Digital Object Identifier

A Novel Approach to Intrusion Detection in IoT Networks Using Wavelet Transforms

Dr. Pradeep Kumar Bhale (Member IEEE), Darpan Maurya,Vaibhav Sodhi
Tabish Farooqui and Tapase Rahul
Department of Computer Science and Engineering, Indian Institute of Information Technology Pune, India (e-mails: pradeep.bhale@iiitp.ac.in, vaibhav.sodhi@iiitp.ac.in, tabish.farooqui@iiitp.ac.in, harsh.singh@iiitp.ac.in)

Corresponding author: Dr. Pradeep Kumar Bhale (e-mail: pradeep.bhale@iiitp.ac.in).

This research was conducted under the supervision of Dr. Pradeep Kumar Bhale, with support from IIIT Pune.

ABSTRACT This paper presents a novel approach for detecting intrusions in today's exponentially growing IoT network. Our model aims to perform attack detection on various network related attacks such as DDoS attacks, Probing attacks, R2L, U2R, as well as some WSN specific attacks such as Grayhole, Scheduling attacks, Blackhole among other attacks. We have extracted features by converting data into both time domain and frequency domain using wavelet transforms to capture even the smallest details which is often lost in time domain data. We have used NSL-KDD, WSN-DS and NF-TON-IOT dataset for training of our Machine learning model and for classification we have used classifiers like Decision tree, Random Forest, Adaboost, Catboost and Multilayer Perceptron(MLP). The promising results gives an insight into how accurate Intrusion Detection can be by going from time domain into frequency domain.

INDEX TERMS Key Terms: Internet of Things (IoT), Network Attack Detection, Wavelet Transforms, Frequency domain, Machine Learning.

I. INTRODUCTION

The Internet of Things (IoT) constitutes a complex network of tangible devices that possess the capability to engage in communication amongst themselves devoid of any human intervention, for example smart home devices like lights and AC's can be controlled from anywhere using our mobile phone. IOT represents a framework wherein common place objects such as smartphones, automobiles, and certain household appliances are linked to the internet.

Typically, IOT devices operate in attack prone environments which increases the chance that they will be affected by intruders . If these issues are not solved or properly managed then there is a significant risk of device failure and leakage of private information at any moment leading to integrity concerns.

IoT devices are frequently placed in challenging and unsecured environments, which increases their susceptibility to various network attacks. If security concerns are not properly managed, there is a significant risk that sensitive information could be exposed at any moment.

An Intrusion Detection System (IDS) is simply a network or a device monitoring tool which is designed in such a way that it can detect malicious activities across multiple IOT nodes. It can serve as an additional support system to prevent network based or host based attacks. An IDS incorporates techniques of analysing patterns in malicious and fishy network traffics and doing continuous traffic monitoring and alerting the user as soon as intrusions are detected. In the same way an IDS can be operated on individual devices to detect any anomalies or irregularities within the device itself. An IDS can be deployed through software as well hardware devices , in this paper we have discussed about the software one.

We are using wavelet transforms to convert data from time domain into frequency domain for network attack detection. Wavelet transforms are very useful tools that cuts up signals into a set of basis functions which is done using contractions or expansions. The word Wavelet means small waves which do not last forever. They are different from traditional fourier transform as fourier transforms only provide information about the frequency but wavelets tell us about both time as well as frequency characteristics. This property of wavelet transforms make them effective for studying non-stationary signals.

Recent studies have used wavelet transforms for tasks like finding anomalies, extracting features, and classifying signals in IoT systems. For example, some research shows that wavelet methods can improve the accuracy of detecting network intrusions by giving a clearer picture of how signals change. Wavelet transforms are also widely used for image processing. But many existing methods often use only a few wavelet functions or don't consider other relevant information like unique feature extraction from frequencies, which can make it harder to spot new types of attacks.

Our proposed model overcomes these challenges by using a hybrid approach which uses wavelet transforms with machine learning techniques and detailed feature extraction techniques. By using a variety of features like percentage deviation, entropy, kurtosis values, skewness coefficients our model improves at classifying network anomalies.We also focus on reducing computational power through specific dimensionality reduction methods, which will allow the model to work well in real-time situations.

We decided on applying wavelet transforms after comparing it's accuracy after using techniques such as Fast Fourier Transform (FFT), Short-Time Fourier Transform (STFT), Z-transform, Hilbert Transform etc. Similarly, for different classifiers, we experimented with several options such as Random Forest (RF), Decision Trees(DT), AdaBoost, CatBoost, and Multi-layer Perceptron (MLP). Upon careful analysis of their performance metrics, we concluded that the AdaBoost, CatBoost, and MLP classifiers best suited our particular application scenario with proper accuracy as well as efficiency. The dataset used for training purpose are NSLKDD , WSN-DS and NF-TON-IOT.

In further sections we will give some introduction to the background in which we are working that is Internet Of Things (IOT), then we will discuss our proposed work(methodology) and we conclude by discussing the results obtained from our experiments.

II. BACKGROUND

A. INTERNET OF THINGS

Internet of things (IoT) can be referred as a network of various physical devices that communicate with each other through the internet. These physical devices are typically embedded with sensors, softwares and other technologies that connect and exchange data. IoT devices are typically resource-constrained nodes and have low power consumption. Nowadays, almost any appliance around us, whether at home or at our place of work, are connected to each other through the internet in one way or another and can be regarded as a part of an IoT infrastructure.

However, an IoT network is highly susceptible to a variety of network related attacks which will be discussed in the upcoming section. Hence, there arises a need to develop strong countermeasures against these types of network attacks. Modern IDS typically use a Machine Learning or a Deep Learning based approach to automate the process of detection of an anomaly within the network traffic.

B. MACHINE LEARNING IN IDS

. ML have significant advantages in intrusion detection for IoT systems.It excels in processing large volume of data and detecting any pattern that indicates an anomaly. Unlike Deep Learning algorithms ML algorithms take less time which is important for network bandwidth. The integration of ML and IDS is rapidly evolving, offering solutions to previously unsolvable cybersecurity challenges and significantly enhancing the detection of both known and emerging threats. Based on the architecture, IDS can be of two types : Host-based IDS (HIDS) or a Network-based IDS (NIDS). Our approach will be focused on NIDS. Now, on the basis of the method of detection, IDS can be further classified into two types : Anomaly-based IDS or Signature-based IDS. Since, the dataset which is being used to train our model is labeled, i.e., our model is a supervised learning model, our IDS will be a Signature-based IDS.

C. ATTACKS

There are multiple types of attacks possible on an IoT network. However, since our model is a NIDS, our main focus will be on network-related attacks. Such attacks may include Distributed Denial of Service (DDoS) attacks, Man-in-the-Middle (MitM) attacks, Replay Attacks, Eavesdropping or Data Interception attacks, Scanning or Probing attacks, Key-logging, etc. Some examples of WSN attacks are Sinkhole, Wormhole, Blackhole, Selective forwarding, etc.

D. WAVELET TRANSFORM

Unlike Fourier Transforms, Wavelet transforms are able to decompose signals into both time and frequency domain.Wavelet transform can be a powerfull tool for analysing non stationary signals. A family of functions called wavelets are localized in both time and frequency. wavelets are scaled and shifted versions of a Mother wavelet.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi \left(\frac{t-b}{a} \right)$$

$\psi_{a,b}(t)$: The scaled and translated version of the mother wavelet $\psi(t)$, defined by the parameters a and b.

a: The scale (or dilation) parameter that controls the width of the wavelet, affecting frequency resolution.

b: The translation (or shift) parameter that determines the position of the wavelet along the time axis. Wavelet Transforms decomposes a signal into components that capture both high-frequency details and low-frequency details. It can isolate different frequency bands which helps in better signal representation.

III. PROPOSED WORK

Data Preprocessing is a very important step to make data suitable and efficient for training of any Machine Learning

Model. The preprocessing steps that we have used are described below.

A. DATASET

Dataset is stored in database created Datasets hold information that captures normal and anomalous behaviors helping to train algorithms for effective threat detection in testing dataset.

NSL-KDD dataset is a refined version of the KDD Cup 1999 dataset.It represents various type of attacks shown below

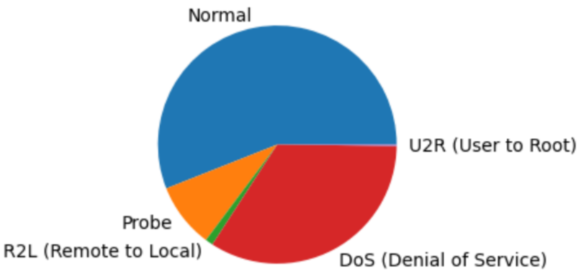


FIGURE 1. NSL-KDD Dataset

	NSL-KDD					
	Normal	**DoS**	**Prob**	**R2L**	**U2R**	**Total**
Train	67,343	45,927	11,656	995	52	**125,973**
Test	9,711	7,460	2,421	2,885	67	**22,544**

TABLE 1. Distribution of Records in NSL-KDD Dataset

WSN-DS (Wireless Sensor Network Dataset) is used in wireless sensor network (WSN) security.Dataset contains instances of common WSN attacks listed below

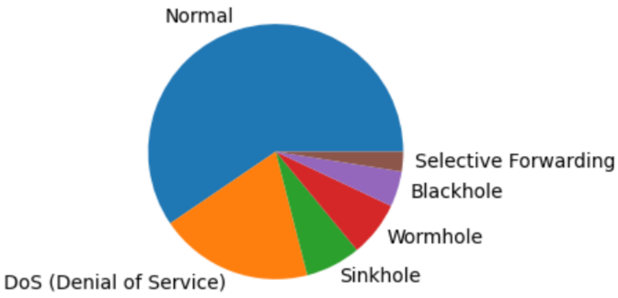


FIGURE 2. WSN-DS Dataset

WSN-DS					
Normal	**Black hole**	**Flooding**	**Gray hole**	**Scheduling**	**Total**
340,066	10,049	3,312	14,596	6,638	**374,661**

TABLE 2. Distribution of Records in WSN-DS Dataset

A dataset designed for evaluating intrusion detection systems in wireless sensor networks, focusing on attacks like Sinkhole, Blackhole, and Grayhole, with features related to node behavior and communication patterns.

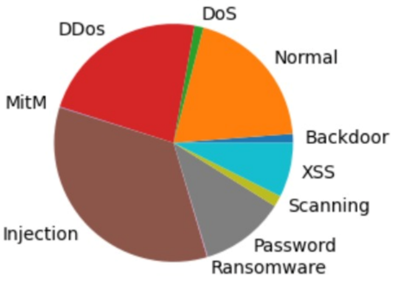


FIGURE 3. NF-TON-IOT Dataset

+depth2.5+width					NFT-TON-IOT						
+depth2.5+width	**Benign**	**Backdoor**	**DoS**	**DDoS**	**Injection**	**MITM**	**Password**	**Ransomware**	**Scanning**	**XSS**	**Total**
+depth2.5+width	270,279	17,247	17,717	345,596	468,539	1,295	156,299	142	21,467	99,944	1,398,5

TABLE 3. Distribution of Records in NFT-TON-IOT Dataset

B. DATASET GENERATION

Wasserstein Generative Adversarial Networks: Iot dataset is very rare to find in abundance. GAN consist of a generator and a discriminator that compete in a minimax game. GAN consists of two neural networks known as the generator (ζ) and the discriminator/critic (ϑ).

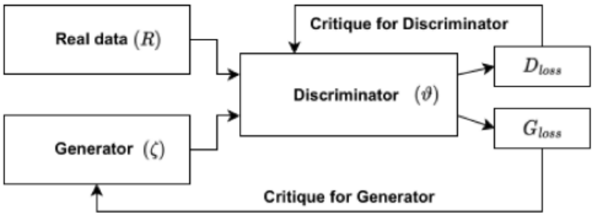


FIGURE 4. GAN

The generator aims to produce data resembling the real distribution, while the discriminator works to distinguish between real and fake samples. Wasserstein (Earth Mover's) Distance. It measures the cost of transporting one probability distribution to another.

$$L_{\text{critic}} = \mathbb{E}_{x \sim P_{\text{real}}}[f(x)] - \mathbb{E}_{z \sim P_z}[f(G(z))] \tag{1}$$

$$L_{\text{generator}} = -\mathbb{E}_{z \sim P_z}[f(G(z))] \tag{2}$$

Symbol	Description
L_{critic}	Critic loss
$L_{\text{generator}}$	Generator loss
$\mathbb{E}_{x\sim P_{\text{real}}}$	Expectation over real data samples
$\mathbb{E}_{z\sim P_z}$	Expectation over noise samples
$f(x)$	Critic function applied to real data sample
$f(G(z))$	Critic function applied to generated data
$G(z)$	Generated data sample produced G using noise z
P_{real}	Probability distribution of the real data.
P_z	Probability distribution of the noise input

TABLE 4. Descriptions of Symbols in Loss Equations

WGAN minimizes this loss function to make the real data and fake data similar. Wgan will help to increase the accuracy of model. Wasserstein Distance leads to smoother and more meaningful gradient updates, hence improving the overall quality of generated data .

C. ENCODING CATEGORICAL VARIABLES

Categorical Data (Data in string format) can not be used as a input for training of Machine Learning models, so in order to use such columns we first need to convert these columns into binary format. When working with the NSLKDD dataset the protocol_type, service and flag columns are encoded using One-Hot Encoding. After encoding all the different categories in any column now have there own seperate columns.

The working of One-Hot Encoding is very simple, lets assume a categorical feature has two categories A and B. These categories will first be given some unique indexes like A will represented by 0 and B by 1. A binary vector will then be created for each category where only the position corresponding to the category's index will be marked as 1, while all other positions are 0. For example A will be represented as [1,0] and B as [0,1]. The transformation will replace each categorical value with it's binary vector. For example if the input column is like [A,B,A] the encoded version will have three columns whose entries will be [1,0] due to first A, [0,1] due to B, [1,0] due to last A.

D. WAVELET TRANSFORMS

Wavelet Transform (WT) is the decomposition of a signal into a set of orthonormal basis functions. We take each record and convert it into a 1D signal for further transformation using wavelets.

Let a function $x(t) \in L^2(\mathbb{R})$, where $L^2(\mathbb{R})$ is the space of square integrable functions on $\mathbb{R}$. The wavelet decomposition of $x(t)$ can be written as:

$$x(t) = \sum_m \sum_n \tilde{x}_{m,n} 2^{-\frac{m}{2}} \bar{\psi}(2^{-m}t - n)$$

where $\tilde{x}_{m,n}$ are the coefficients of a discrete biorthogonal wavelet transform of $x(t)$. These coefficients are given by:

$$\tilde{x}_{m,n} = \int_{-\infty}^{\infty} 2^{-\frac{m}{2}} \psi(2^{-m}t - n) x(t)\, dt$$

The functions $\psi(t)$ and $\bar{\psi}(t)$ are called the analysis and synthesis wavelets, respectively.

We propose to use the Daubechies wavelet transform, as they are chosen to have the highest number A of vanishing moments. The wavelet family is denoted by 'dbA', where A refers to the number of vanishing moments. In our case, we choose the Daubechies wavelet with 4 vanishing moments (i.e., 'db4'). A vanishing moment limits the wavelet's ability to represent polynomial behavior or information in a signal.

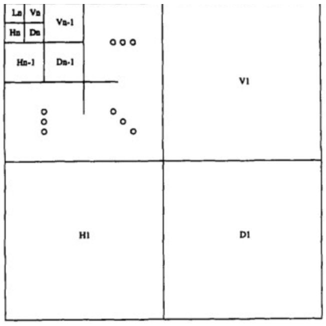


FIGURE 5. Daubechies 4 Wavelet Transform

For a record, a signal $S(t)$ can be written as:

$$S(t) = [x_1, x_2, \ldots, x_n]$$

assuming there are n features.

1) Discrete Wavelet Transform

The discrete wavelet transform (DWT) of a signal $f(t)$, using the Daubechies wavelet ('db4'), can be written as:

$$f(t) = \sum_k A_{j,k}\phi_{j,k}(t) + \sum_{j=1}^{J} \sum_k D_{j,k}\psi_{j,k}(t)$$

where:

- $A_{j,k}$ are the approximation coefficients,
- $D_{j,k}$ are the detail coefficients,
- $\phi_{j,k}(t)$ are the scaling functions (approximations),
- $\psi_{j,k}(t)$ are the wavelet functions (details).

The approximation coefficients $A_{j,k}$ represent the low-frequency (smooth) components of the signal at each level of decomposition, while the detail coefficients $D_{j,k}$ capture the high-frequency variations.

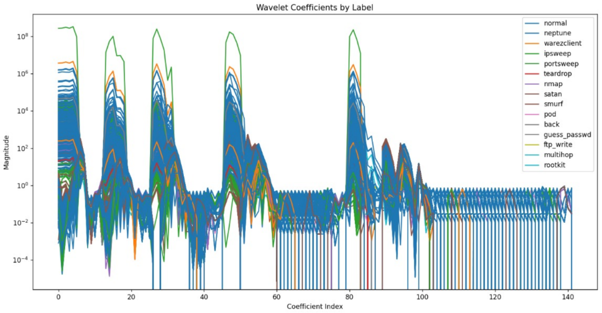


FIGURE 6. Wavelet Coefficients after preprocessing

The above figure 6 shows how wavelet coefficients look like after the preprocessing step of our proposed algorithm. The graph shows the wavelet coefficients of different type of network attacks which are categorized as normal, U2R, R2L, probe attacks etc. Each line shows the transformed wavelet representation of each network activity.

The x axis represent the sequence of wavelet coefficients, which are simply different frequency components. While moving from left to right the x axis shows how the frequency components change across the signal. The y axis shows the strength of the magnitude of each frequency component, it is in the logarithmic scale so that a wide range of magnitude (very low to very high) can be captured.

Each type of attack have its own unique frequency distribution which plays an important role for intrusion detection. From the figure it can be seen that attacks like warezclient, ipsweep, portsweep, terdrop etc. have a fixed frequency pattern which is being repeated periodically, these type of patterns can be detected by the ML model which helps in intrusion detection.

Wavelet transformation is multi-scale it can capture both low-frequency or long-term and high-frequency or short-term anomalies which can be seen in the plot above (figure 6).

E. FEATURE EXTRACTION

1) Principle Component Analysis

PCA can be effectively used to reduce dimensionality of the data to make pattern recognition easier and avoid curse of dimensionality.The principle components here are calculated in such a way that the newly transformed data should retain 95%.

PCA calculates eigenvectors and eigenvalues of covariance matrix to identify principal components, which represent the directions of maximum variance in the dataset. The original dataset is then projected onto these components, transforming it into a lower-dimensional subspace.

2) Feature Extraction After Wavelet transform

After doing Wavelet Transformation on Dataset we have extracted many statistical features from the wavelet coefficients which are explained in detail in this section.

i) **Mean**: It is average of all the wavelet coefficients, it gives a measure of central tendency of the coefficients, which indicates the level of that signal. The mean (or average) of a dataset $x_1, x_2, \ldots, x_n$ is given by:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

where:

- $\bar{x}$ is the mean,
- n is the number of data points,
- x_i represents each individual data point.

ii) **Standard Deviation (STD)**: It tells about the measure of amount of variation or dispersion in a set of given values. The STD value of any data point indicates how much is it's spread from the mean of that data. The standard deviation (STD) of a dataset $x_1, x_2, \ldots, x_n$ is given by:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

where:

- σ is the standard deviation,
- n is the number of data points,
- x_i represents each individual data point,
- $\bar{x}$ is the mean of the dataset.

iii) **Variance**: It is a measure of spread or dispersion of the dataset and is calculated as the average of the squared distances from the mean. It's the square of Standard Deviation (STD).Variance will provide information about the spread of the coefficients, which will help to understand the distribution's width. The variance of a dataset $x_1, x_2, \ldots, x_n$ is given by:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

where:

- σ^2 is the variance,
- n is the number of data points,
- x_i represents each individual data point,
- $\bar{x}$ is the mean of the dataset.

iv) **Mean Absolute Deviation (MAD)**: It is calculated as the average of the absolute difference between each wavelet coefficients and the mean. MAD can provide a better measure of variability that is less sensitive to outliers when compared to standard deviation(STD). The Mean Absolute Deviation (MAD) of a dataset $x_1, x_2, \ldots, x_n$ is given by:

$$\text{MAD} = \frac{1}{n} \sum_{i=1}^{n} |x_i - \bar{x}|$$

where:

- MAD is the Mean Absolute Deviation,
- n is the number of data points,
- x_i represents each individual data point,
- $\bar{x}$ is the mean of the dataset.

.

v) **Percentage Deviation**: It is the average percentage difference of the wavelet coefficients from there mean. It can help to understand the relative variability of the coefficients in a quite normalized manner.

The Percentage Deviation of a dataset $x_1, x_2, \ldots, x_n$ is given by:

$$\text{Percentage Deviation} = \frac{|x_i - \bar{x}|}{\bar{x}} \times 100$$

where:

- Percentage Deviation is the deviation of each data point x_i expressed as a percentage,
- x_i represents each individual data point,
- $\bar{x}$ is the mean of the dataset.

vi) **Entropy**: It is the measure of randomness in the data. It is the measure of uncertainty or the disorder within coefficients which are very important to detect anomalies. Higher entropy values will indicate more complexity and variation in the data, while lower values will indicate uniformity. The Entropy of a dataset with probability distribution $p_1, p_2, \ldots, p_n$ is given by:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

where:

- $H(X)$ is the entropy,
- p_i is the probability of occurrence of each unique value x_i in the dataset.

vii) **Skewness**: It is the measure of asymmetry of the probability distribution of the wavelet coefficients around there mean. There are three types of skewness namely:

- Left Skew: Also known as negative skew it occurs when the distribution of data is more inclined (longer) on the left side of its peak than the right side.
- Right Skew: Also known as positive skew it occurs when the distribution of data is more inclined (longer) on the right side of its peak than the left side.
- Zero Skew: This occurs when the distribution of data is symmetrical that is it's right side and left side are mirror images of each other. The Skewness of a dataset $x_1, x_2, \ldots, x_n$ is given by:

$$\text{Skewness} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{3/2}}$$

where:

  - Skewness indicates the asymmetry of the data distribution,
  - n is the number of data points,
  - x_i represents each individual data point,
  - $\bar{x}$ is the mean of the dataset.

viii) **Kurtosis**: Kurtosis measures the tailedness of a distribution of the wavelet coefficients or how often an outlier can be there in the dataset. It measures how the data disperse or spreads between a distribution center and tails. High kurtosis will indicate that the distribution has heavier tails and a sharper peak compared to a normal distribution on the other hand a low kurtosis will indicate that a distribution has lighter tails and a flatter peak when compared to a normal distribution. The Kurtosis of a dataset $x_1, x_2, \ldots, x_n$ is given by:

$$\text{Kurtosis} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{2}} - 3$$

where:

- Kurtosis indicates the tailedness of the data distribution,
- n is the number of data points,
- x_i represents each individual data point,
- $\bar{x}$ is the mean of the dataset.

F. CLASSIFICATION MODEL

CLASSIFICATION MODELS

Classification models are a subset of supervised learning algorithms designed to assign input data to predefined classes or labels based on feature patterns. These models are widely applied across various domains, including finance, healthcare, and cybersecurity, due to their ability to handle structured data and provide valuable insights.

OVERVIEW OF CLASSIFICATION ALGORITHMS

Several algorithms are commonly used for classification tasks, each with unique strengths and application scenarios:

CATBOOST (CATEGORICAL BOOSTING)

CatBoost is a gradient boosting algorithm designed for datasets with categorical features. It handles categorical variables efficiently without extensive preprocessing and can achieve high accuracy in many real-world classification problems.

ADABOOST (ADAPTIVE BOOSTING)

AdaBoost combines multiple weak classifiers to create a robust model. By assigning higher weights to misclassified samples in each iteration, AdaBoost aims to minimize errors and improve model accuracy. It is frequently used with decision trees as base learners.

MLP (MULTI-LAYER PERCEPTRON)

A type of neural network, the MLP consists of multiple layers and can capture non-linear relationships in complex datasets. However, it requires significant data and computational power and may not be ideal for small or simple datasets.

RANDOM FOREST

This ensemble technique aggregates predictions from multiple decision trees, providing a robust and accurate model. Random Forest is highly effective for large datasets with multiple features and offers the added benefit of assessing feature importance.

DECISION TREE

A non-linear model that splits data based on feature values, making it easy to interpret and visualize.

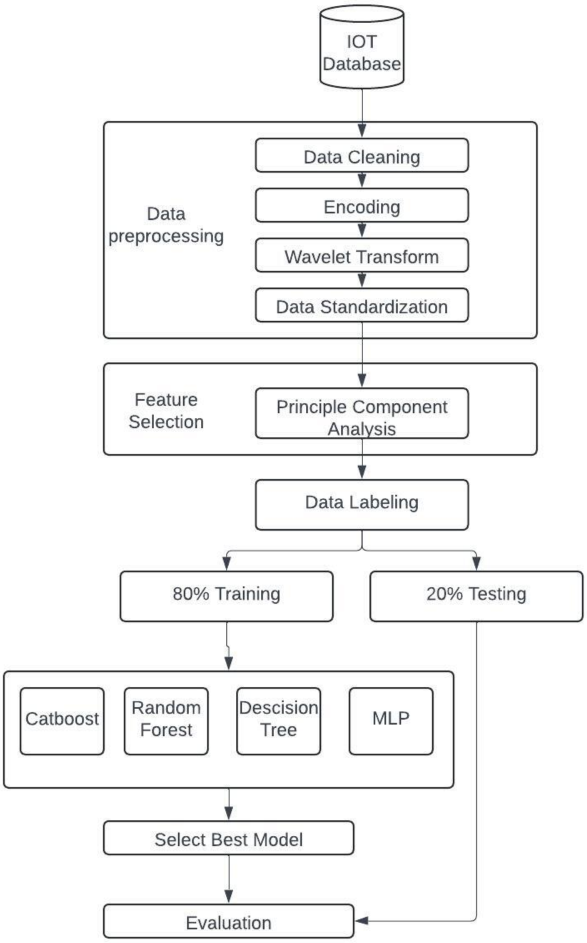


FIGURE 7. Model

G. PERFORMANCE METRICS

The following metrics are defined to evaluate the performance of the IDS solution

1) **Accuracy (ACC)**: It denotes the percentage of correctly classified flows as true attack or true legitimate flows with respect to the total number of flows. Accuracy is given by

$$\text{ACC} = \frac{TP + TN}{TP + FN + FP + TN} \times 100 \tag{3}$$

where FP = Legitimate flow wrongly classified; FN = Attack wrongly classified; TP = Attack recognized accurately; and TN = Legitimate flow recognized accurately.

2) **Precision (PREC)**: It is the percentage of correctly predicted MrDDoS attack flows out of all predicted MrDDoS attack flows. It is calculated as:

$$\text{PREC} = \frac{TP}{TP + FP} \times 100 \tag{4}$$

3) **Recall (REC)**: It indicates the percentage of predicted MrDDoS attack flows by the classifier out of all real MrDDoS attack flows of the system. Recall, also known as sensitivity, is estimated by:

$$\text{REC} = \frac{TP}{TP + FN} \times 100 \tag{5}$$

4) **F1-Score**: It indicates the overall efficiency of the proposed OPTIMIST by combining PREC and REC. It is the harmonic mean of the PREC and REC as given below:

$$\text{F1-score} = \frac{2 \times \text{PREC} \times \text{REC}}{\text{PREC} + \text{REC}} \times 100 \tag{6}$$

IV. EVALUATION AND RESULTS

The results of all the experiments that we have performed are explained in this section below. Firstly we have converted categorical columns of the dataset into binary format for giving them input to the machine learning model, then we have generated artificial data using WGAN for better training of the model, then we have applied wavelet transform on the dataset. After this we have extracted many features from the wavelet coefficients as explained in the data preprocessing section III-E1.

After feature extraction we have applied various classification models on the converted frequency domain dataset.

A. RESULTS WITH NSL-KDD DATASET

Our model aims to perform multiclass (all classes) classification with the NSL-KDD dataset, however some of the work has been done with only 5 classes of the dataset.
There are 4 major experiments which we have done with the NSL-KDD dataset which are described below

- Training and testing with combined KDDTrain and KDDTest dataset which will be referred as KDDTrain in images below.
- Training and testing with combined KDDTrain20% and KDDTest dataset which will be referred as KDDTrain20% in images below.
- Training with KDDTrain and testing with KDDTest dataset which will be referred as KDDTrain with KDDTest in images below.
- Training with KDDTrain20% and testing with KDDTest dataset which will be referred as KDDTrain20% with KDDTest in images below.

1) Accuracy

i) Accuracy Table

Dataset / Classifier	Ada Boost	Cat Boost	Random Forest	Decision Tree	Multilayer Perceptron
KDDTrain+	0.9902	0.9787	0.9911	0.9867	0.9914
KDDTrain+20_Percent	0.9791	0.9673	0.9835	0.9743	0.9795
KDDTrain+ with KDDTest+	0.7024	0.6931	0.7013	0.6972	0.7004
KDDTrain+20_Percent with KDDTest+	0.6873	0.6908	0.6929	0.6872	0.6994

TABLE 5. Comparison of Classifier Accuracy on Different Datasets

The above table 3 shows the different accuracies which we have got with various classifiers on the subsets of the NSL-KDD Datset. From the table it can be observed that AdaBoost, CatBoost and Multilayer Perceptron(MLP) give the highest accuracy when the combined KDDTrain+ and KDDTest+ data is used for both training and testing purpose.

ii) Accuracy vs Classifiers

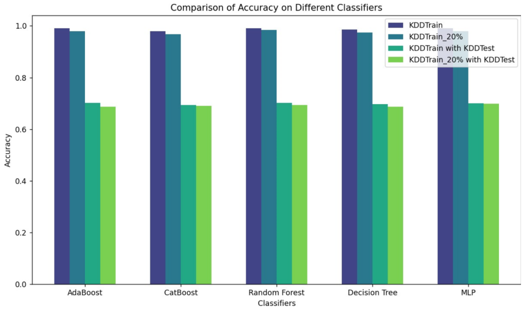


FIGURE 8. Accuracy Comparison

The above bar graph (figure 8) gives a comparison plot of the performance of various models in terms of accuracy on different subsets of the NSLKDD dataset.

2) Precision

i) Precision Table

Dataset / Classifier	Ada Boost	Cat Boost	Random Forest	Decision Tree	Multilayer Perceptron
KDDTrain+	0.98	0.99	0.99	0.99	0.98
KDDTrain+20_Percent	0.98	0.97	0.98	0.97	0.98
KDDTrain+ with KDDTest+	0.60	0.59	0.65	0.64	0.65
KDDTrain+20_Percent with KDDTest+	0.65	0.63	0.64	0.61	0.64

TABLE 6. Comparison of Classifier Precision on Different Datasets

The above table 4 depicts the different precision values which we have got after applying our model on various classifiers using the subsets of the NSLKDD Datset. From the table it can be observed that CatBoost, Random Forest and Decision Tree give the highest Precision score of 0.99 when the combined KDDTrain+ and KDDTest+ data is used for both training and testing purpose.

ii) Precision vs Classifiers

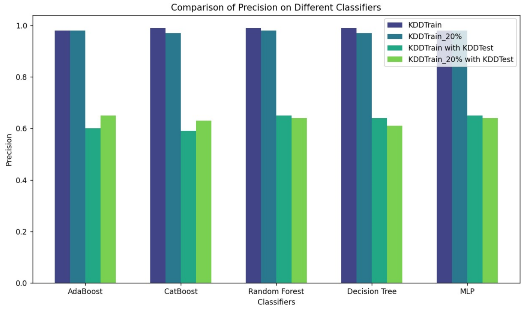


FIGURE 9. Precision Comparison

The above bar graph (figure 9) gives a comparison plot of the precision of various models on different subsets of the NSL-KDD dataset.

3) Recall

i) Recall Table

Dataset / Classifier	Ada Boost	Cat Boost	Random Forest	Decision Tree	Multilayer Perceptron
KDDTrain+	0.97	0.98	0.99	0.98	0.98
KDDTrain+20_Percent	0.98	0.97	0.98	0.97	0.98
KDDTrain+ with KDDTest+	0.69	0.72	0.69	0.70	0.70
KDDTrain+20_Percent with KDDTest+	0.73	0.70	0.71	0.69	0.70

TABLE 7. Comparison of Classifier Recall on Different Datasets

The above table 5 shows the different recall which we have got with various classifiers on the subsets of the NSLKDD Datset. From the table it can be observed all of the calssifiers have almost same recall values with the best coming from Random Forest (0.99) when the combined KDDTrain+ and KDDTest+ data is used for both training and testing purpose.

ii) Recall vs Classifiers

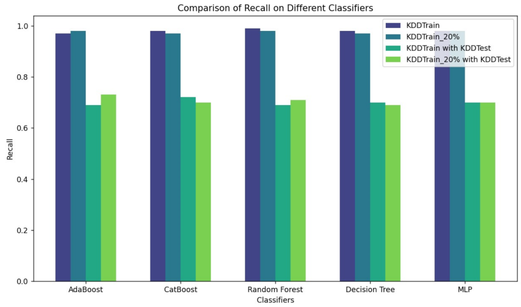


FIGURE 10. Recall Comparison

The above bar graph (figure 10) gives a comparison plot of different recall values obtained from various classifiers using different subsets of the NSLKDD dataset.

4) F1 Score

i) F1 Score Table

Dataset / Classifier	Ada Boost	Cat Boost	Random Forest	Decision Tree	Multilayer Perceptron
KDDTrain+	0.98	0.99	0.98	0.98	0.99
KDDTrain+20_Percent	0.97	0.96	0.98	0.97	0.98
KDDTrain+ with KDDTest+	0.64	0.62	0.64	0.62	0.64
KDDTrain+20_Percent with KDDTest+	0.63	0.63	0.65	0.67	0.67

TABLE 8. Comparison of Classifier F1 score on Different Datasets

The above table 6 shows the different F1 scores which we have obtained after testing our model using various classifiers on the subsets of the NSLKDD Datset. From the table it can be seen that CatBoost and Multiplayer Perceptron give the highest F1 score of 0.99 when the combined KDDTrain+ and KDDTest+ data is used for both training and testing purpose.

ii) F1 Score vs Classifiers

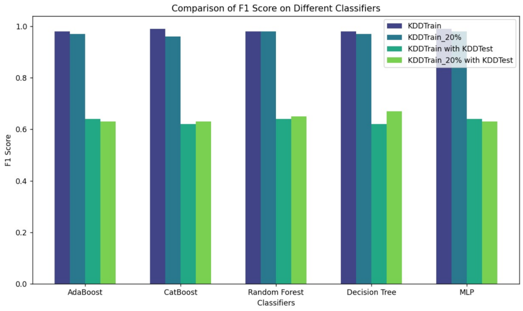


FIGURE 11. F1 Score Comparison

The above bar graph (figure 11) gives a comparison plot of the F1 scores of various models on different subsets of the NSLKDD dataset.

5) Average Metrices Comparison

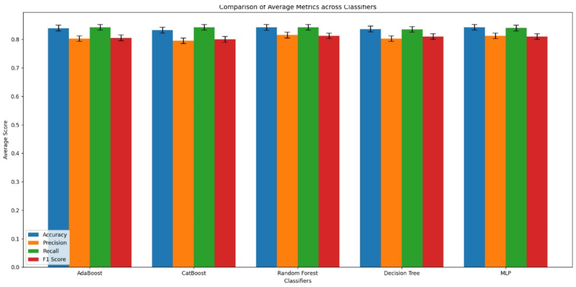


FIGURE 12. Average Metrics among all classifiers

The above bar graphs (figure 12) shows the average Accuracy, Precision, Recall, F1 Score of all the used classifiers with different subsets of the NS-LKDD Dataset. From our experiments we have observed that Random Forest and MLP were the best performers in terms of accuracy and precision, having an average accuracy of 0.8422 and 0.8426 respectively and an average precision of 0.8150 and 0.8125 respectively.

CatBoost and Random forest were the best performers in terms of average recall, both having an average recall of 0.8425. Random Forest and Decision Tree were the best performers in terms of F1 Score , having an average F1 score of 0.8125 and 0.8099 respectively.

B. RESULTS WITH WSN-DS DATASET

1) All Metrics Table

Metric / Classifier	Ada Boost	Cat Boost	Random Forest	Decision Tree	Multilayer Perceptron
Accuracy	0.9781	0.9614	0.9849	0.9779	0.9768
Precision	0.97	0.96	0.98	0.97	0.98
Recall	0.98	0.95	0.98	0.98	0.98
F1 Score	0.97	0.96	0.98	0.98	0.98

TABLE 9. Performance Metrics of Various Classifiers

The above table 7 gives a brief about the various results which we have got with the WSN-DS dataset. From the table it can be observed that the best performance was given by Random Forest in terms of accuracy which is 0.9849. On an average except CatBoost all the other classifiers have performed very well with the WSN-DS dataset.

2) Classifiers vs Accuracy and Classifiers vs Precision

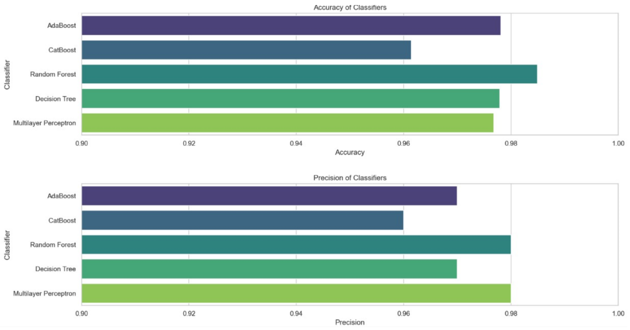


FIGURE 13. Accuracy and Precision with various classifiers

The above horizontal bar graph (figure 13) gives us a insight to the performance of various classifiers in terms of Accuracy and Precision with the WSN-DS dataset. It can be observed that best accuracy is achieved by AdaBoost and Random Forest on the other hand best precision is achieved by Multilayer Perceptron and Random Forest.

3) Classifiers vs Recall and Classifiers vs F1 Score

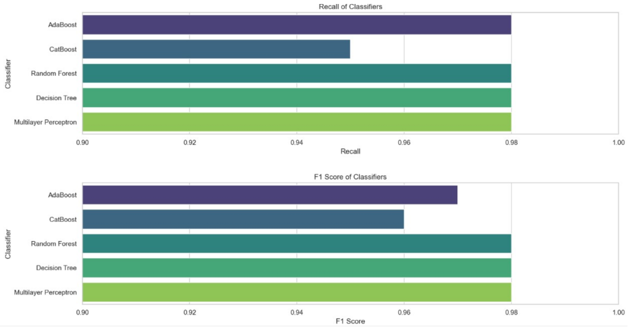


FIGURE 14. Recall and F1 Score with various classifiers

The above horizontal bar graph (figure 14) depicts to the performance of various classifiers in terms of Recall and F1 Score trained on the WSN-DS dataset. It can be seen that except CatBoost all other classifiers have got a Recall of 0.98 on the other except CatBoost and AdaBoost all the other classifiers have got a F1 Score of 0.98.

C. RESULTS WITH NF-TON-IOT DATASET

1) All Metrics Table

Metric / Classifier	Ada Boost	Cat Boost	Random Forest	Decision Tree	Multilayer Perceptron
Accuracy	0.9987	0.9854	0.9990	0.9977	0.9981
Precision	0.99	0.98	0.99	0.98	0.99
Recall	0.99	0.98	0.99	0.98	0.99
F1 Score	0.99	0.99	0.98	0.99	0.98

TABLE 10. Performance Metrics of Various Classifiers

The above table 8 tells about the results which we have got with the NF-TON-IOT dataset. From the table it can be seen that Ada Boost and Random Forest are best performers with an accuracy of 0.9987 and 0.9990 respectively. The results which we have got with NF-TON-IOT dataset are the best so far in terms of performance, but this dataset took more time in offline training when compared to others as it has around 1,157,995 rows or entries in it.

2) Classifiers vs Accuracy and Classifiers vs Precision

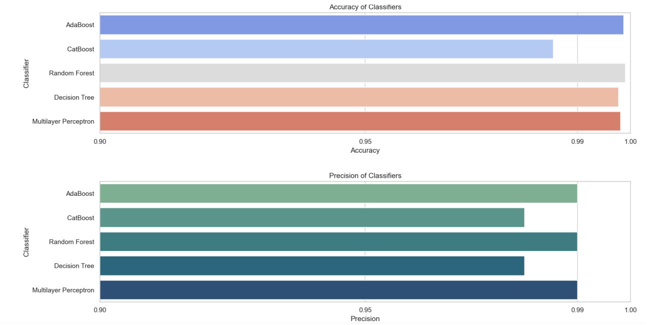


FIGURE 15. Accuracy and Precision with various classifiers

The above figure 19 gives a graphical representation of the performance of various classifiers in terms of Accuracy and Precision with the NF-TON-IOT dataset. All classifiers have performed very well but Cat Boost's performance in terms of accuracy, precision and Decision Tree's performance in terms of precision, is slightly low when compared to other classifiers.

3) Classifiers vs Recall and Classifiers vs F1 Score

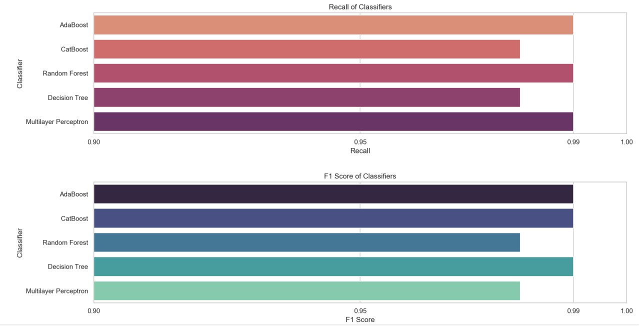


FIGURE 16. Recall and F1 Score with various classifiers

The above figure 20 gives a plot of the performance of various classifiers in terms of Recall and F1 Score with the NF-TON-IOT dataset. It can be observed that Cat Boost's, Decision Tree's performance in terms of recall and Random Forest's, Multilayer perceptron's performance in terms of F1 Score, is slightly low when compared to other classifiers.

D. PROPOSED WORK COMPARISON WITH EXISTING WORK

The table 11 below gives a comparison of the performance of existing work and our proposed model. The accuracy given in the comparison below (both existing work and our work) is the average of all the used classifiers or techniques since many of the works have implemented there model with various classifiers.

Reference and Author	Methodology	Dataset Used	Accuracy	Proposed Accuracy
BAT Deep Learning Method.. by TONGTONG SU, HUAZHI SUN	A deep learning model combining Bidirectional Long Short term memory and attention mechanism.	NSL-KDD	0.756 (With KDD test+ and 5 classes)	0.691 (With KDD test+ and all classes)
A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks.. by CHUANLONG YIN , YUEFEI ZHU.	A deep learning approach using Recurrent Neural Network	NSL-KDD	0.7648 (With KDD test+ and 5 classes) and 0.979 (With KDD train+)	0.691 (With KDD test+ and all classes) and 0.987 (With KDD train+)
Fuzziness based semi-supervised learning approach for intrusion detection system.. by Rana Aamir Raza Ashfaqa , Xi-Zhao Wanga.	A semi-supervised learning approach by utilizing unlabeled samples assisted with supervised learning algorithm	NSL-KDD	0.832 (with KDDTest+ and Binary classification)	0.838 (with KDDTest+ and Binary classification)
Deep Learning Approach for Intelligent Intrusion Detection System by R. VINAYAKUMAR , MAMOUN ALAZAB	Used Deep Neural Network(DNN) and classical Machine Learning Techniques.	NSL-KDD	0.794 (with KDDTest+ and Binary classification)	0.838 (with KDDTest+ and Binary classification)
Stochastic Gradient Descent Intrusion Detection for Wireless Sensor Network.. by HADEEL M.SALEH , HEND MAROUANE	Used Gaussian Nave Bayes (GNB) and Stochastic Gradient Descent (SGD).	WSN-DS	0.961 (Multiclass)	0.975 (Multiclass)
An Edge Computing-Based Preventive Framework With Machine Learning.. by ABDULMOHSEN ALGARNI , TAYFUN ACARER	Used Long Short-Term Memory (LSTM) and Isolation Forests (IF).	WSN-DS	0.930 (Multiclass)	0.975 (Multiclass)
Deep Learning Approach for Intelligent Intrusion Detection System by R. VINAYAKUMAR , MAMOUN ALAZAB	Used Deep Neural Network(DNN) and classical Machine Learning Techniques.	WSN-DS	0.959 (Multiclass)	0.975 (Multiclass)
Securing IoT Devices in e-Health using Machine Learning.. by Haifa Khaled Alanazi1, A.A.Abd El-Aziz2.	Simple classification using Naive Bayes (NB), Random Forest (RF), Decision Tree (DT), Artificial Neural Network (ANN), and Support Vector Machine (SVM).	NF-TON-IOT	0.877 (Binary classification)	0.995 (Binary Classification)
DIDS: A Deep Neural Network based real-time Intrusion detection system.. by Monika Vishwakarma and Nishtha Kesswani.	A deep learning model utilizing batch normalization that normalizes the contributions to a layer for every mini-batch.	NF-TON-IOT	0.991 (Binary Classification)	0.995 (Binary Classification)
Analysis and modelling of a ML-based NIDS for IoT networks by Martina Karanfilovska, Teodora Kochovska.	Used supervised learning and implemented Azure automated ML (AML) and a custom made Automated ML (AE2EML)	NF-TON-IOT	0.972 (Binary Classification)	0.995 (Binary Classification)

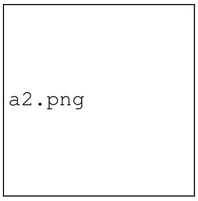
TABLE 11. Comparison of Classifier Accuracy on Different Datasets

REFERENCES AND FOOTNOTES

E. REFERENCES

FIRST A. AUTHOR (M'76–SM'81–F'87) and all authors may include biographies. Biographies are often not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author's educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state, and country, and year the degree was earned. The author's major field of study should be lower-cased.

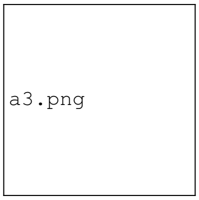
The second paragraph uses the pronoun of the person (he or she) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (publisher name, year) similar to a reference. Current and previous research interests end the paragraph. The third paragraph begins with the author's title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, it should be of good quality, and professional-looking. Following are two examples of an author's biography.



SECOND B. AUTHOR was born in Greenwich Village, New York, NY, USA in 1977. He received the B.S. and M.S. degrees in aerospace engineering from the University of Virginia, Charlottesville, in 2001 and the Ph.D. degree in mechanical engineering from Drexel University, Philadelphia, PA, in 2008.

From 2001 to 2004, he was a Research Assistant with the Princeton Plasma Physics Laboratory. Since 2009, he has been an Assistant Professor with the Mechanical Engineering Department, Texas A&M University, College Station. He is the author of three books, more than 150 articles, and more than 70 inventions. His research interests include high-pressure and high-density nonthermal plasma discharge processes and applications, microscale plasma discharges, discharges in liquids, spectroscopic diagnostics, plasma propulsion, and innovation plasma applications. He is an Associate Editor of the journal Earth, Moon, Planets, and holds two patents.

Dr. Author was a recipient of the International Association of Geomagnetism and Aeronomy Young Scientist Award for Excellence in 2008, and the IEEE Electromagnetic Compatibility Society Best Symposium Paper Award in 2011.



THIRD C. AUTHOR, JR. (M'87) received the B.S. degree in mechanical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2004 and the M.S. degree in mechanical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2006. He is currently pursuing the Ph.D. degree in mechanical engineering at Texas A&M University, College Station, TX, USA.

From 2008 to 2009, he was a Research Assistant with the Institute of Physics, Academia Sinica, Tapei, Taiwan. His research interest includes the development of surface processing and biological/medical treatment techniques using nonthermal atmospheric pressure plasmas, fundamental study of plasma sources, and fabrication of micro- or nanostructured surfaces.

Mr. Author's awards and honors include the Frew Fellowship (Australian Academy of Science), the I. I. Rabi Prize (APS), the European Frequency and Time Forum Award, the Carl Zeiss Research Award, the William F. Meggers Award and the Adolph Lomb Medal (OSA).