Djaidja *et al.* [10]. The objective of this layer is to map input data into higher dimensions, which helps to expand relationships and simplify the complexity of analysis. The features extracted from the packet are fed as input into the feedforward layer. This layer is then followed by the dropout function D, which applies a specific dropout probability to randomly set a portion of the input tensor elements to zero. The output is calculated using Equation (7).

$$D' = D(W^T * I + B) \quad (7)$$

Where W is weight matrix and B is bias matrix.

3.3.2. Encoder Layer

The encoder section contains the details about organizing Bi-LSTM and MHA mechanisms to represent the data into latent representation for detecting the coordinated attacks in smart grid. In this work, the attacks are identified through the cumulative analysis of data from all meters. The *D'* data from projection layer is initially processed by the Bi-LSTM layer. It is organized into two hidden layers i.e., one is the forward layer and another is a backward layer. It is calculated using Equations (8) and (9).

$$Forward\ layer, \vec{h}_t = f_{fwd}(T, \vec{h}_{t-1}) \tag{8}$$

$$Backward\ layer, \overleftarrow{h}_t = f_{bwd}(T, \overleftarrow{h}_{t-1}) \tag{9}$$

The number of time sequences considered in the deep learning architecture corresponds to the maximum number of packets planned for message splitting. The advantage of this model is that it can make predictions at early stages while continuing the process until the flow is completed at the backend. The first packet is fed into the first pair LSTM and is moved forward toward the next sequence over time '*t*'. The final calculation from that algorithm is calculated using Equation (10).

$$O_t = w_O \vec{h}_t + w_O \overleftarrow{h}_t + b_O \tag{10}$$

Where w_O-Weight function of O and b_O-bias value. Bi-LSTM has the limitation of processing at slower speed due to low convergence rate and sometimes lead to overfitting. To overcome these drawbacks, it needs to be tuned with advanced methods. Thus, the hidden sequences obtained from these layers are provided as input to the MHA layer, as illustrated in Figure 3. In MHA, multiple heads are employed in the attention mechanism. Typically, each head focuses on a specific category of the input data. In this case, each head is assigned a specific node packet, and the values are updated accordingly. This permits the algorithm to retain the informative features from the outputs of the Bi-LSTM algorithm.

3.3.3. Decoder Layer

In this layer, the data from the MHA layer is processed with feed-forward layer to find the attack as well as its category. It is written as in Equation (11),

$$O = W^T * Encoder\ Output + B_O \tag{11}$$

Where W^T is the weight matrix of encoder size and B_O is bias value. The output of the decoder is collected in the form of a matrix of size *K*, where *K* is the number of attacks used for training. Each row represents the weight of a particular class, and the row corresponding to the attack data with maximum value is selected as the output of classifier.

4. Experimental Results and Analysis

In this section, the proposed system is experimented within a device having 'I5' processor of speed 2.00 GHz and 16 GB RAM with a graphic adapter of NVIDIA Geforce RTX 3060 4 GB. It is implemented by simulating the data in NS3 simulator as a coordinated manner. The collected data is then processed using python codes with advanced libraries. The result is examined on the basis of attack detection of the classifier and early detection capabilities.

4.1. Datasets

The experimentation of the developed method on real network is difficult to implement and is expensive. Thus, the standard datasets are utilized to evaluate the developed model. It has the advantage of processed data with already predicted labels. The standard datasets such as ADFA-LD [7], CICIDS2017 [28], and 5G-NIDD [27] are used in this work. The data related to DoS attacks is given higher priority, and only attacks of this type are considered. The simulation replicates the network to generate labeled flow sequences for model input.

Creech *et al*. [7] generated the ADFA dataset with the idea of evaluating system calls in the University of New South Whales, Australia. In this work, they generated the data in a Linux environment, ADFA-LD was chosen for simulation. The dataset comprises 44 features under normal and 9 attacks classes. The flow category related to smart grid environment is considered for assessing this research are Normal, Exploits, Denial-Of-Service (DOS), Reconnaissance, Generic and worms.

The University of New Brunswick, Canada has generated a CICIDS2017 dataset for evaluating the designed algorithms for network applications [28]. It captures the characteristics of 25 protocols like HTTP, HTTPS, SSH, FTP, etc., to ensure the reality of a real environment. The CIC-Flow meter captures the PCAP file of simulated network flow and is coded under 77 features. The attacks highlighted for evaluation of developed model are DOS, web attack, SSH_Parator, FTP_Parator and Benign.

5G-NIDD dataset is a recently generated dataset from an actual 5G network by researchers of Oulu University, Finland [27]. It is collected by the tracing devices fixed in two-base station in same time. The designed model focuses on two categories of attacks in their simulation as Denial-Of-Service (DOS) and port scan. The former category includes slow-rate DoS, ICMP flood, UDP flood, HTTP flood, and SYN flood. Similarly, UDP, SYN, and TCP connect scans are simulated under the port scanning category. In this research, DoS attacks only considered to evaluate the execution of the algorithm.

4.2. Simulation

The NS3 simulator is enrolled to simulate the smart meter data transmitted to the substation. In the simulator, the NS3 class "Node" is configured to

represent meter data. The substation node has more functions that evaluate the collected data. The AODV routing protocol is incorporated to establish the communication. The features like flow time, number of nodes, addresses, ports, etc., of each dataset are simulated based on the metadata information provided by the designers. The data are highly sensitive and all the attacks are simulated individually. The packets are captured and analyzed with the help of PacketSink and PacketSniffer classes provided by the NS3 developers. Tables 1, 2, and 3 present the number of packets captured from the simulated flow of the original dataset.

Table 1. Packet simulated from the flow data of ADFA- LD dataset.

Type	Number of Flows	No. of Packets
Normal	3300	16400
DOS	3200	36400
Fuzzers	1460	14570
Generic	2600	16850
Worms	50	800

Table 2. Packet simulated from the flow data of CICIDS 2017 dataset.

Type	Number of Flows	No. of Packets
Normal	15000	330000
DDOSLOIT	10027	280756
DoSHulk	7034	1062134
DoSGoldenEye	3567	49938
DoSSlowhttptest	2124	19116
FTp-Parator	1345	37660
DOSSlowIoris	798	8778
SSH-Parator	345	18975

Table 3. Packet simulated from the flow data of 5G-NIDD dataset.

Type	Number of Flows	No. of Packets
Benign	25692	205536
GoldenEye	2543	81376
Torshammer	3452	120820
SynFlood	2034	6102
Slowloris	943	136735

The collected labeled packet information is split in the ratio of 80:20 as training and testing data respectively. The training data is fed as input to the developed BI-LSTM and MHA-based model for training purposes. This trained model is subsequently validated using test data to assess its performance in attack detection. The evaluation is conducted by writing the scripts developed in the python language and open-source libraries. Although Python binding scripts for NS3 are available, they are not supported in low-resource simulation environments. Therefore, the simulation and detection are executed in separate environments to imitate the complete proposed system implementation in a real environment.

4.3. Performance Analysis

The learning rate used to optimize the classifier's performance is 0.02, batch size as 128, input layer as total number of features collected from the packet data and the total number of heads belonging to number of attacks. The output data is measured in parameters such as true negative, true positive, false negative and false positive. It can be analysed by calculating:

$$Accuracy=(TN+TP)/(TP+TN+FN+FP \quad (12)$$

$$Recall=TP/(FN+TP) \quad (13)$$

$$Precision=TP/(FP+TP) \quad (14)$$

$$F1\text{-}Score=(2*(recall*precision))/(recall*precision) \quad (15)$$

Tables 4, 5, and 6 show the results obtained from ADFA-LD, CICIDS2017 and 5G-NIDD datasets respectively. In Table 4, the precision values for fuzzers and worms highlight the challenges faced by the developed model in detecting positive data when trained on a limited dataset. This impacts the F1-Score, which serves as a balanced evaluation metric for both negative and positive data. It is foremost to note that while the accuracy for these attacks is high, the lower values of the other metrics suggest that these attacks are somewhat similar to normal data. This similarity increases the likelihood of these attacks being deployed in modified forms on a larger scale.

Table 4. Performance analysis of proposed method on ADFA-LD dataset.

Type	Number of flows	Accuracy	Recall	Precision	F1-score
Normal	3300	0.9712	0.977448	0.92268	0.94928
DOS	3200	0.99804	0.99968	0.99269	0.99617
Fuzzers	1460	0.95730	0.91538	0.78504	0.84522
Generic	2600	0.98685714	0.97603	0.96321	0.96958
Worms	50	0.99763265	0.8	0.73846	0.768

Similarly, in Table 5, the precision and F1-scores for certain attacks deviate from the detection performance of other data due to unbalanced training data. In this table, the DoSSlowIoris training data is lower in quantity but achieves better detection performance compared to some classes with larger datasets. It highlights the dissimilarity of attack class with normal data and the corresponding amount of training data will impact the detection ratio.

Table 5. Performance analysis of proposed method on CICIDS 2017.

Type	Number of flows	Accuracy	Recall	Precision	F1-score
Normal	15000	0.964960	0.963	0.94669	0.954
DDOSLOIT	10027	0.973658	0.960	0.9366	0.948
DoSHulk	7034	0.987350	0.963	0.95714	0.960
DoSGoldenEye	3567	0.972862	0.919	0.81478	0.864
DoSSlowhttptest	2124	0.991128	0.962	0.88526	0.922
FTP-Parator	1345	0.991003	0.922	0.83908	0.878
DOSSlowIoris	798	0.997514	0.974	0.90676	0.939
SSH-Parator	345	0.99778	0.907	0.85897	0.882

Table 6. Performance analysis of proposed method on 5G-NIDD dataset.

Type	Number of Flows	Accuracy	Recall	Precision	F1-Score
Benign	25692	0.981565	0.98934	0.98585	0.9875
GoldenEye	2543	0.995701	0.966641	0.97534	0.9709
Torshammer	3452	0.982575	0.905702	0.92473	0.9151
SynFlood	2034	0.995066	0.954438	0.96404	0.9592
Slowloris	943	0.992210	0.82929	0.92023	0.8724

In the 5G-NIDD dataset, a similar issue of insufficient training data arises in a different form in Table 6. The low F1-score and recall values are due to

the model's inability to effectively detect negative data. A potential solution suggested to address this imbalance dataset is to create a balanced dataset using transformer models that will implement in the future. Figure 4 shows the comparison of presented Bi-LSTM based model with existing deep learning algorithms on the ADFA-LD, CICIDS 2017 and 5G-NIDD dataset. It proves the developed Bi-LSTM + MHA has better performance over other models at little level. But this small deviation can cause a much difference in the sensitive networks like the smart grid. It also demonstrates the influence of multihead attention enhances the performance of proposed system. Table 7 presents the comparison of proposed model with the algorithms given in paper 7, [8, 25]. It clearly demonstrates the significance of incorporating attention mechanism in early attack detection. The proposed method shows slightly better performance compared to Djaidja *et al*. [10] but the gap is more compared with [8, 25] models due to the lack of attention mechanism. The F1-score indicates the small difference between recall and precision values that highlights the algorithm performance in positive data detection. Furthermore, it reveals that the proposed method achieves better performance in complex datasets, whereas LSTM+Attention mechanism from Djaidja *et al*. [10] offers strong competition in 5G-NIDD dataset.

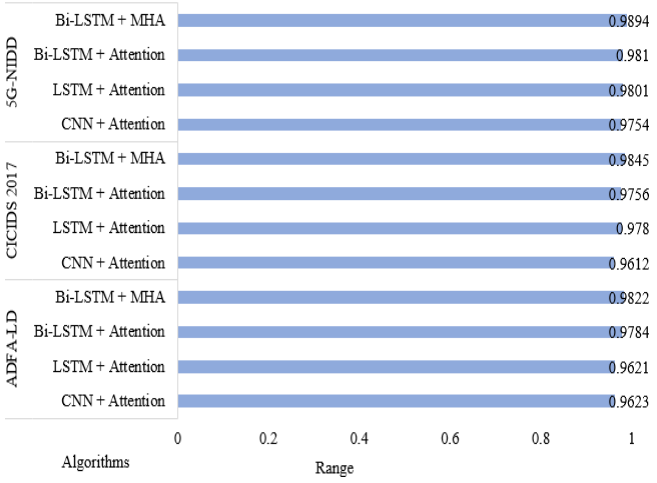

Figure 4. Comparison of proposed and existing algorithms on standard datasets.

Table 7. Comparison of proposed algorithm with existing works on standard datasets.

Dataset	Algorithm	Accuracy	Recall	Precision	F1-Score
ADFA-LD	BI-LSTM+Attention	0.982206	0.933708	0.880416	0.906279
	LSTM+Attention [10]	0.9651	0.9261	0.8734	0.898978
	GRU+Attention [10]	0.9241	0.9021	0.8492	0.874851
	CNN+GRU [8]	0.9011	0.8321	0.8012	0.816358
	CNN+RNN [25]	0.9163	0.8567	0.8671	0.861869
CICIDS-2017	BI-LSTM+Attention	0.984532	0.94625	0.89316	0.918375
	LSTM+Attention [10]	0.9551	0.8961	0.8734	0.898978
	GRU+Attention [10]	0.9235	0.8234	0.9021	0.860955
	CNN+GRU [8]	0.9028	0.8772	0.8363	0.856262
	CNN+RNN [25]	0.8935	0.7943	0.7458	0.769286
5G-NIDD	BI-LSTM+Attention	0.989423	0.929082	0.954038	0.94102
	LSTM+Attention [10]	0.9921	0.9908	0.9876	0.989197
	GRU+Attention [10]	0.9882	0.9324	0.8876	0.909449
	CNN+GRU [8]	0.9534	0.9856	0.8324	0.902545
	CNN+RNN [25]	0.9623	0.9759	0.8457	0.906147

4.4. Early Detection Analysis

This section shows the early detection capability of the proposed model in detecting the attacks on smart meter data communication. It includes the detection time and number of packets that helps to predict the attacks. Figure 5 demonstrates the early detection capabilities of the proposed model on the ADFA-LD dataset. It highlights that attacks are identified before flow completion and also illustrates the overall detection time following flow completion. It indicates that normal and worm data are identified more quickly compared to other attacks. This is attributed to the larger volume of training data for the former class and the significant pattern deviation observed in the worm data.

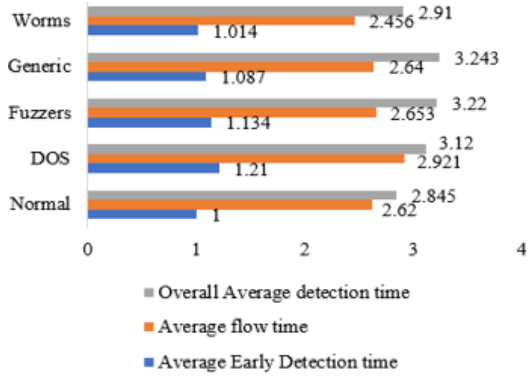

Figure 5. Detection time analysis (in seconds) on ADFA-LD dataset.

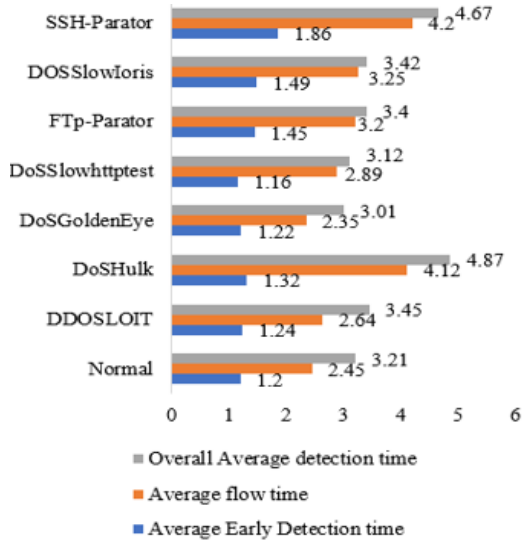

Figure 6. Detection time analysis (in seconds) on CICIDS2017 dataset.

Similarly, Figure 6 presents the average attack detection analysis on CICIDS2017 dataset. It clearly shows that the proposed method outperforms others, achieving detection performance three times better than detecting attacks after flow completion.

Figure 7 displays the performance of the proposed system on IOT environment-based 5G-NIDD dataset. Normal data typically exhibits faster detection times compared to other attacks; however, in this dataset, the SYN flood attack is detected even earlier than normal data. This underscores the effective working of the

proposed model in achieving early attack detection within responsive networks such as smart grids.

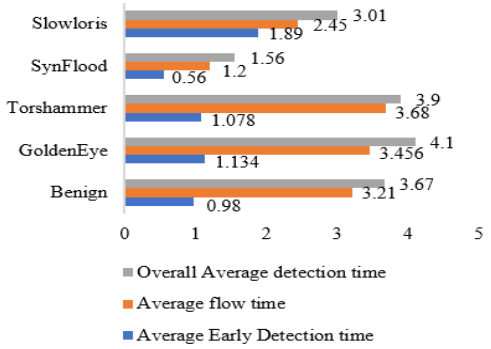

Figure 7. Detection time analysis (in seconds) on 5G-NIDD dataset.

Tables 8, 9, and 10 evaluate the early attack detection on the basis of successive packets required for correct attack classification on standard datasets. It includes the total number of packets generated for each category of flow. Each flow exhibits a different packet range, even within the same data category, as shown in the flow ratio column. This indicates that packets of the same class are analyzed over varying time intervals. A comparison of the tables reveals that the attacks in the 5G-NIDD and CICIDS2017 datasets are simulated with a larger number of packets compared to the ADFA-LD dataset. However, an analysis of the minimum and maximum packets required for detection shows that the necessary packet count does not vary significantly.

Although the attacks are detected earlier, fully analysing the performance remains challenging due to difficulties in collecting data during the simulation process. In the future, it will be resolved by implementing in the real network.

Table 8. Packet required for attack classification on ADFA-LD dataset.

Type	Number of flows	Number of packets generated	Packet to flow ratio	Average packets used for early detection	Min packets used to identify	Max packets used to identify
Normal	3300	16400	4.969697	3	2	3.2
DOS	3200	36400	11.375	3.562	3	6
Fuzzers	1460	14570	9.979452	3.83	2	5
Generic	2600	16850	6.480769	2.987	2	5
Worms	50	800	16	3.6	3	7

Table 9. Packet required for attack classification on CICIDS2017 dataset.

Type	Number of Flows	Number of Packets generated	Packet to flow ratio	Average Packets used for early detection	Min packets used to identify	Max packets used to identify
Normal	15000	330000	22	5	3	12
DDOSLOIT	10027	280756	28	6	5	15
DoSHulk	7034	1062134	151	11	5	16
DoSGoldenEye	3567	49938	14	4	2	6
DoSSlowhttptest	2124	19116	9	3	2	6
FTp-Parator	1345	37660	28	6	4	8
DOSSlowIoris	798	8778	11	5	3	8
SSH-Parator	345	18975	55	8	4	12

Table 10. Packet required for attack classification on 5G-NIDD dataset.

Type	Number of flows	Number of packets generated	Packet to flow ratio	Average Packets used for early detection	Min packets used to identify	Max packets used to identify
Benign	25692	205536	8	4	3	8
GoldenEye	2543	81376	32	12	8	18
Torshammer	3452	120820	35	13	8	19
SynFlood	2034	6102	3	3	2	3
Slowloris	943	136735	145	23	17	43

5. Conclusions

In this work, a hybrid method using BiLSTM and MHA is proposed for the detection of coordinated attacks in smart grid. It identifies multiple attacks earlier before the traffic flow is completed which is much required for highly sensitive networks like smart grid. The performance of proposed algorithm is demonstrated using standard datasets such as ADFA-LD, CICIDS2017 and 5G-NIDD. The traffic flow of standard datasets is evaluated by simulating it using the NS3 simulation tool, based on the meta-information provided by the dataset developers. The research mainly focused on denial-of-service attacks and the result proves that the proposed model has effectively detected the attacks earlier. Limitations such as increased delays encountered during the implementation of the developed model in the simulation will be addressed in future work.

References

[1] Abid A., Jemili F., and Korba O., "Distributed Deep Learning Approach for Intrusion Detection System in Industrial Control Systems Based on Big Data Technique and Transfer Learning," *Journal of Information and Telecommunication*, vol. 7, no. 4, pp. 513-541, 2023. https://doi.org/10.1080/24751839.2023.2239617

[2] Alaidaros H., Mahmuddin M., and Al Mazari A., "An Overview of Flow-based and Packet-based Intrusion Detection Performance in High-Speed Networks," *in Proceedings of the International Arab Conference on Information Technology*,

Riyadh, pp. 1-9, 2011. file:///C:/Users/acit2k/Downloads/AnOverviewofFlow-BasedandPacket-BasedIntrusionDetectionPerformanceinHighSpeedNetworks%20(1).pdf

[3] Alrayes F., Nemri N., Aljaffan N., Alshuhail A., and et al., "Distributed Multiclass Cyberattack Detection using Golden Jackal Optimization with Deep Learning Model for Securing IoT Networks," *IEEE Access*, vol. 12, pp. 132434-132443, 2024. https://doi.org/10.1109/ACCESS.2024.3443202

[4] Anley M., Genovese A., Agostinello D., and Piuri V., "Robust DDoS Attack Detection with Adaptive Transfer Learning," *Computers and Security*, vol. 144, pp. 103962, 2024. https://doi.org/10.1016/j.cose.2024.103962

[5] Chinnasamy R., Subramanian M., and Sengupta N., "Empowering Intrusion Detection Systems: A Synergistic Hybrid Approach with Optimization and Deep Learning Techniques for Network Security," *The International Arab Journal of Information Technology*, vol. 22, no. 1, pp. 60-76, 2025. DOI: 10.34028/iajit/22/1/6

[6] Cisco NetFlow Configuration, https://www.cisco.com/c/dam/en/us/td/docs/security/stealthwatch/netflow/Cisco_NetFlow_Configuration.pdf, Last Visited, 2025.

[7] Creech G. and Hu J., ''Generation of a New IDS Test Dataset: Time to Retire the KDD Collection,'' *in Proceedings of the IEEE Wireless Communication and Networking Conference*, Shanghai, pp. 4487-4492, 2013. https://doi.org/10.1109/WCNC.2013.6555301

[8] Diaba S. and Elmusrati M., "Proposed Algorithm for Smart Grid DDoS Detection based on Deep Learning," *Neural Networks*, vol. 159, pp. 175-184, 2023. https://doi.org/10.1016/j.neunet.2022.12.011

[9] Ding Y., Tian Y., Li X., Mishra Y., and et al., "Constrained Broadcast with Minimized Latency in Neighborhood Area Networks of Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 309-318, 2020. https://doi.org/10.1109/TII.2019.2915826

[10] Djaidja T., Brik B., Senouci S., Boualouache A., and Doudane Y., "Early Network Intrusion Detection Enabled by Attention Mechanisms and RNN," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 7783-7793, 2024. https://doi.org/10.1109/TIFS.2024.3441862

[11] Dong J., Xing L., Cui N., Zhao L., and et al., "Estimating Reference Crop Evapotranspiration Using Improved Convolutional Bidirectional Long Short-Term Memory Network by Multi-Head Attention Mechanism in the Four Climatic Zones of China," *Agricultural Water Management*, vol. 292, pp. 108665, 2024. https://doi.org/10.1016/j.agwat.2023.108665

[12] Guha D., Chatterjee R., Sikdar B., "Anomaly Detection Using LSTM-based Variational Au-Toencoder in Unsupervised Data in Power Grid," *IEEE Systems Journal*, vol. 17, no. 3, pp. 4313-4323, 2023. https://doi.org/10.1109/JSYST.2023.3266554

[13] Hassan N., Miah A., and Shin J., "A Deep Bidirectional LSTM Model Enhanced by Transfer-Learning-based Feature Extraction for Dynamic Human Activity Recognition," *Applied Sciences*, vol. 14, pp. 14020603, 2024. https://doi.org/10.3390/app14020603

[14] Hu C., Yan J., and Liu X., "Adaptive Feature Boosting of Multi Sourced Deep Autoencoders for Smart Grid Intrusion Detection," *in Proceedings of the IEEE Power and Energy Society General Meeting*, Montreal, pp. 1-5, 2020. https://doi.org/10.1109/PESGM41954.2020.9281934

[15] Jayalaxmi P., Saha R., Kumar G., Alazab M., and et al., "PIGNUS: A Deep Learning Model for IDS in Industrial Internet-of-Things," *Computers and Security*, vol. 132, pp. 103315, 2023. https://doi.org/10.1016/j.cose.2023.103315

[16] Kardi M., Alskaif T., Tekinerdogan B., and Catalao J., "Anomaly Detection in Electric-Ity Consumption Data Using Deep Learning," *in Proceedings of the IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe*, Bari, pp. 1-6, 2021. https://doi.org/10.1109/EEEIC/ICPSEurope51590.2021.9584650

[17] Kasongo S. and Sen Y., "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on a UNSW-NB15 Dataset," *Journal of Big Data*, vol. 7, no.1, pp. 1-20, 2020. https://doi.org/10.1186/s40537-020-00379-6

[18] Khan M., Houran M., Kauhaniemi K., Zafar M., and et al., "Efficient State of Charge Estimation of Lithium-Ion Batteries in Electric Vehicles Using Evolutionary Intelligence-Assisted GLA-CNN-Bi-LSTM Deep Learning Model," *Heliyon*, vol. 10, pp. 1-20, 2024. https://doi.org/10.1016/j.heliyon.2024.e35183

[19] Kumar S., "Survey of Current Network Intrusion Detection Techniques," *Journal of Information Security*, pp. 1-18, 2007. https://www.scirp.org/reference/referencespapers?referenceid=436521

[20] Lansky J., Ali S., Mohammadi M., Majeed M., Karim S., and Rashidi S., "Deep Learning-based Intrusion Detection Systems: A Systematic Review," *IEEE Access*, vol. 9, pp. 101574-101599, 2021. https://doi.org/10.1109/ACCESS.2021.3097247

[21] Li H., Ge H., Sang Y., and Gao C., "An Optimized

Multi-Layer Ensemble Model for Airborne Networks Intrusion Detection," *Applied Soft Computing*, vol. 167, pp. 112282, 2024. https://doi.org/10.1016/j.asoc.2024.112282

[22] Li Z., Zhong Z., Zuo P., and Zhao H., "A Personalized Federated Learning Method Based on the Residual Multi-Head Attention Mechanism," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, pp. 102043, 2024. https://doi.org/10.1016/j.jksuci.2024.102043

[23] Mahmud R., Vallakati R., Mukherjee A., Ranganathan P., and Nejadpak A., "A Survey on Smart Grid Metering Infrastructures: Threads and Solutions," *in Proceedings of the IEEE International Conference on Electro/Information Technology*, Dekalb, pp. 386-391, 2015. https://doi.org/10.1109/EIT.2015.7293374

[24] Mokbal F., Dan W., Osman M., Ping Y., and Alsamhi S., "An Efficient Intrusion Detection Framework Based on Embedding Feature Selection and Ensemble Learning Technique," *The International Arab Journal of Information Technology*, vol. 19, no. 2, pp. 237-248, 2022. DOI: 10.34028/iajit/19/2/11

[25] Peng W., Kong X., Peng G., Li X., and Wang Z., "Network Intrusion Detection based on Deep Learning," *in Proceedings of the International Conference on Communications, Information System and Computer Engineering*, Haikou, pp. 431-435, 2019. https://doi.org/10.1109/CISCE.2019.00102

[26] Pietro R. and Mancini L., *Intrusion Detection Systems*, Springer New York, 2008. https://link.springer.com/book/10.1007/978-0-387-77265-3

[27] Samarakoon S., Siriwardhana Y., Porambage P., Liyanage M., and et al., "5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated Over 5G Wireless Network," *Cryptography and Security*, 2022. https://arxiv.org/abs/2212.01298v1

[28] Sharafaldin I., Lashkari A., and Ghorbani A., "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *in Proceedings of the 4th International Conference on Information System Security and Privacy*, Funchal, pp. 108-116, 2018. https://doi.org/10.5220/0006639801080116

[29] Umer M., Sher M., and Bi Y., "Flow-based Intrusion Detection: Techniques and Challenges," *Computers and Security*, vol. 70, pp. 238-254, 2017. https://doi.org/10.1016/j.cose.2017.05.009

[30] Vijayanand R., Devaraj D., and Kannapiran B., "A Novel Intrusion Detection System for Wireless Mesh Network with Hybrid Feature Selection Technique Based on GA and MI," *Journal of Intelligent and Fuzzy Systems: Applications in Engineering and Technology*, vol. 34, pp. 1243-1250, 2018. https://doi.org/10.3233/JIFS-169421

[31] Wang N., Chen Y., Xiao Y., Hu Y., Lou W., and Hou Y., "Manda: On Adversarial Example Detection for Network Intrusion Detection System," *in Proceedings of the IEEE Transactions on Dependable and Secure Computing*, Vancouver, pp. 1139-1153, 2023. https://doi.org/10.1109/INFOCOM42981.2021.9488874

[32] Yakubu B., Khan M., Khan A., Jabeen F., and Jeon G., "Blockchain-based DDoS Attack Mitigation Protocol for Device-to-Device Interaction in Smart Home," *Digital Communications and Networks*, vol. 9, pp. 383-392, 2023. https://doi.org/10.1016/j.dcan.2023.01.013

Vijayanand Radhakrishnan is working as an Associate Professor in Jain (Deemed-to-be) University, Bengaluru. He has more than 13 years of experience in Teaching and Research. His research interest includes Machine Learning, Network security, IOT and AMI communication of smart grid.

Ulaganathan Meenakshisundaram is working as an Associate Professor in PSR Engineering College, Sivakasi. He has more than 16 years of Teaching and Research. His research interest includes Machine Learning, Smart grid, MPPT tracking and Emulators.

Ganeshkumar Pugalendhi is an Associate Professor in College of Engineering, Anna University, Guindy. He has more than 20 years of Teaching and Research experience. His research interest includes Network Security, Image proc, Optimization Techniques and Deep Learning algorithms

John Basha is working as an Assistant Professor in Jain (Deemed-to-be) University, Bengaluru. He has more than 18 years of experience in Teaching. His research interest includes Data Processing, Network Security, Deep Learning algorithms, and Quantum Computing.