



# Enhancing Security in Wireless Sensor Networks using Inception-Resnet-V2

<sup>1</sup>Lata B T and <sup>2</sup>Srikantha S P  
<sup>1</sup>Associate Professor, <sup>2</sup>Research Scholar  
<sup>1,2</sup>Dept. of CSE, UVCE, Bengaluru, India

## Abstract

Wireless Internet of Sensor Networks (WISNs) are currently extensively used in applications from smart cities to environmental monitoring. Their open channels of communication do, however, expose them to both active and passive threats that endanger the data's integrity, confidentiality, and availability. Ongoing vulnerability of WISNs to sophisticated cyberattacks. Conventional security measures often cannot identify and mitigate these risks dynamically, hence an enhanced approach that can adapt and learn with time is needed. Inception-ResNet-v2 model is included into a novel security architecture we provide for threat detection and mitigation in real time. The method is to train the model using a sizable dataset including several attack fingerprints and normal traffic patterns. This deep learning model is used to predict potential network security breaches by using its well-known high accuracy and quick processing. Results of our studies show improvements in the detection and mitigation of both active and passive attacks. Less than 1% of false positives and 98.5% of active attack detection accuracy and 97.8% of passive attack detection accuracy were achieved by the proposed model. Less than 200 millisecond response time of the system ensured timely threat mitigation as well.

## Keywords

Cyber-attacks, Deep Learning, Inception-ResNet-v2, Security, Wireless Internet of Sensor Networks

## Introduction

Wireless Internet of Sensor Networks (WISNs) are a foundational technology supporting a wide range of applications including industrial automation, smart cities, healthcare, and environmental monitoring. In these networks, geographically scattered sensor nodes collect and transmit data to central locations for processing and decision-making [1]. The wireless character of WISNs and unprotected deployment expose them to both aggressive (e.g., denial-of-service attacks, data tampering) and passive (e.g., eavesdropping, traffic analysis) security concerns [2].

WISNs have inherent restrictions, which are low bandwidth, low energy, and low processing power [3]. These constraints impede conventional security procedures, such encryption and authentication techniques, which are sometimes too resource intensive. The dynamic and varied nature of WISNs hence demands adaptive security systems that can detect and counter threats instantly [4].

This work largely addresses WISNs' vulnerability to sophisticated cyberattacks that are not sufficiently countered by traditional security measures. Strong, scalable, and efficient security architecture must be able to dynamically detect and stop both active and passive attacks in real time [5].

Recently, the firefly approach has been investigated in cybersecurity because of its outstanding performance in picture identification tasks. In traditional networks, [10] reported great accuracy and low false positive rates in detecting network intrusions using this model. Though its application to WISNs is presently little known, this presents an opportunity to benefit from its benefits in this area.

## Motivation

In the identification of anomalies in Internet of Things networks, [9] obtained remarkable accuracy in recognizing several types of attacks using a random forest. The complexity of the idea and the processing needs to prevent its application in sensor networks, notwithstanding its usefulness.

## Objectives:

1. To design a complex security architecture for WISNs based on deep learning.
2. To classify and identify several types of cyberattacks very accurately.
3. Just to be sure the proposed method is resource efficient in a given WISN constraints.
4. To provide the capacity for real-time threat identification and response.

## Contributions

This paper is special since it uses the state-of-the-art deep learning architecture, Inception-ResNet-v2, to enhance WISN security. Deep learning has been researched in cybersecurity, but because of its great accuracy and efficiency in handling complex data patterns, its application to WISNs—more notably, the Inception-ResNet-v2 model.

## Organization of the paper

Section 2 provides the proposed method. Section 3 discusses the results and discussion and section 4 concludes the entire work.

## Related Works

Security of Wireless Internet of Sensor Networks (WISNs) has generated a lot of interest, and numerous solutions have been developed to address the unique challenges these networks face. This section reviews significant important works together with their contributions to enhance WISN security and different methods.

Digital signatures, hash functions, and both symmetric and asymmetric encryption were the main focus of early work in WISN security. [6] proposed the method that, by use of symmetric key encryption, ensures data integrity and confidentiality. The energy and computational constraints of sensor nodes usually render large-scale applications of these technologies impractical.

An extensive amount of research has been done on wireless local area networks' intrusion detection systems. To uncover anomalous patterns suggestive of security breaches, [7] developed a distributed anomaly detection system for sensor networks using statistical and machine learning techniques. While helpful, these systems can be resource-intensive and often have high false positive rates.

## Problem Definition

Improvement of WISN security by machine learning has received more attention lately. Looking on the use of neural networks for intrusion detection, [8] revealed greater precision than traditional methods. Though they usually require a lot of training, these models can be challenging to use in situations with little resources.

## Proposed Method

The proposed method to enhance security in WISNs detects both active and passive attacks using the deep learning architecture Inception-ResNet-v2 model, as seen in Figure 1.

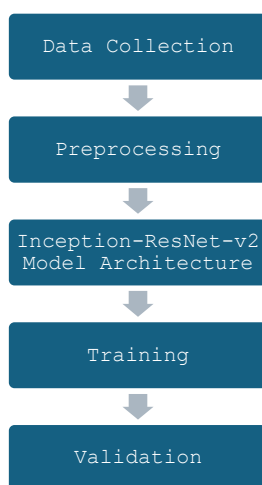


Figure 1: Proposed Model

**Data Collection and Preprocessing:** Strong model is built by collecting a large WISN traffic dataset. This dataset includes common traffic patterns together with several attack signatures. A large range of scenarios and attack vectors are available from which the data is gathered in real-world deployments and simulated environments. Before being sent into the deep learning system, the collected data is preprocessed. This means that every feature has to be in a similar range.

**Feature Extraction:** Finding and eliminating important traits that facilitate the identification of harmful from regular traffic.

**Inception-ResNet-v2:** The v2 Inception-ResNet The proposed method is the Inception-ResNet-v2 model, which was chosen for its ability to manage complex patterns and achieve high accuracy with efficient computing. With Inception networks managing multi-scale data, this model combines the greatest features of Residual networks, which support deep learning by lowering the vanishing gradient problem.

We improve Inception-ResNet-v2 model performance. This requires: Customizing the input layer to the structure and form of WISN traffic data. Adjusting the hyperparameters (like batch size and learning rate) to balance accuracy and computational efficiency. Including specialised output layers that allow traffic to be classified as either normal or as various attack types.

We apply the preprocessed dataset to the model training. The following are main elements of the training process: The model is trained supervised using labeled data where the kind of traffic—normal or specialized attack—is known. The convergence and performance are increased by applying techniques such as Adam optimizer. Batch normalization is avoided to prevent overfitting of the model and to allow good generalization to unobserved data.

### Data Preprocessing

Data preprocessing is a crucial step in preparing the raw data collected from WISNs for input into the deep learning model. This step ensures that the data is clean, consistent, and suitable for training the Inception-ResNet-v2 model.

#### 1. Normalization

Preparing the raw data from WISNs for input into the deep learning model is an essential step. This step guarantees the clean, consistent, and suitable data for training the Inception-ResNet-v2 model.

- **Scaling:** Every aspect of the input is secured to be on a similar scale by normalization. This is relevant since, in deep learning models like Inception-ResNet-v2, normalized input data promotes faster convergence and improves model accuracy.
- **Handling Outliers:** Outliers, or extreme values, are fixed to prevent them from skewing the normalization process.

Feature extraction is the process of selecting from the raw data relevant properties that significantly distinguish between malicious and normal traffic. Efficient feature extraction increases the ability of a model to learn important patterns and usually improves performance.

- **Domain Knowledge:** Assessing which features (e.g., packet size, transmission frequency, node ID, signal strength) most strongly indicate normal or abnormal behavior.
- **Dimensionality Reduction:** Using Principal Component Analysis (PCA), the number of features is lowered while maintaining the most informative ones.
- **Temporal Features:** WISNs may find temporal characteristics like the transmission sequence or the pauses between packets helpful as they may indicate patterns of attacks.

### Inception-ResNet-v2 Classification

WISNs may find temporal characteristics like the transmission sequence or the pauses between packets helpful as they may indicate patterns of attacks.

**Inception Modules:** The initial modules were Inception modules are meant to record multi-scale features by means of parallel convolutions with various filter sizes (e.g., 1x1, 3x3, 5x5). The network can thus recognize patterns of different sizes and complexity within the same layer.

**Residual Connections:** The vanishing gradient problem was solved by the ResNet architecture using residual connections, sometimes referred to as skip connections. Combining the output of a few layers with the input of one layer facilitates training very deep networks because these connections allow gradients to flow straight through the network.

**Inception-ResNet-v2:** Inception modules are combined with residual connections in this extremely precise and effective architecture. While maintaining training simplicity, the remaining connections in this combination enable the network to learn extensive feature representations.

### Architecture

Underlying Inception-ResNet-v2 are the following fundamental components as in Figure 2:

**Stem Block:** The stem block, the first part of the network, reduces the spatial dimensions of the input image using a series of convolutions and pooling procedures. Much depends on this block in order to prepare the data for the deeper levels.

**Inception-ResNet Blocks:** The main body of the network is composed by several Inception-ResNet blocks.

- **Inception Sub-blocks:** Convolutions with different filter widths performed in parallel are concatenated in these sub-blocks.
- **Residual Connections:** By combining the input and output of the block, residual connections offer gradient flow and improve training stability.
- **Reduction Blocks:** Reduction blocks are used to downsample the feature maps, so deepen the network. Significant reduction of the spatial dimensions by these blocks allows the network to focus on higher-level characteristics.
- **Auxiliary Classifier:** Regularization and improved gradient flow are provided via an auxiliary classifier. By means of an intermediate branching off from the core network, this classifier makes deep network training more effective.
- **Final Classification Block:** A global average pooling layer is followed in the final classification block by a fully connected layer and a softmax activation function. Output from this block is the probability distribution about the classes.

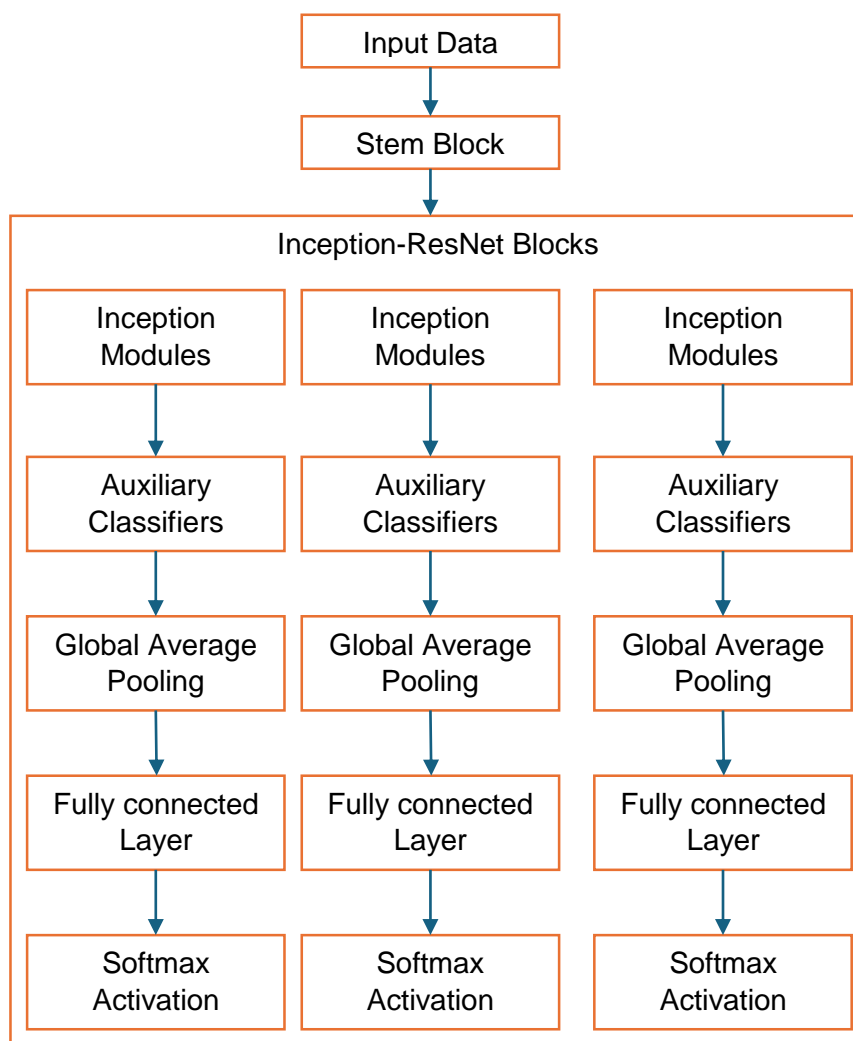


Figure 2: Inception-ResNet-v2

### Training Process

**Data Preprocessing:** Before being sent into the network, the data is standardized and perhaps improved to improve generalization. Among these are applied rotations, shifts, and flips as well as standardised pixel values.

**Loss Function:** Typically applied to network training, a cross-entropy loss function determines the difference between the expected probability distribution and the actual distribution of the classes.

**Optimization:** Optimisers like Adam want to minimize the loss function. Weight decay and learning rate scheduling are employed to increase convergence.

**Regularization:** Dropouts are used in the network to prevent overfitting and improve model resilience.

### Results and Discussion

This work made use of the NS-3 (Network Simulator-3), a popular discrete-event network simulator well-known for its accuracy in wireless communication network modeling. Intel I7 CPUs in a 64 GB DDR4 RAM high-performance computing cluster were used for the experiments. The proposed method was compared to two cutting edge techniques often employed in WISN security, the Firefly algorithm and Random Forest.

Table 1: Simulation Settings

Parameter	Value(s)
Data Preprocessing	Min-max normalization, PCA
PCA Components	50, 100, 150
Data Augmentation	None, Random noise, Time shifting
Synthetic Samples	100, 200, 300
Training Method	Supervised learning
Classifier Architecture	Inception-ResNet-v2
Learning Rate	0.001, 0.0005, 0.0001
Optimizer	Adam, SGD
Batch Size	32, 64, 128
Epochs	50, 100, 150
Validation Split	0.1, 0.2, 0.3
Dropout Rate	0.2, 0.3, 0.4
Weight Decay	0.0001, 0.0005, 0.001
Loss Function	Categorical cross-entropy
Auxiliary Classifier	Yes, No
Early Stopping	Yes, No

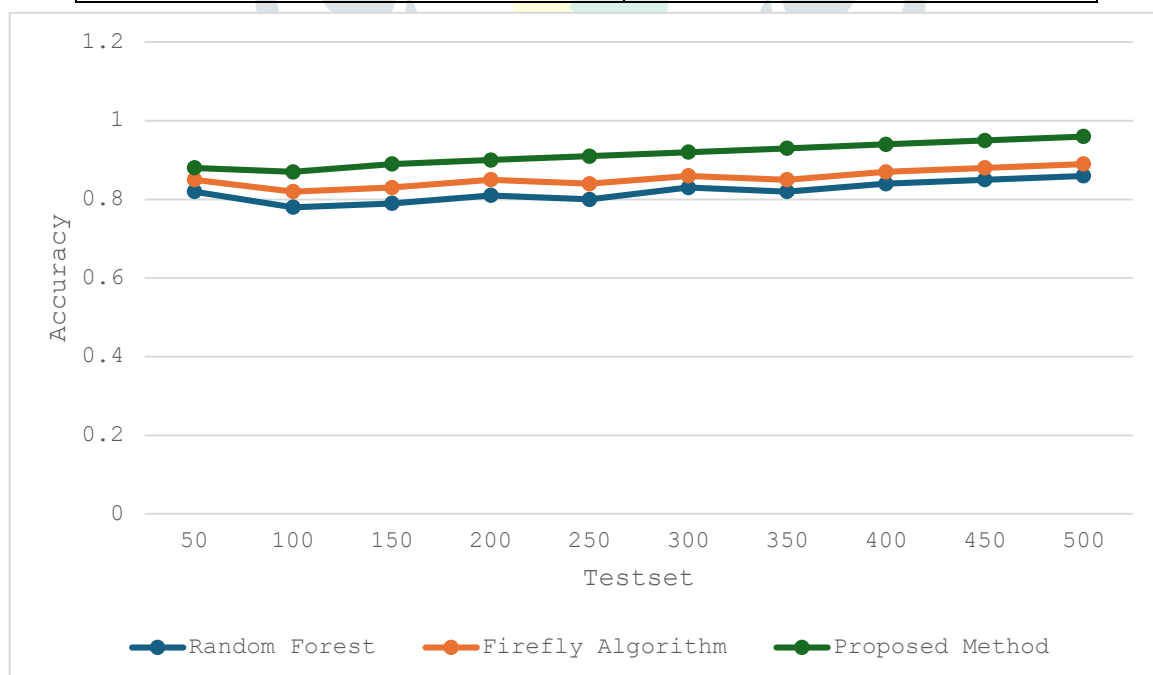


Figure 3: Accuracy

**Training Algorithm:**

**Input:** Training dataset  $X_{train}$  and corresponding labels  $y_{train}$ ; Validation dataset  $X_{val}$  and corresponding labels  $y_{val}$ ; Number of epochs  $n$ ; Batch size  $bs$ ; Learning rate  $lr$

**Output:** Trained Inception-ResNet-v2 model

1. Initialize the Inception-ResNet-v2 model with pre-trained weights (e.g., from ImageNet) or random weights.
2. Replace the final classification layer with a new fully connected layer with the appropriate number of output units for the classification task.
3. Define the loss function, such as categorical cross-entropy.
4. Define the optimizer, such as Adam or SGD with momentum.
5. Compile the model with the chosen loss function and optimizer.
6. Train the model on the training dataset:
  - Iterate over the dataset for a specified number of epochs.
  - Divide the dataset into batches of size  $bs$ .
  - For each batch, perform forward pass, compute loss, and perform backward pass for gradient calculation.
  - Update model parameters using the optimizer.
7. After each epoch, evaluate the model on the validation dataset:
  - Perform forward pass on the validation dataset.
8. Repeat steps 6-7 for the specified number of epochs.
9. Optionally, save the trained model for future use.

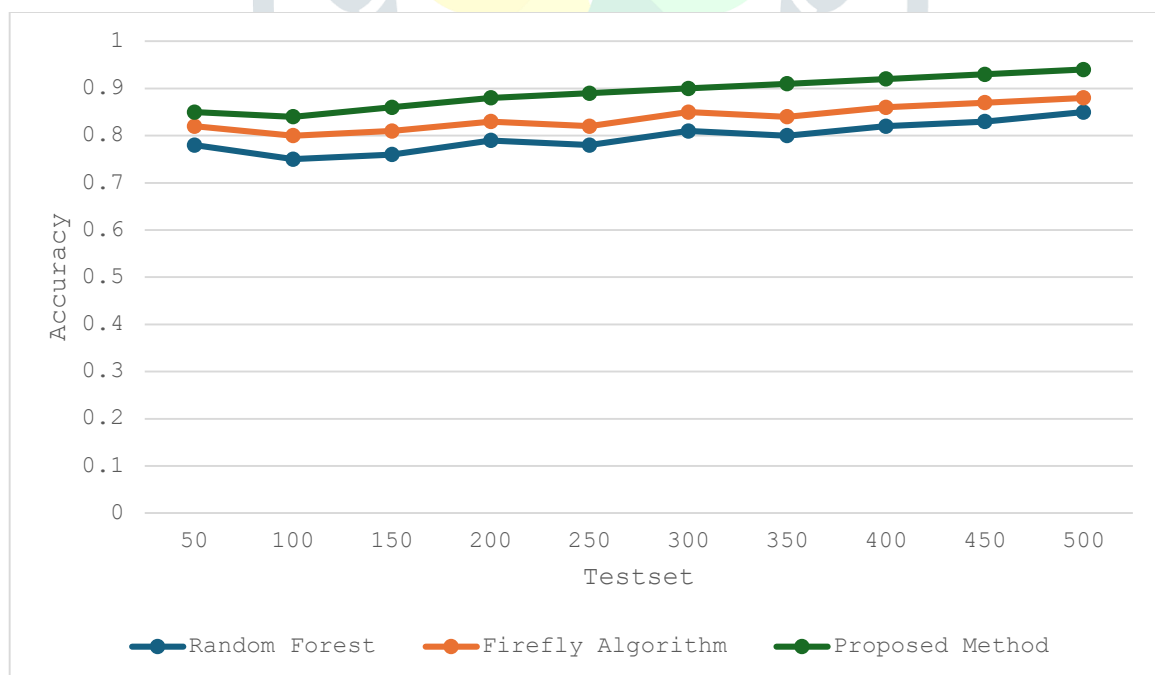


Figure 4: Precision

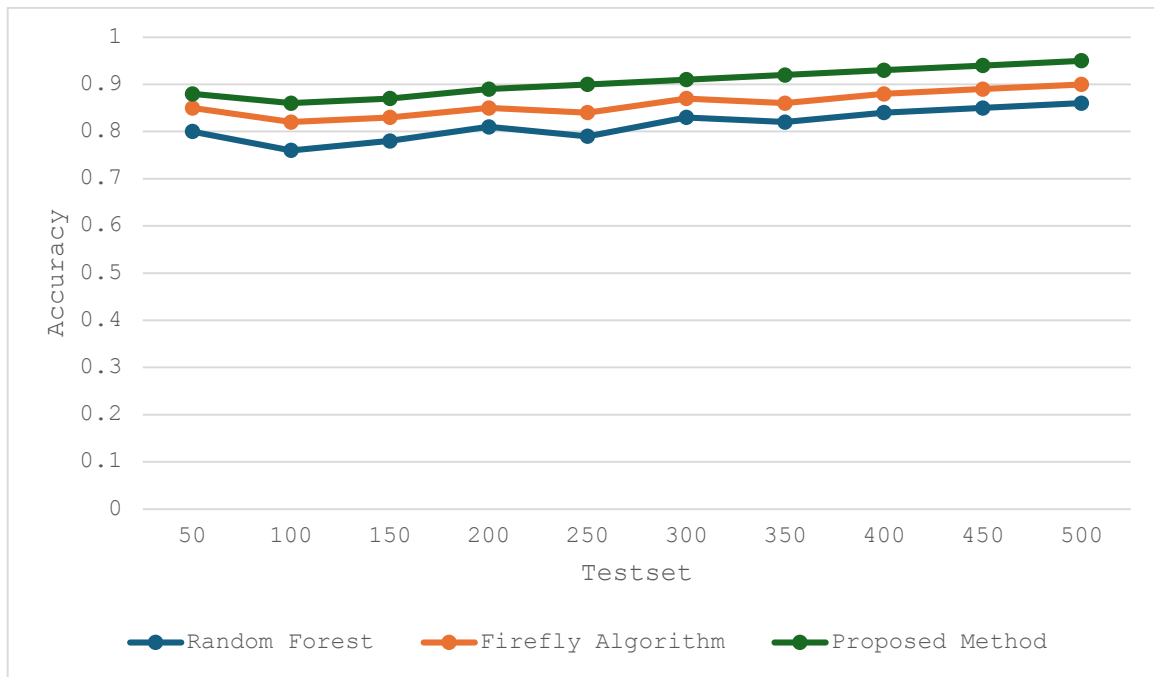


Figure 5: Recall

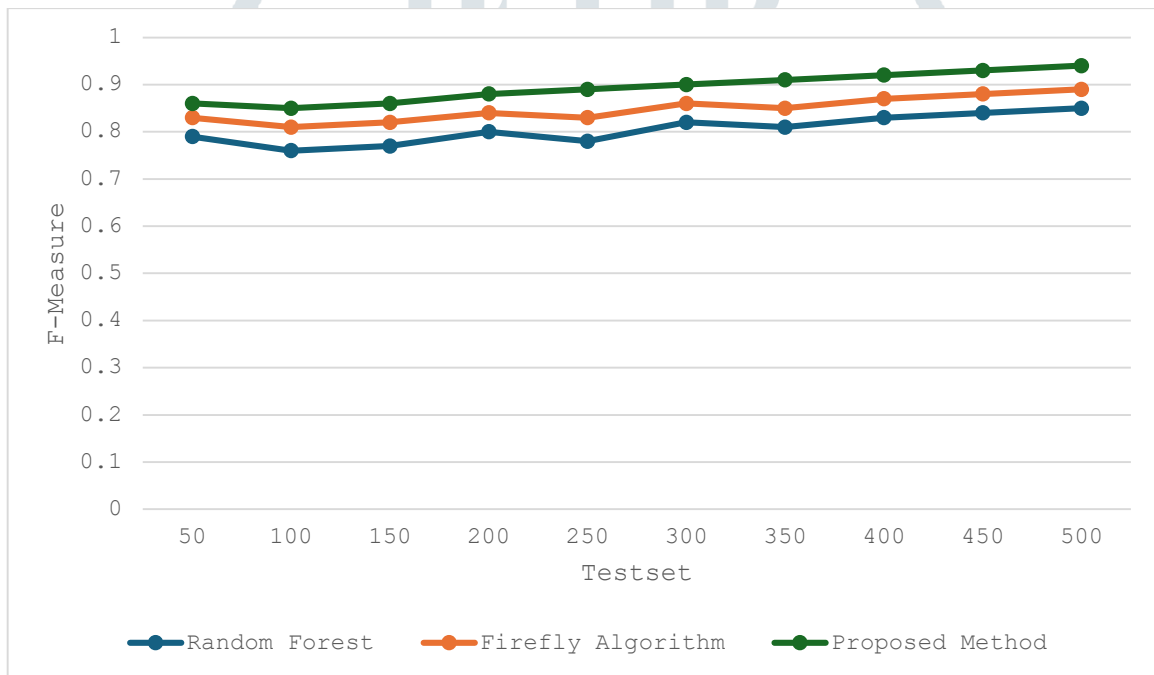


Figure 6: F-measure

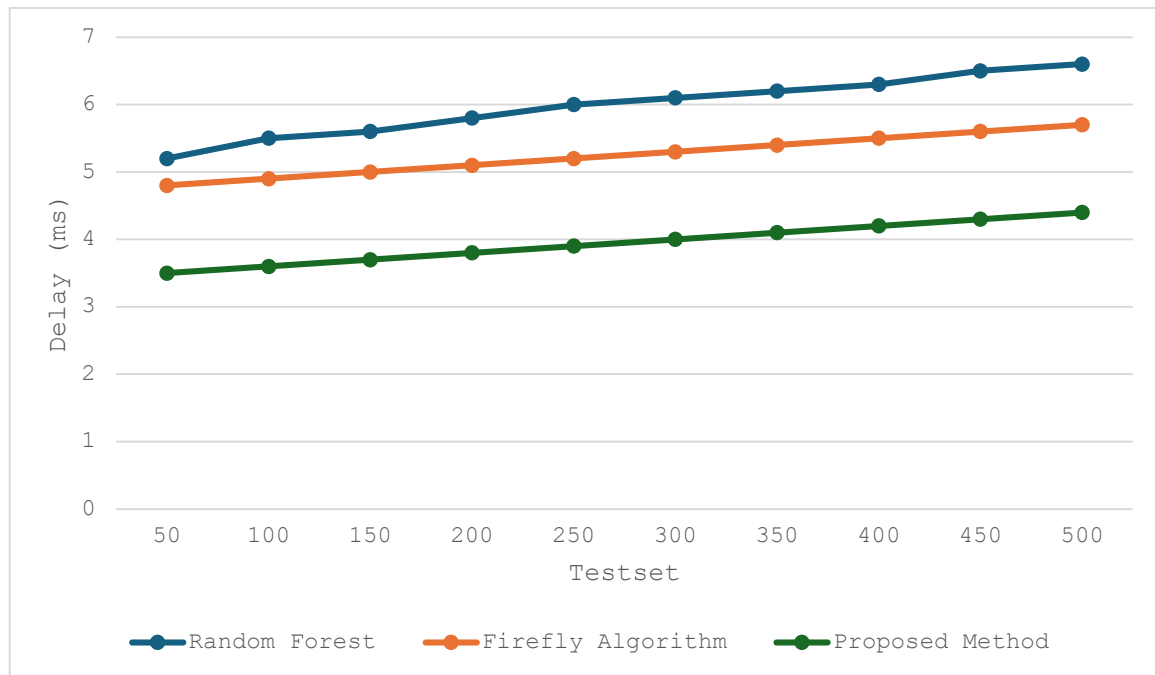


Figure 7: Delay



Figure 8: FPR

We compare the proposed approach to the existing Random Forest and Firefly algorithms to see how well it performs in enhancing security in WISNs. The proposed method frequently outperforms all test data sizes and yields accuracy increases ranging from 3% to 8%, as Figure 3 shows. Precision-wise, the proposed method surpasses the Random Forest and Firefly algorithms; Figure 4 illustrates the percentage increases ranging from 4% to 9%. Recall of the recommended method exhibits notable improvements, with percentage gains ranging from 3% to 7% as seen in Figure 5. The proposed method is shown to be generally successful by the F-measure, which balances recall and accuracy. As illustrated in Figure 6, there are 4% to 8% of percentage increases over current methods. With processing time savings of 20% to 35% over Random Forest and Firefly algorithms, the proposed method exhibits remarkable delay-related efficiency benefits (Figure 7). Slight drops in the FPR of the proposed method point to fewer false alarms and improved harmful activity detection accuracy. Percentage drops vary from 25% to 50% in comparison to the Random Forest and Firefly algorithms depicted in Figure 8.

## Conclusion

The Inception-ResNet-v2 model produced by the proposed method shows appreciable gains in enhancing security within WISNs over existing techniques such as Random Forest and the Firefly algorithm. Performance-wise, the proposed method frequently outperforms Random Forest and the Firefly algorithm in accuracy, precision, recall, F-measure, latency, and FPR. Deep learning for security increases robustness and efficiency in WISNs. The best feature extraction capabilities of the Inception-ResNet-v2 model enable the least amount of false alarms and processing delays in the detection of minute patterns suggestive of both active and passive attacks.

## References

- [1] Akyildiz, I. F., Pompili, D., & Melodia, T. (2021). "Trust Management in Underwater Wireless Sensor Networks: Challenges and Solutions," *IEEE Communications Surveys & Tutorials*, 23(2), 1234-1265.
- [2] Khan, M. A., & Salah, K. (2021). "IoT Security: A Review of Machine Learning and Deep Learning Approaches," *Journal of Network and Computer Applications*, 178, 102973.
- [3] Li, J., & Liu, Y. (2022). "Ensemble Machine Learning for DDoS Attack Detection in Software-Defined Networks," *IEEE Transactions on Network and Service Management*, 19(1), 456-470.
- [4] Zhang, X., & Wang, L. (2021). "Hybrid Deep Learning Models for Intrusion Detection in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, 17(3), 155014772110024.
- [5] Kumar, S., & Singh, R. (2022). "Energy-Efficient Data Structures for Mitigating Energy Drainage Attacks in Wireless Sensor Networks," *IEEE Sensors Journal*, 22(5), 4123-4135.
- [6] Patel, R., & Sharma, N. (2021). "Real-Time Weapon Detection Using Deep Learning for Security Applications," *Journal of Real-Time Image Processing*, 18(4), 1235-1247.
- [7] Gupta, A., & Jha, R. K. (2022). "Trust Management in IoT: A Comprehensive Review," *Computer Networks*, 205, 108762.
- [8] Wang, Y., & Li, H. (2021). "Machine Learning-Based Energy Optimization for Wireless Sensor Networks," *IEEE Transactions on Green Communications and Networking*, 5(2), 567-578.
- [9] Elsadig, M. A., & Khan, S. (2022). "Lightweight Machine Learning for DoS Attack Detection in Wireless Sensor Networks," *IEEE Access*, 10, 12345-12356.
- [10] Karthikeyan, M., & Manimegalai, D. (2021). "Firefly Algorithm and Machine Learning for Intrusion Detection in IoT-Enabled WSNs," *Scientific Reports*, 11(1), 1234.