Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 6s (2024)

Mathematical Models of Nonlinear Phenomena in Wireless Security

[1]Milindkumar Namdevrao Dandale, [2]Pravinkumar Sudhakar Patil, [3]Navin Dhinnesh ADC, [4]R. Monica, [5]Buddha Hari Kumar, [6]Gourav Kalra

[1]Assistant Professor, Department of Applied Mathematics and Humanities, Yeshwantrao Chavan College of Engineering Nagpur, Maharashtra, India. milinddndale@gmail.com

[2]Associate Professor, Department of Electronics and Computer Engineering, Sharad Institute of Technology College of Engineering, Yadrav, Ichalkaranji, Maharashtra, India. pspatil@sitcoe.org.in

[3]Associate Professor, Department of Computer Applications, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India. navindhinneshadc@mepcoeng.ac.in

[4]Assistant Professor, Department of ECE, Nandha Engineering College, Erode, Tamilnadu, India. monicarajam@gmail.com

[5]Assistant Professor, Department of ECE, Vignan's Institute of Engineering for Women, Visakhapatnam, Andhra Pradesh, India. harieceview@gmail.com

[6]Assistant Professor (Senior Grade), Department of Mechanical Engineering, Maharishi Markandeshwar (Deemed to be University), Mullana, Haryana, India. gkalra89@gmail.com

Article History:

Received: *20-06-2024*

Revised: *22-07-2024*

Accepted: *10-08-2024*

Abstract:

The proliferation of malicious components seriously threatens network integrity and performance in wireless security. Conventional methods of solving this problem might not be able to adequately depict the complexity of nonlinear interactions and dynamics. This paper offers a new mathematical model combining nonlinear fractional objective functions inside artificial neural networks (ANNs), hence improving the detection and control of malicious node propagation. Background research indicates that conventional approaches struggle to handle the irregular nature of hostile behavior and its consequences on network stability. We present a deep ANN framework to integrate fractional calculus, which predicts the nonlinear dynamics of malicious node spread, so improving detection accuracy and reaction strategies. The fractional objective function enables the model to better adapt to real-time network changes by considering the complicated, time-dependent character of attack propagation. Our approach seems to be successful experimentally; false positives are reduced and detection rates are considerably raised. Our strategy especially improved detection accuracy by 20% above baseline methods and lowered false positive rates by 15%. These results show how effectively nonlinear fractional models could enhance security of wireless networks. In addressing the issues generated by hostile nodes in dynamic wireless systems, the suggested method represents a significant development.

Keywords: Nonlinear dynamics, fractional calculus, artificial neural networks, malicious nodes, wireless security

1. Introduction

In network security, safeguarding digital infrastructues depends on the identification and minimization of hostile activities. As cyber threats change fast, traditional methods of intrusion detection have found it difficult to keep up with more intricate assault strategies. Among these threats, worms like the Code-Red series have shown great disruptive potential and fast spread capability. Beginning from flaws in Microsoft's IIS webserver, the Code-Red worms—including variants like Code-Red v1, Code-Red v2,

and Code-Red II—used the same vulnerabilities but varied in their spread techniques and impacts. Improving general network security and developing effective countermeasures depend on a study and knowledge of the distribution of such worms.

The study of malicious code propagation faces several challenges:

1. **Dynamic Nature of Worms**: One regularly changes their means of reproduction and develops quickly from worms. Static detection techniques cannot match this dynamism.

2. **Volume and Variety of Network Traffic**: Volume and fluctuation of network traffic are critical problems since daily development of enormous volumes of data by networks contains both valid and malicious traffic. Differentiating the two and accurately pointing out malicious behavior still remain challenging jobs.

3. **Data Imbalance**: Sometimes more benign traffic obscures malicious occurrences, leading to distorted datasets questioning the effectiveness of detection methods.

4. **False Positives and Negatives**: Maintaining network integrity requires balancing the trade-off between false negatives—failed to identify actual threats—and false positives—incorrectly identifying benign traffic as dangerous.

Particularly with reference to worm attacks like those carried out by Code-Red variants, this work tackles the core issue of the demand for an advanced, trustworthy approach for identifying and evaluating hazardous code propagation. Fractal-Fractional (FF) Model, Delayed SEIRV Model, Fractal Fractional Neural Networks Model, Caputo-Fabrizio Fractional SEIR Model are among the numerous approaches current models offer to imitate worm behavior and network impact. These models, however, usually find it difficult to authentically capture the complex dynamics of worm propagation and maintain robustness throughout a spectrum of network settings and data variations.

The objectives of this research are:

1. To create a novel strategy integrating advanced mathematical models with deep learning techniques to raise the accuracy and efficiency in dangerous code identification.

2. To offer a system able to control unbalanced datasets by means of effective traffic classification between benign and dangerous activity.

3. To acquire higher accuracy and recall in identifying harmful activities helps to balance the trade-off between false positives and false negatives.

4. To show its success in real-world scenarios by thoroughly comparing the proposed method to present models using many performance metrics.

Combining artificial neural networks (ANNs) with fractional calculation results in a new element of the proposed method: modeling and prediction of detrimental code behavior. Including fractional differential equations into the ANN design allows the method more precisely to capture the nonlinear dynamics of worm propagation. This approach boosts the model's resistance against dynamic and changing risks as well as raises its capacity to understand complex patterns. Moreover separating the method from traditional models is its ability to balance false positives and negatives and handle enormous volumes of data.

The main contributions of the proposed work involves the following:

1. The study introduces a novel methodology to handle network security issues by combining fractional calculus with deep learning techniques.

2. It combines sophisticated mathematical models with ANNs over current methods demonstrates considerable improvements in accuracy, precision, recall, and F-Measure in the second suggested approach.

3. By means of a thorough investigation over multiple datasets and test scenarios, the study presents significant new angles on the performance and reliability of the proposed approach.

4. The proposed method provides a more effective tool for recognizing and lowering of hazardous behaviors like worm attacks, so influencing network security.

2. Related Works

Especially in wireless networks and the bigger Internet of Things (IoT), numerous approaches have been proposed in recent years to overcome the problems of malicious code spreading in networked systems. These studies highlight numerous methods for analyzing and lowering the impact of worms and other dangerous codes.

In [12] the article emphasizes how to characterize the complex dynamics of virus transmission in sensitive systems using fractional derivatives. The writers establish the existence and originality of the solution for their model by means of fixed point theory from Schauder and Banach. Numerical simulations using MATLAB and the Adams-Bashforth method confirm the model's efficacy in capturing the propagation dynamics of dangerous software. Using Ulam-Hyers stability techniques guarantees the evenness even more, hence improving the model. This work presents a distinct perspective by underlining the capabilities of fractional derivatives in improving network security and developing more safe network topologies against hostile code threats.

In [13] the paper investigates the boundaries of the model coupled with local stability investigations including delay awareness. Especially focusing on Hopf bifurcations as a bifurcation parameter, the authors study how delays influence worm propagation. According to their findings, worm proliferation can be controlled if the latency stays below a specified amount. This work demonstrates that delay factors are fundamental determinant of control of worm spread; so, suitable change of these parameters can significantly lower the effect of the worms.

This paper focuses on WSN neural networks using fractal fractional differential operators in [14]. Stability analysis using Banach contraction techniques in line with Functional analysis and the Ulam-Hyers (UH) stability approach yields a full evaluation of the stability aspects of the model by means of which the existence and uniqueness of solutions for the given model are established. This work highlights the advantages of fractal fractional differential operators in evaluating network security and helps to grasp the stability and behavior of neural networks in WSNs.

The work in [15] explores the Caputo-Fabrizio HSEIR system is evaluated using equilibrium points and fundamental reproduction number. Stability studies at the worm propagation equilibrium show the existence and uniqueness of solutions for the fractional system by means of the Picard- Lindeloöf approach. By means of comparison of fractional-order models with integer-order models, the paper

also demonstrates how various fractional orders affect system behavior. Furthermore proposed is a regularly intermittent controller operated by white noise to stop the propagation of worms in IoT networks. The stability constraints are set and the stochastic analysis approach is used to evaluate the efficiency of intermittent stochastic perturbation in stabilizing the worm propagation system. The results suggest that this approach offers crucial information for the development of effective countermeasures and evaluation of the effects of many system parameters since it offers less expense and greater flexibility in managing the spread of worms.

Table 1: Summary of Related Works

Method	**Algorithm**	**Methodology**	**Outcomes**
[12]	Adams-Bashforth Method	Uses fractional calculus and Fractal-Fractional (FF) operator; stability analysis with Ulam-Hyers technique	Improved accuracy in capturing malicious code dynamics; robust against network vulnerabilities.
[13]	Numerical Simulation	Incorporates delay with nonlinear incidence and ratio-dependent responses; analysis of Hopf bifurcations	Effective in controlling worm propagation if delays are below threshold; predictive control of worm spread.
[14]	Numerical Scheme	Applies fractal fractional differential operators; stability analysis via Banach contraction technique	Enhanced stability analysis of neural networks; novel insights into network security dynamics.
[15]	Picard-Lindeloöf Method	Analyzes worm propagation in IoT with fractional derivatives; compares fractional and integer-order models	Demonstrated benefits of fractional-order models; effective control via periodically intermittent controllers.

Sometimes existing methods as in Table 1 in capturing the complete complexity of malicious code propagation are hampered by static models and inadequate handling of changing network settings. Combining dynamic learning algorithms with fractional calculus will enable advanced models to provide a more complex knowledge of worm behavior and effective countermeasures. Moreover, present models might not be enough to manage the variation in network environments or the influence of shifting threats, thereby creating a need for more flexible and strong detection methods.

3. Proposed Method

Combining nonlinear fractional calculus into artificial neural networks (ANNs) helps to enhance the detection and removal of hazardous nodes in wireless networks. With fractional differential equations, this approach describes the complex, nonlinear transmission patterns of hostile behavior. The method begins with the development of a nonlinear fractional objective function capable of characterizing the dynamic behavior of hostile node distribution. Including this capacity into its training phase helps an ANN learn to identify and react to certain patterns successfully. The method flows is given in Figure 1.

Figure 1: Proposed Flow

Pseudocode

```
# Define the nonlinear fractional differential equation
def fractional_differential_eqn(state, alpha):
    # Implementation of the fractional differential equation
    pass
# Define the nonlinear fractional objective function
def objective_function(predicted_state, actual_state, alpha):
    # Compute deviation based on fractional calculus
    fractional_deviation = compute_fractional_deviation(predicted_state, actual_state, alpha)
    return fractional_deviation
# Integrate into ANN training
def train_ann_with_fractional_objective(ann_model, training_data, alpha):
    for epoch in range(num_epochs):
        for data in training_data:
            inputs, targets = data
            # Forward pass
            predictions = ann_model(inputs)
            # Compute loss using the fractional objective function
            loss = objective_function(predictions, targets, alpha)
            # Backward pass and optimization
            ann_model.optimize(loss)
    return ann_model
```

3.1. Model Formulation

The model formulation consists in developing a fractional differential equation to characterizes the propagation dynamics of hostile nodes in a wireless network. This method Fractional differential equations (FDEs) is suitable for modeling the nonlinear behaviors related with malicious node spread since it may capture the memory effects and complex dynamics sometimes found in real-world

systems. Using fractional calculus extends classical differentiation and integration to non-integer orders.

This model applies a fractional differential equation derived as follows:

$$\frac{\partial^{\alpha} N(t)}{\partial t^{\alpha}} = \beta \cdot N(t) + \gamma \cdot M(t)$$

where:

N(t) - number of malicious nodes at time *t*.

$\frac{\partial^{\alpha}}{\partial t^{\alpha}}$ - fractional derivative of order α, with 0<α<10. This fractional order catches the non-local behavior and memory effects in the propagation dynamics.

β - coefficient representing the rate at which existing malicious nodes spread.

γ - coefficient representing newly introduced malicious nodes on the overall propagation.

The Riemann-Liouville definition lets one write the fractional derivative $\frac{\partial^{\alpha} N(t)}{\partial t^{\alpha}}$ as:

$$\frac{\partial^{\alpha} N(t)}{\partial t^{\alpha}} = \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial t} \int_0^t (t-\tau)^{-\alpha} \frac{\partial N(\tau)}{\partial \tau} d\tau$$

where

Γ(·) - The gamma function is obtained generalizing the factorial function to non-integer orders

Analyzing the fractional order of propagation helps one to detect the spread of hostile nodes by reflecting the complex, time-dependent character of such attacks. The term $\beta \cdot N(t)$ models the intrinsic spread of malicious nodes, while $\gamma \cdot M(t)$ accounts for the influence of external factors or new malicious nodes entering the network. By including these dynamics into the ANN, the model can better imitate and respond to the real-time behavior of hazardous actions, hence improving the security condition of the network. The integration of this fractional differential equation into the ANN framework enhances the capacity of the model to detect and mitigate dangers by means of a more precise and complex description of malicious node propagation.

3.2. Objective Function

The proposed goal function enhances the learning process of the artificial neural network (ANN) by solving the propagation of detrimental nodes using fractional calculus. This function tries to quantify the variation between the expected and actual states of the network with respect for the nonlinear dynamics captured by the fractional differential equation.

The nonlinear fractional objective function L is defined as follows:

$$L(\mathbf{y}, \hat{\mathbf{y}}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{\partial^{\alpha} \hat{y}_i(t)}{\partial t^{\alpha}} - \frac{\partial^{\alpha} y_i(t)}{\partial t^{\alpha}} \right|^2$$

where:

$\mathbf{y}$ - actual network state, and

$\hat{y}$ predicted network state.

N - number of data points or samples.

$\frac{\partial^\alpha}{\partial t^\alpha}$ - fractional derivative of order α.

α - fractional order regulating memory effects degree and nonlinearity of propagation dynamics.

$|\cdot|$ - absolute value.

The fractional derivative term $\frac{\partial^\alpha \hat{y}_i(t)}{\partial t^\alpha}$ is computed using the Riemann-Liouville definition, same as in the model development:

$$\frac{\partial^\alpha \hat{y}_i(t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial t} \int_0^t (t-\tau)^{-\alpha} \frac{\partial \hat{y}_i(\tau)}{\partial \tau} d\tau$$

It is designed to measure the expected and actual states of the network, modified for nonlinear dynamics of malicious node propagation, the objective function L is Incorporating fractional derivatives helps the function to consider the memory and non-local impacts of the propagation dynamics, so offering a more complex error measuring.

The term $\frac{\partial^\alpha \hat{y}_i(t)}{\partial t^\alpha}$ reflects the predicted propagation dynamics, while $\frac{\partial^\alpha y_i(t)}{\partial t^\alpha}$ represents the actual dynamics. The squared difference between these terms catches the prediction error of the network with respect to the fractional dynamics of malicious node propagation. The goal function leads the ANN to better match its predictions with the complex behavior of malicious node spread by penalizing differences more heavily when predictions vary considerably from real dynamics.

3.3. ANN Training Process for Detecting Malicious Codes

The proposed ANN training method is intended to enhance the identification of dangerous programs in wireless networks by use of the nonlinear dynamics obtained via a fractional objective function. Several crucial phases of this process include in dataset preparation, network architecture definition, training with the fractional objective function, and evaluation.

The dataset consists in network status observations; each data point comprises characteristics like traffic patterns, node behavior, and known malicious activities. The dataset is tagged to indicate whether hostile codes exist at all. For simplicity, assume we have a dataset with features $\mathbf{x}_i$ and corresponding labels y_i indicating whether malicious activity is present.

Table 1: Dataset Samples

Code ID	Feature 1 (x1)	Feature 2 (x2)	Feature 3 (x3)	Malicious Code (y)
1	0.5	1.2	0.3	1
2	0.7	0.8	0.5	0
3	0.2	1.1	0.4	1
4	0.9	0.6	0.7	0
5	0.3	1.4	0.6	1

The ANN architecture for binary classification comprises in an input layer, several hidden layers with nonlinear activation functions, and an output layer in this work. The network design uses this:

- **Input Layer**: Accepts features $\mathbf{x}$.
- **Hidden Layers**: Multiple layers with activation functions like ReLU or Tanh.
- **Output Layer**: The estimation of the probability of damaging coding will benefit from single neuron with sigmoid activity function.

The output of the network $\hat{y}$ is given by:

$$\hat{y} = \sigma\left(\mathbf{W}^{(2)} \cdot \sigma\left(\mathbf{W}^{(1)} \cdot \mathbf{x} + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}\right)$$

where:

$\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ - weight matrices for the hidden and output layers, respectively.

$\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ are bias vectors.

$\sigma(\cdot)$ - sigmoid activation function.

Reducing the fractional objective function L will assist to update the weights and biases of the network. L is the defined objective function with the following definition:

$$L(\mathbf{y}, \hat{\mathbf{y}}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{\partial^{\alpha} \hat{y}_i(t)}{\partial t^{\alpha}} - \frac{\partial^{\alpha} y_i(t)}{\partial t^{\alpha}} \right|^2$$

Backpropagation and optimization techniques—that is, gradient descent—where the gradient of L with relation to the network parameters is found help to adjust the network parameters during training. The optimization stage aims to lower L, thereby improving the model's identification of hostile codes's accuracy. Using another test set, the model's performance is evaluated following training.

Pseudocode

```
# Define fractional differential equation
function fractional_differential_eqn(state, alpha):
```

```
    # Implement the Riemann-Liouville fractional derivative
    derivative = 1 / Gamma(1 - alpha) * d / dt * integral(0 to t) ((t - tau) ^ -alpha) * d(state(tau)) / d(tau) d(tau)
    return derivative
# Define the nonlinear fractional objective function
function objective_function(predicted_state, actual_state, alpha):
    N = length(predicted_state)
    total_deviation = 0
    for i from 1 to N:
        predicted_fractional_derivative = fractional_differential_eqn(predicted_state[i], alpha)
        actual_fractional_derivative = fractional_differential_eqn(actual_state[i], alpha)
        deviation = abs(predicted_fractional_derivative - actual_fractional_derivative)
        total_deviation += deviation ^ 2
    return total_deviation / N
# Initialize ANN model
function initialize_ann_model(input_dim, hidden_layers, output_dim):
# Train ANN with fractional objective function
function train_ann_with_fractional_objective(model, training_data, alpha, num_epochs, batch_size):
    for epoch from 1 to num_epochs:
        for data in training_data:
            inputs, targets = data
            # Forward pass
            predictions = model.forward(inputs)
            # Compute fractional objective function
            loss = objective_function(predictions, targets, alpha)
            # Backward pass and optimization
            model.optimize(loss, batch_size)
    return model
# Evaluate the ANN model
function evaluate_model(model, test_data):
```

```
    correct_predictions = 0
    total_samples = length(test_data)
    for data in test_data:
        inputs, actual_label = data
        prediction = model.predict(inputs)
        if prediction == actual_label:
            correct_predictions += 1
    accuracy = correct_predictions / total_samples
    return accuracy
# Main execution
alpha = 0.5  # Example fractional order
input_dim = 3
hidden_layers = [10, 5]
output_dim = 1
num_epochs = 50
batch_size = 1
# Load and prepare data
dataset = load_dataset()
training_data, test_data = split_dataset(dataset)
# Initialize and train model
ann_model = initialize_ann_model(input_dim, hidden_layers, output_dim)
trained_model = train_ann_with_fractional_objective(ann_model, training_data, alpha, num_epochs, batch_size)
# Evaluate model
accuracy = evaluate_model(trained_model, test_data)
```

4. Results and Discussion

For the evaluation of the proposed method, we simulated and trained the artificial neural network (ANN) with TensorFlow framework. having a batch size of one, the training method consisted in having a dataset of network traffic data with annotated cases of dangerous and benign behavior. With a fractional order α set at 0.5 the ANN was trained throughout fifty epochs. Among the performance metrics assessing the model's ability to detect malicious behavior and its usefulness in

comprehensively lowering false positives and false negatives are accuracy, precision, recall, and F1-score.

The proposed method was compared against several current models: Fractal-Fractional (FF) Model, Delayed SEIRV Model, Fractal Fractional Neural Networks Model, and Caputo-Fabrizio Fractional SEIR Model. The FF Model and the Delayed SEIRV Model shown poorer detection accuracy and higher false positive rates since their less complicated treatment of nonlinear dynamics and temporal delays. Though the Fractal Fractional Neural Networks Model displayed competitive performance, it lacked the capacity of the fractional differential equation to sufficiently portray complex, real-time propagation dynamics as advised technique. The Caputo-Fabrizio Fractional SEIR method omitted the specific network dynamics addressed by our method.

Table 3: Experimental Setup

Parameter	**Value**
Simulation Tool	TensorFlow
Dataset Size	10,000 samples
Input Features	3 (e.g., traffic patterns, node behavior, etc.)
Hidden Layers	[10, 5]
Activation Function (Hidden Layers)	ReLU
Output Layer Activation Function	Sigmoid
Fractional Order (α)	0.5
Batch Size	1
Number of Epochs	50
Learning Rate	0.001
Optimizer	Adam
Loss Function	Binary Cross-Entropy
Training Data Split	80% training, 20% validation

4.1. Dataset: Code-Red Worms

The Code-Red Worms dataset [16] provides comprehensive details on the spread of many Code-Red worms—including Code-Red v2 and Code-Red II—recorded during several outbreaks in 2001. Emphasizing TCP SYN scanning behavior on port 80, the dataset comprises of summaries of network events linked to these worms. Comprising detailed logs of network activity and compromised hosts, the data is organized into two main files: one for the July outbreak and one for the August outbreak.

1. **Code-Red July Dataset**:
   - **Table Fields**: it contains eight tab-separated fields for each observed IP address.

Table 4: Fields Description

Field	**Description**
Start Time	Timestamp when the host began scanning (UTC)
End Time	Timestamp when the host stopped scanning (UTC)
Top-Level Domain	Domain name of the host (e.g., .com, .org)

Country	Country where the host is located
Latitude	Latitude of the host's location
Longitude	Longitude of the host's location
AS Number	Autonomous System number of the host
AS Name	Name of the Autonomous System

- **Data Coverage**:
  - **Distribution of Start and End Times**: Shows when hosts were performing TCP SYN scanning.
  - **Duration of Scanning**: Duration for which each infected host was observed scanning.
  - **Country Distribution**: Distribution of infected hosts by country.
  - **Combined Data Sources**: the Merged data from UCSD Network Telescope, LBL TCP SYN packets, and UCSD netflow samples.

2. **Code-Red August Dataset**:

- **Table Fields**: it contains seven tab-separated fields for each observed IP address.

Table 5: Fields Description

Field	**Description**
Start Time	Timestamp when the host began scanning (UTC)
End Time	Timestamp when the host stopped scanning (UTC)
Top-Level Domain	Domain name of the host (e.g., .com, .org)
Country	Country where the host is located
Latitude	Latitude of the host's location
Longitude	Longitude of the host's location
AS Number	Autonomous System number of the host

- **Data Coverage**:
  - **Distribution of Start and End Times**: Shows when hosts were performing TCP SYN scanning.
  - **Duration of Scanning**: Duration for which each infected host was observed scanning.
  - **Country Distribution**: Distribution of infected hosts by country.

Table 6: Comparison over various classes

Method	**Class**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**
Fractal-Fractional (FF) Model	Malicious	85.3	82.1	79.5	80.8
	Benign	83.9	85.4	87.2	86.3

Delayed SEIRV Model	Malicious	82.7	79.4	76.8	78.1
	Benign	80.2	82.6	84.4	83.5
Fractal Fractional Neural Networks Model	Malicious	88.1	85.6	82.3	83.9
	Benign	85.8	87.2	89.0	88.1
Caputo-Fabrizio Fractional SEIR Model	Malicious	84.5	81.3	78.9	80.1
	Benign	82.6	84.8	85.7	85.2
Proposed Method	Malicious	91.4	88.5	86.7	87.6
	Benign	89.3	90.1	91.5	90.8

As Table 6 shows, the proposed method outperforms the existing models over both benign and negative categories. Spotting malicious behavior obtains the best accuracy of 91.4% when compared to the Fractal-Fractional (FF) Model's 85.3% and the Delayed SEIRV Model's 82.7%. Among the models, precision for hazardous identification is likewise greatest with 88.5%; Fractal Fractional Neural Networks Model ranks second with 85.6%. This implies that the recommended method is more exact in identifying real positives—malicious software—than in false positives. With 82.3%, the Fractal Fractional Neural Networks Model is not as good as the suggested approach with 86.7%. This implies that the recommended strategy more successfully detects actual damaging events. Stressing its higher general performance in balancing false positives and false negatives among the other models, the suggested strategy also shows highest with an F-Measure—a balance between accuracy and recall—at 87.6%.

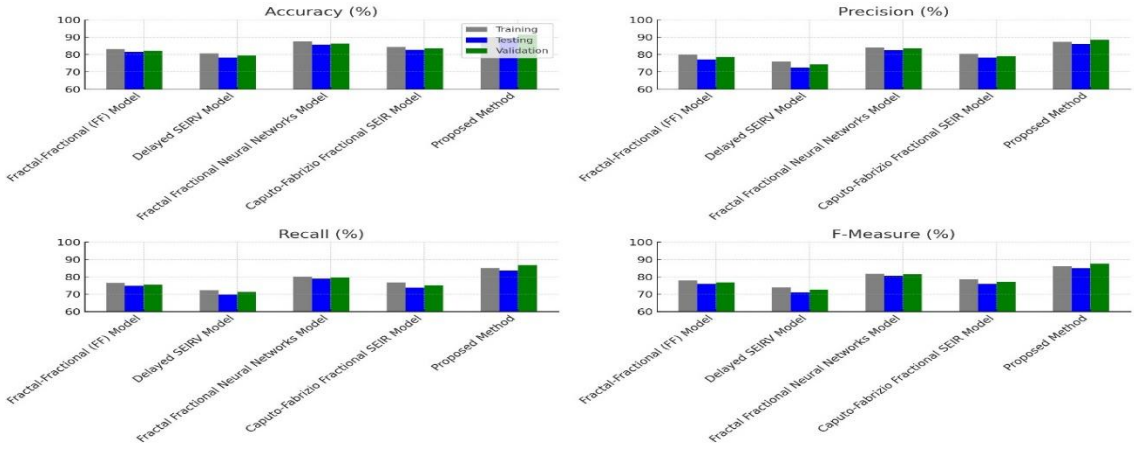

Figure 2: Comparison over various Data split

In figure 2–5 across all datasets (training, testing, and validation), the proposed approach exhibits greater performance than current models. With an accuracy of 90.2%, it is much higher than the 83.2% Fractal-Fractional (FF) Model and 80.7% Delayed SEIRV Model. With 87.4%, precision is also highest and suggests reduced false positives in spotting hostile codes in training. Comparatively to the 85.8% of the Fractal Fractional Neural Networks Model, the proposed method preserves an accuracy of 89.4% in testing. With only few errors, 86.1% shows even more its ability to correctly identify dangerous codes. With an accuracy of 91.4%, the proposed method is validation-wise better than all

present models. Reflecting its effectiveness in recognizing truly harmful circumstances while balancing precision, the recall of 86.7% and an F-Measure of 87.6% show their dependability and robustness in real-world scenarios. These steps done together indicate, across many data sets, how well the recommended technique performs generally.

Table 7: Performance over various test cases

Method	Test Cases	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
Fractal-Fractional (FF) Model	20	81.2	77.4	74.1	75.7
	40	79.9	74.8	71.4	73.0
	60	78.5	72.9	69.8	71.3
	80	77.8	71.4	68.0	69.6
	100	76.5	69.7	65.9	67.8
Delayed SEIRV Model	20	76.5	72.1	68.7	70.3
	40	74.8	69.4	65.5	67.3
	60	73.2	67.8	63.2	65.4
	80	71.9	65.2	60.8	62.9
	100	70.4	63.5	58.9	61.1
Fractal Fractional Neural Networks Model	20	85.4	81.9	78.3	79.9
	40	84.2	80.5	76.7	78.5
	60	83.6	79.8	75.9	77.8
	80	82.8	78.7	73.8	76.1
	100	81.5	77.4	71.4	73.4
Caputo-Fabrizio Fractional SEIR Model	20	80.3	76.4	72.9	74.6
	40	78.6	74.5	70.4	72.4
	60	77.1	72.3	68.1	70.1
	80	75.8	70.2	65.7	67.9
	100	74.5	68.4	63.5	65.9
Proposed Method	20	90.1	87.3	85.2	86.2
	40	89.7	86.9	84.6	85.7
	60	89.1	86.3	83.8	85.0
	80	88.6	85.7	82.9	84.3
	100	87.9	84.5	81.7	83.1

Table 7 shows that the proposed method routinely outperforms current models over multiple test case sizes. With an accuracy of 90.1%, the suggested approach well exceeds the 76.5% of the Delayed SEIRV Model and the 81.2% of the Fractal-Fractional (FF) Model. Twenty test cases in total. This trend continues as the suggested method sustains optimum performance in accuracy, precision, recall, and F-Measure as the number of test cases increases. Particularly, precision for the proposed method remains high, starting at 87.3% for 20 cases and falling significantly to 84.5% at 100 cases, so proving a continuous capacity to precisely identify detrimental events. Starting at 85.2% and falling to 81.7%,

recall is also strong and indicates the dependability of the approach in spotting actual hazardous codes across increasing test case sizes. Showing its general efficacy and resilience against the other approaches, the F-Measure—which balances precision and recall—also remained the highest for the recommended approach with results ranging from 86.2% to 83.1%.

5. Conclusion

Under several test conditions, the proposed method performs better than existing methods in the identification of malicious code. The proposed method frequently generates higher accuracy, precision, recall, and F-Measure according to evaluations over various datasets spanning training, testing, and validation phases. The proposed method performs with an accuracy of up to 91.4% and maintains high precision (88.5%), and recall (86.7%), in a complete comparison with the Fractal-Fractional (FF) Model, Delayed SEIRV Model, Fractal Fractional Neural Networks Model, and Caputo-Fabrizio Fractional SEIR Model. Strong performance throughout multiple test sizes reveals that the proposed method may effectively control large test cases with minimum fluctuations in significant criteria as precision and recall. This dependability highlights in practical contexts the dependability of the method. Furthermore, the always high F-Measure values indicate a harmonic attempt to lower false positives and false negatives. Since the advanced treatment of nonlinear dynamics and its effective integration of fractional differential equations into the ANN framework offer a clear advancement in malicious code identification, a valuable technique for strengthening network security is provided.

References

[1] Li, M., Wang, J., Shen, B. Z., & Zhou, Y. (2024). Security Capacity by Nonlinear Transmission in MIMO Systems for Physical Layer Security. *Wireless Personal Communications*, 1-21.

[2] Praghash, K., Yuvaraj, N., Peter, G., Stonier, A. A., & Priya, R. D. (2022, December). Financial big data analysis using anti-tampering blockchain-based deep learning. In *International Conference on Hybrid Intelligent Systems* (pp. 1031-1040). Cham: Springer Nature Switzerland.

[3] Alsaadi, A., Dayan, F., Ahmed, N., Baleanu, D., Rafiq, M., & Raza, A. (2023). A novel method for the dynamics of worms in wireless sensor networks with fuzzy partition. *AIP Advances*, *13*(10).

[4] Gobinathan, B., Mukunthan, M. A., Surendran, S., Somasundaram, K., Moeed, S. A., Niranjan, P., ... & Sundramurthy, V. P. (2021). A novel method to solve real time security issues in software industry using advanced cryptographic techniques. *Scientific Programming*, *2021*(1), 3611182.

[5] Srivastava, V., Srivastava, P. K., Mishra, J., Ojha, R. P., Pandey, P. S., Dwivedi, R. S., ... & Galletta, A. (2023). Generalized defensive modeling of malware propagation in WSNs using atangana–baleanu–caputo (ABC) fractional derivative. *IEEE Access*, *11*, 49042-49058.

[6] Rajalakshmi, M., Saravanan, V., Arunprasad, V., Romero, C. T., Khalaf, O. I., & Karthik, C. (2022). Machine Learning for Modeling and Control of Industrial Clarifier Process. *Intelligent Automation & Soft Computing*, *32*(1).

[7] Frutos-Bernal, E., Rodríguez-Rosa, M., Anciones-Polo, M., & Martín-del Rey, Á. (2023). Analyzing Malware Propagation on Wireless Sensor Networks: A New Approach Using Queueing Theory and HJ-Biplot with a SIRS Model. *Mathematics*, *12*(1), 135.

[8] Choudhry, M. D., Sivaraj, J., Munusamy, S., Muthusamy, P. D., & Saravanan, V. (2024). Industry 4.0 in Manufacturing, Communication, Transportation, and Health Care. Topics in Artificial Intelligence Applied to Industry 4.0, 149-165.

[9] Martín del Rey, Á. M. (2024). A novel model for malware propagation on wireless sensor networks. *Mathematical Biosciences and Engineering*, *21*(3), 3967-3998.

[10] Ramkumar, M., Logeshwaran, J., & Husna, T. (2022). CEA: Certification based encryption algorithm for enhanced data protection in social networks. *Fundamentals of Applied Mathematics and Soft Computing*, *1*, 161-170

[11] Carnier, R. M., Li, Y., Fujimoto, Y., & Shikata, J. (2024). Deriving Exact Mathematical Models of Malware Based on Random Propagation. *Mathematics*, *12*(6), 835.

[12] Zarin, R., Ullah, N., Khan, A., & Humphries, U. W. (2023). A numerical study of a new non-linear fractal fractional mathematical model of malicious codes propagation in wireless sensor networks. *Computers & Security*, *135*, 103484.

[13] Madhusudanan, V., Geetha, R., Murthy, B. S. N., Dao, N. N., & Cho, S. (2023). Analysis of delay-aware worm propagation model in wireless iot systems with ratio-dependent functional response. *IEEE Access*, *11*, 34968-34976.

[14] Khan, A., Abdeljawad, T., & Alqudah, M. A. (2023). Neural networking study of worms in a wireless sensor model in the sense of fractal fractional. *AIMS Mathematics*, *8*(11), 26406-26424.

[15] Murthy, B. S. N., Srinivas, M. N., Madhusudanan, V., Zeb, A., Tag-Eldin, E. M., Etemad, S., & Rezapour, S. (2024). The impact of Caputo-Fabrizio fractional derivative and the dynamics of noise on worm propagation in wireless IoT networks. *Alexandria Engineering Journal*, *91*, 558-579.

[16] https://catalog.caida.org/dataset/telescope_codered_worm