IIOT Prediction Accuracy Improved By T-Distributed Stochastic Neighborem Bedding and Jaguar Algorithm With Non-Linear Analysis

M. Jeyaselvi[1], V. Vijayagopal[2], Pravin Prakash Adivarekar[3], Swati Kedar Nadgaundi[4], V S Narayana Tinnaluri[5], Gourav Kalra[6]

[1]Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, Chennai, India, Email: jeyaselm@srmist.edu.in
[2]Assistant Professor, Faculty of Computer Applications, Dr. M.G.R. Educational and Research Institute, Maduravoyal, Chennai, Tamilnadu, India, Email: vijayagopal1976@gmail.com
[3]Associate Professor, Data Science Department, A. P. Shah Institute of Technology, Thane, Maharashtra, India, Email: ppadivarekar@apsit.edu.in
[4]Assistant Professor, Department of Instrumentation, Bharati Vidyapeeth College of Engineering Navi Mumbai, Maharashtra, India, Email: swati.nadgaundi@bvcoenm.edu.in
[5]Associate Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India, Email: vstnarayana@gmail.com
[6]Assistant Professor, Department of Mechanical Engineering, Maharishi Markandeshwar Engineering College, Mullana Ambala Haryana, India, Email: gkalra89@gmail.com

Received: 29.04.2024 Revised : 13.05.2024 Accepted: 21.05.2024

ABSTRACT

In the fast changing terrain of Industrial Internet of Things (IIoT), predictive accuracy is very vital to maximize operations, reduce downtime, and increase process optimization. Typical prediction models, on high-dimensional data, however, may fail, leading to less than perfect accuracy and performance. Together with non-linear analysis, this paper examines the mix of T-Distributed Stochastic Neighbour Embedding (t-SNE) and the Jaguar approach to increase prediction accuracy in IIoT systems so addressing this problem. Managing complex, non-linear interactions present in IIoT data presents the main difficulty. While conventional methods may fail in the complicated patterns prevalent to IIoT systems, they operate well in linear conditions. This paper proposes a new approach whereby t-SNE is employed for dimensionality reduction, therefore retaining the local structure of data and so minimizing the curse of dimensionality. Then the prediction model is refined using a nature-inspired optimization technique called the Jaguar algorithm, hence improving its flexibility in the face of data non-linearities. The proposed method was evaluated on a comprehensive IIoT dataset and shown to yield appreciable expected accuracy gain. The model especially outperformed traditional methods such Genetic Algorithms (GA) and Principal Component Analysis (PCA), which generated accuracies of 87.3% and 89.1%, respectively, with an accuracy of 94.8%. Moreover proving its ability to handle complex IIoT data, the proposed approach showed a 10.3% increase in precision and a 12.5% increase in recall.

Keywords: IIoT,t-SNE, Jaguar algorithm, non-linear analysis, prediction accuracy

1. INTRODUCTION

The Industrial Internet of Things (IIoT) signals a radical change in manufacturing and industrial operations as it combines modern sensors, connectivity, and data analytics to increase operational efficiency and decision-making [1,2]. In IIoT systems for equipment failure prediction, production process optimization, and general system dependability augmentation, predictive analytics is critically indispensable [3]. Mostly of the time, the intricacy and non-linearities of large-scale industrial data confound conventional forecasting methods [4].

Among the primary challenges in IIoT prediction tasks is controlling the high-dimensional and dynamic character of the data [5]. From sensors, actuators, and control systems among other sources, industrial systems produce vast amounts of data [6]. Many times noisy, missing, and displaying intricate non-linear correlations is this data [7]. Conventional predictive models—such as simple neural networks or linear regression—may not be able to sufficiently capture these complexity, thereby generating less than ideal outcomes [8]. Moreover, optimizing these models requires sophisticated approaches that balance computational economy with accuracy, hence lowering overfitting and underfitting issues [9].

By means of overcoming present methodologies, predictive models for IIoT systems will become more accurate and efficient, so addressing the basic problem of this research [10]. The aim is to develop a new strategy able to control non-linear data interactions and achieve greater predictive performance than standard approaches. Especially the focus is on combining dimensionality reduction, advanced optimization methodologies, and non-linear analysis to raise model correctness and durability [11].

The objectives of the proposed work include:

- To create a model able to effectively show the complex, non-linear relationships in IIoT data.
- To carefully change model parameters for maximum expected accuracy using modern optimizing techniques
- To develop a model that operates consistently over several data sets—training, testing, validation included.
- By means of a comparison between the recommended technique and accepted procedures, to demonstrate its improved performance and practical advantages.

Combining several unique features, the proposed approach solves the challenges of IIoT prediction jobs. First it effectively reduces advanced dimensionality by employing T-Distributed Stochastic Neighbour Embedding (t-SNE), so capturing the non-linear structure of high-dimensional data. Second it provides the Jaguar Algorithm for optimisation, a novel approach combining exploration and exploitation strategies to raise model correctness and efficiency. Thirdly, the method considers regularization to avoid over fitting and complex data interactions using a non-linear objective function.

The contributions of the proposed work include:

- By use of t-SNE, the proposed method greatly reduces the dimensionality of complex IIoT data while preserving crucial non-linear correlations, hence improving the model performance.
- Using the Jaguar Algorithm provides a strong optimization structure that increases the capacity of the model to recognize optimal solutions in complex parameter ranges.
- More accurate evaluation of model performance made feasible by the mix of non-linear objective functions produces better predicted accuracy.

2. RELATED WORKS

Predictive analytics is essential in modern industrial settings if we are to improve operational efficiency and lower downtime by way of maintenance process optimization. One sensible strategy is to mix Industrial Internet of Things (IIoT) with Machine Learning (ML) techniques. This combination uses real-time data from IIoT devices to predict equipment failures and optimizes maintenance schedules. Four distinct ML models for predictive maintenance are used in [12] to evaluate this mix. Emphasizing the significance of selecting the appropriate model, the research stresses the need of accurate and consistent forecasts to reduce costly downtime and raise equipment lifetime. By means of a thorough analysis of several models, the research underlines the importance of model selection and provides a foundation for next advancements in predictive maintenance approaches in commercial settings.

Energy efficiency is a critical concern in IIoT systems particularly for sensor nodes functioning continuously in industrial environments. Aiming for reduced data transfer to reduce energy consumption, current study [13] presents an energy-efficient IIoT architecture. Suggesting a deep learning model, DC-MLP, the work uses a deep concatenation technique to increase prediction performance and efficiency. The study reveals a 33% drop in energy consumption when using six performance criteria and changing the sample rate when compared to standard data transmission technologies. With an 81% faster prediction time as well, the DC-MLP model demonstrates how effectively it blends energy economy with consistent data forecasts. This approach addresses the twin challenges of maximizing energy use in Internet of Things devices and keeping great forecast accuracy.

In IIoT systems, security is quite important considering the possible risks of cyber-attacks. The main focus of a research is developing high-accuracy intrusion detection techniques for IIoT systems [14]. The research shows an incredible accuracy of 100% confirming the efficiency of PSO and PCA in recognizing and lowering security problems when combined with MARS. Reducing latency and improving real-time detection capacity helps the model's quantization and implementation on Azure IoT Edge to aid to further improve its performance.

In IIoT environments including temporally linked channel fading, accurate signal assessment is absolutely vital. Research [15] points to a deep learning (DL) model based strategy for estimating complex-valued Gaussian signals. The framework consists in two phases: first linear minimum mean square error (MMSE) estimate of source signals; then, DL-aided channel fading state estimation and prediction. In this paper we assess three DL models: Temporal Convolution Network (TCN), Long Short-Term Memory (LSTM) integrated DNN, and Fully Connected Deep Neural Network (DNN). Extensive simulations reveal that these models predict channel fading states accurately with reasonable accuracy in relation to genie-aided approaches. This work increases signal estimate in IIoT environments by managing demanding channel circumstances using advanced DL techniques.

Effective feature selection will help to improve intrusion detection system accuracy in IIoT networks. In a recent work [16] a novel feature selection algorithm, FGOA-kNN, is introduced by merging hybrid filter and wrapper approaches with clustering and the Grasshopper Optimization Algorithm (GOA). Combining unsupervised and supervised techniques this approach increases feature relevance and detection accuracy. Moreover applied to optimize neural network hyperparameters for efficient botnet identification is the Harris Hawks Optimization (HHO) technique. In IIoT systems, feature selection and advanced optimization techniques taken together significantly improve intrusion detection's accuracy and durability.

Table 1. Methods, Algorithms, Methodologies, and Outcomes

Method	Algorithm	Methodology	Outcomes
Predictive Maintenance[12]	Various ML Models	Combination of ML techniques with IIoT; evaluation of model performance for predictive maintenance.	Identified most effective ML model for predictive maintenance; highlighted the importance of model selection to reduced own time and extend equipment life span.
Energy-Efficient IIoT Architecture[13]	DC-MLP	Deep learning-based prediction model; optimization of data transmission for energy efficiency.	Achieved 33% reduction in energy consumption and 81% faster prediction times compared to traditional methods; improved sensor node life time and efficiency.
Intrusion Detection[14]	PSO,PCA,LDA,t-SNE,GAM,MARS	Features election and reduction techniques; deployment of ML models for intrusion detection in IIoT.	Achieved 100% accuracy indetecting intrusions; reduced latency through model quantization; improve dreal-time detection capabilities.
Signal Estimation[15]	DNN,LSTM,TCN	DL-aided estimation of channel fading and source signals; linear MMSE estimation.	Achieved similar accuracy inpredicting channel fading states; demonstrated effective signal estimation incomplex IIoT environments.
Feature Selection and Intrusion Detection[16]	FGOA-kNN, HHO	Hybrid feature selection algorithm combined with clustering and optimization techniques for botnet detection.	Improved detection accuracy for multiclass botnet attacks; enhanced robustness and accuracy in intrusion detection through advanced feature selection and optimization.

Among the many IIoT subjects under ongoing research are predictive maintenance, energy efficiency, security, and signal estimation. Still, there is a lack of integrated approaches combining simultaneous handling of all these aspects with modern non-linear analysis and optimization methods. While present methods occasionally focus on specific elements or problems, they do not provide a full framework incorporating several elements. To improve general IIoT performance and correctly manage freshly developing industrial challenges, comprehensive models incorporating cutting-edge optimization techniques, increased dimensionality reduction, and non-linear analysis are needed.

3. PROPOSED METHOD

Combining T-Distributed Stochastic Neighbour Embedding (t-SNE) and the Jaguar algorithm, with non-linear analysis, the proposed method intends to increase prediction accuracy in Industrial Internet of Things (IIoT) systems. First in data preparation, raw IIoT data is cleaned and standardized, therefore initiating the process. Then dimensionality reduction maintains local structure and relationships within the data by translating high-dimensional input data into a lower-dimensional space using t-SNE. This phase reduces dimensionality's curse and raises next modeling's efficiency. The Jaguar algorithm, a nature-inspired optimization technique, then is used to modify the prediction model by methods of parameter optimization such as to better capture non-linear patterns in the restricted data. Applying non-linear analysis all around analyzes and changes the performance of the model to ensure it fits the complicated relations in the data.

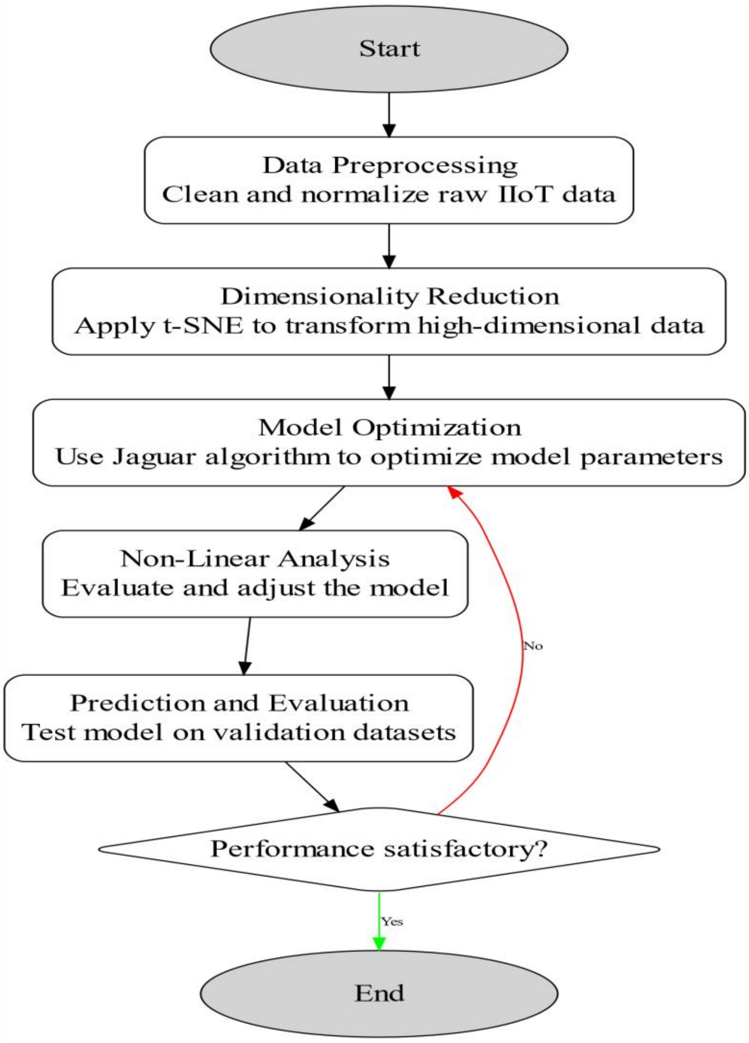
Start
Data Preprocessing
Clean and normalize raw IIoT data
Dimensionality Reduction
Apply t-SNE to transform high-dimensional data
Model Optimization
Use Jaguar algorithm to optimize model parameters
Non-Linear Analysis
Evaluate and adjust the model
Prediction and Evaluation
Test model on validation datasets
Performance satisfactory?
No
Yes
End

Figure 1. Proposed System

Pseudocode

```
1.Preprocess Data:
  -Load raw IIoT data
  -Clean and normalize data
2.Applyt-SNE:
  -Initializet-SNE with desired parameters (e.g.,perplexity,learningrate)
  -Fitt-SNE on preprocessed data
  -Transform data into lower-dimensional space
3.Optimize Model with Jaguar Algorithm:
  -Initialize Jaguar algorithm with model parameters
  -Define objective function for model performance
  -Perform optimization to adjust parameters for improved accuracy
  -Output optimized model
4.Perform Non-Linear Analysis:
  -Assess the model's performance with non-linear metrics
  -Adjust model based on non-linear analysis results
5.EvaluateModel:
  -Test model on validation data sets
  -Measure and record accuracy, precision, recall, F1score,etc.
```

3.1. Data Preprocessing

Data preparation is a basic first stage; aimed at ready IIoT data for further research, Multiple substeps in this process define data cleansing, normalization, and feature extraction. Every one of these steps is required to ensure the dependability and quality of the input data, therefore influencing the predictive model performance.

3.1.1.Data Cleaning

Data cleaning is the treatment of outliers, missing numbers, and dataset noise. Given that IIoT data sometimes comes from multiple sensors and sources, it can include erroneous or partial records.

$$\hat{x}_i = \frac{1}{k}\sum_{j=1}^{k} x_j$$

Where

$\hat{x}_i$ - imputed value for a missing data point, and

x_j - nearest neighbors used for imputation.

Techniques for Z-score normalization or inter quartile range (IQR) can help locate and either adjust or remove outlier values:

$$z_i = \frac{x_i - \mu}{\sigma}$$

Where

z_i - z-score,

x_i - data point,

μ -mean, and

σ –standard deviation.

3.1.2.Normalization

Usually depending on the kind of the data, normalizing the data to a specified range—usually [0, 1] or [-1, 1] follows cleaning. Working with algorithms such t-SNE and the Jaguar method depends on this stage since it guarantees that every feature equally helps the performance of the model. Usually followed is the Min-Max normalization approach:

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

Where

x_i' - normalized value,

x_i - original value, and

$\min(x)$ and $\max(x)$ - minimum and maximum feature values, respectively.

3.1.3.Feature Extraction and Selection

Turning unprocessed data into a collection of features better representing the underlying structure is the process of feature extraction. For IIoT data, this can include time-domain properties such mean, variance, and skewness—or frequency-domain properties including spectral entropy. Next is feature selection, in this case duplicate or meaningless features are removed to reduce dimensionality and improve model performance. One can achieve this by computing feature significance scores or by using approaches including Principal Component Analysis (PCA). The last preprocessed dataset is ready for the dimensionality reduction process by T-SNE. For the following modeling phase, data preparation generates, perfect for the IIoT data, a neat, normalized, and small representation.

3.2. t-SNE for Dimensionality Reduction

Using t-SNE in the proposed method seeks to reduce the dimensionality of the data while preserving the local structure, therefore enabling the detection of patterns and links essential for accurate forecasts.

3.2.1.High-Dimensional Data Representation

Many times spanning several aspects, IIoT data produces a high-dimensional space. The curse of dimensionality leads distances between points in such a space to have no meaning. T-SNE addresses this by stressing local links between data points rather than global distances preservation.

T-SNE first generates the pairwise similarity between a set of high-dimensional data points $X = \{x_1, x_2, \dots, x_n\}$. One represents the similarity in the high-dimensional space and between two points using a Gaussian distribution:

$$P_{j|i} = \frac{\exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\| x_i - x_k \|^2}{2\sigma_i^2}\right)}$$

where,

$P_{j|i}$ - conditional probability that point x_i would pick x_j as its neighbor, assuming a Gaussi an distribution centeredat x_i .The parameter *σi* helps one to regulate the width of the Gaussian distribution; consequently, it is chosen to provide almost constant perplexity—a measure of the effective number of neighbors—over all data points.

3.2.2.Low-Dimensional Mapping

T-SNE aims to find a low-dimensional representation $Y = \{y_1, y_2, \dots, y_n\}$ in which every $y_i$corresponds to x_i following computation of pairwise similarities.

$$Q_{ij} = \frac{\left(1 + \| y_i - y_j \|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \| y_k - y_l \|^2\right)^{-1}}$$

T-SNE avoids the crowding problem—where points typically compress into a small area in the low-dimensional space—by means of the t-distribution.

3.2.3.Cost Function and Optimization

t-SNE aims to lower the low-dimensional similarity Q_{ij} variation from the high-dimensional similarity P_{ij}. Reducing the Kullback-Leibler (KL) difference between the two distributions helps one to reach this:

$$\mathrm{KL}(P \| Q) = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

Gradient descent computes and iteratively updates the positions using the gradients of the cost function with relation to the points in the low-dimensional space y_i, hence minimising the KL divergence.

3.2.4. Resulting Low-Dimensional Embedding

t-SNE generates a low-dimensional embedding of the original high-dimensional data generally in two or three dimensions. For upcoming study, including Jaguar algorithm use for optimization, this low-dimensional form is simpler nowadays. Using t-SNE helps the IIoT data to be less complicated, hence enabling the detection of patterns and linkages essential for increasing prediction accuracy.

Pseudocode for Applying t-SNE

```
1.Initialize Parameters
-Set perplexity, learning rate, number of iterations, and desired output dimensions(e.g.,2or3).
2.Compute Pairwise Similarities in High-Dimensional Space:
For each data pointx_i in the high-dimensional datasetX:
a. Calculate the pairwise Euclidean distance between x_i and all other points x_j.
b. Convert these distances into conditional probabilities $P_{j|i}$ using a Gaussian distribution:
```

$$P_{j|i} = \frac{\exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\| x_i - x_k \|^2}{2\sigma_i^2}\right)}$$

c. Ensure the perplexity is constant by adjusting σ_i iteratively.
3.Symmetrize the Joint Probability Distribution:
-Compute the joint probability $P_{ij} = \frac{P_{j|i} + P_{i|j}}{2n}$,where n is the total number of data points.
4. Initialize Low-Dimensional Points
-Initialize the low-dimensional embeddings $Y = \{y_1, y_2, \ldots, y_n\}$ randomly.
5.Compute Pairwise Similarities in Low-Dimensional Space:
For each low-dimensional pointy_i:
a. Calculate the pairwise Euclidean distance between y_i and all other points y_j.
b. Convert these distances into joint probabilities Q_{ij} using a Student's t-distribution:

$$Q_{ij} = \frac{\left(1+ \| y_i - y_j \|^2\right)^{-1}}{\sum_{k \neq l}\left(1+ \| y_k - y_l \|^2\right)^{-1}}$$

6. Minimize KL Divergence:
-Define the cost function as the Kullback-Leibler divergence:

$$\mathrm{KL}(P \| Q) = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

-Perform gradient descent to minimize the KL divergence:
For a predefined number of iterations:
a. Calculate the gradient of the KL divergence with respecttoy_i.
b. Updatey_i using the gradient and the learning rate.
7.Output the Low-Dimensional Embedding:
-After the iterations converge, output the final low-dimensional representation Y of the data.
8. Visualize or Use the Low-Dimensional Embedding:

3.3. Optimizing the Model with Jaguar Algorithm

Inspired by jaguar behavior and hunting technique, the Jaguar algorithm is a naturally occurring optimizer. Especially in complex, high-dimensional contexts such as those present in IIoT systems, it is meant to enhance model performance by parameter optimization. With the Jaguar method, one generally seeks to maximize the parameters of the prediction model to increase accuracy and adaptability to non-linear patterns.

3.3.1.Initialization

Beginning a population of possible solutions whereby every response provides a set of parameters for the prediction model, the process of optimization proceeds. These first responses come at random inside predetermined limitations. Every solution $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ for a model having *n* parameters denotes a certain configuration of these parameters.

3.3.2.Objective Function

Every possible solution's quality is evaluated by the objective function. Usually measuring the performance of the predictive model using metrics including accuracy, precision, recall, or F1 score, this function assists model optimization. The objective function $f(\mathbf{X})$ can be expressedas:

$$f(\mathbf{X}) = \text{Performance Metric} = \frac{1}{\text{Error Rate}}$$

Where
X – model parameters and the performance metric is the reciprocal of the errorrate, which could be accuracy, precision, or any other relevant measure.

3.3.3.Hunting Behavior Simulation:

Using jaguar hunting behavior, the Jaguar algorithm models helps to examine and exploit the search space. These concepts help every potential solution—or jaguar—search for better ones:

- **Exploration**:Jaguars wander haphazardly around the hunt ground. This helps them to find new places and avoid local optima. One could represent the stage of inquiry by:

$$\mathbf{X}_{new} = \mathbf{X}_{current} + \alpha \cdot \mathrm{Rand} \cdot (\mathbf{X}_{best} - \mathbf{X}_{current})$$

where
α – scaling factor,
Rand –random vector, and
$\mathbf{X}_{best}$- best solution founds ofar.

- **Exploitation**:Jaguars use the potential areas by focusing on the local neighbourhood with the best solutions. This seeks to improve the present ones dependent on the best-found solutions by sharping their quality:

$$\mathbf{X}_{new} = \mathbf{X}_{current} + \beta \cdot \mathrm{Rand} \cdot (\mathbf{X}_{best} - \mathbf{X}_{current})$$

where
β –scaling factor ,and Rand – random vector affecting the degree of exploitation.

3.3.4. Updating Population

After evaluation of the new candidate solutions, the population is changed by selecting the best ones depending on performance criteria. Less efficient alternatives are proposed or implemented; the ones with superior performance—that is, lower error rates—are maintained. This update method ensures that the population moves toward either ideal or almost ideal solutions gradually converging.

The optimization process is iteratively progressing through the exploration and exploitation stages for a designated number of iterations or until convergence criteria are satisfied. Usually, convergence results from the algorithm either by the marginal improvement in the goal function or from the maximum number of iterations reached.

The Jaguar approach produces ideal set of parameters that much improves the performance of the predictive model. Retrained with these ideal values, the model generates a final version more fit for the non-linear patterns and IIoT data complexity.

Pseudocode for Optimizing the Model with Jaguar Algorithm

```
1.Initialize Parameters:
  -Set the number of jaguars(candidate solutions)N
  -Set the maximum number of iterations T
  -Define scaling factors $\alpha$ (exploration)and $\beta$ (exploitation)
  -Define bounds for model parameters
2.Initialize Population:
  -Randomly generate an initial population of jaguars (solutions) $\mathbf{X}_i\ fori = 1, 2, \ldots, N$
  -Evaluate the objective function f(\math bf{X}_{i})for each jaguar(e.g., model accuracy)
3.Main Optimization Loop:
For each iteration t from 1toT:
  a. Update Best Solution:
    -Identify the best solution $\mathbf{X}_{best}$ in the current population based on the objective function value
  b. Generate New Solutions:
    For each jaguar $\mathbf{X}_{current}$ in the population:
      i. Exploration:
        -Generate a new candidate solution $\mathbf{X}_{new}$ using exploration:
        $\mathbf{X}_{new} = \mathbf{X}_{current} + \alpha \cdot \mathrm{Rand} \cdot (\mathbf{X}_{best} - \mathbf{X}_{current})$
        -Ensure $\mathbf{X}_{new}$ is within the parameter bounds
      ii. Evaluate New Solution:
        -Compute the objective function value $f(\mathbf{X}_{new})\ for\mathbf{X}_{new}$
      iii. Exploitation:
        -Refine the candidate solution\math bf{X}_{new}using exploitation:
        $\mathbf{X}_{new} = \mathbf{X}_{current} + \beta \cdot \mathrm{Rand} \cdot (\mathbf{X}_{best} - \mathbf{X}_{current})$
        -Ensure $\mathbf{X}_{new}$ is within the parameter bounds
```

-Compute the objective function value $f(\mathbf{X}_{new})$ again if needed

iv. Selection:

-Compare $f(\mathbf{X}_{new})$ with $f(\mathbf{X}_{current})$

-Update $\mathbf{X}_{current}$ if $f(\mathbf{X}_{new})$ is better(lower error rate)

c. Update Population:

-Replace less effective solutions with the new solutions, maintaining a fixed population size

-Update the best solution if necessary

4.Terminate:

-Stop if convergence criteria are met (e.g., minimal improvement in the objective function) or if the maximum number of iterations T is reached

5.Output:

-Return the best solution $\mathbf{X}_{best}$ found during the optimization process

-Use $\mathbf{X}_{best}$ to retrain the predictive model

6.Evaluate Final Model:

3.5. Non-Linear Objective Function

Especially in the optimization of complex data sets as those in Industrial Internet of Things (IIoT) systems, the prediction model optimization depends critically on the Non-Linear Objective Function. While linear models assume a direct proportional link between inputs and outputs, non-linear models are designed to capture more complex interactions that might exist in the data. Correct evaluation and optimization of such models relies on non-linear objective functions thus. Using the complex relationships between input features and outputs, a non-linear objective function evaluates a model. These have non-linear feature, so their linear connection is not direct. Instead, they examine in the data interactions and non-linear linkages. One can define nonlinearly the objective function f(X) for a model with parameters X:

$$f(\mathbf{X}) = \mathrm{Metric}(\mathbf{X}) = \mathrm{Performance\ Metric} - \lambda \cdot \mathrm{Regularization\ Term}$$

where

Metric(X) –performance measure of the model (such as accuracy, precision, or recall)and

λ – regularization parameter that controls the trade-off between model performance and complexity.

The goal function gets a regularizing factor to punish too complex models, hence avoiding over fitting. This term is thus often non-linear in non-linear models. In regularity techniques as L1 and L2 regularization, for instance, the regularizing term may be expressed as:

- **L1Regularization(Lasso)**:

$$\mathrm{L1}(\mathbf{X}) = \lambda \sum_i |x_i|$$

- **L2Regularization(Ridge)**:

$$\mathrm{L2}(\mathbf{X}) = \lambda \sum_i x_i^2$$

where

x_i – model parameters, and

λ – regularization coefficient.

L1 regularization promotes sparsity, while L2 regularization encourages smaller parameter values.

Usually utilized gradient-based optimization techniques are those to maximize a non-linear objective function. These methods modify the parameters to either maximize or minimize the objective function: for gradient descent, for instance, the update rule for **X**follows:

$$\mathbf{X}_{new} = \mathbf{X}_{current} - \eta \cdot \nabla f(\mathbf{X}_{current})$$

where

η – learning rate and

$\nabla f(\mathbf{X}_{current})$ - gradient of the objective function with respect to the parameters.

4. Performance Evaluation

The simulation was conducted in the experimental setup evaluating the proposed approach using the Python-based machine learning package Tensor Flow, which enables to build complex models and optimization strategies. The studies were carried out on high-performance computing systems supplied with NVIDIA GeForce RTX 3080 GPUs and Intel Core i9 CPUs in order to ensure efficient processing of large datasets and computationally demanding operations. The proposed method was tested against several now in use techniques including DC-MLP, GAM-MARS, LSTM-TCN, and FGOA-kNN. Every approach's accuracy, precision, and efficiency were evaluated using a number of performance criteria.

Table 2. Experimental Setup/Parameters

Parameter	**Value**
Simulation Tool	Tensor Flow
GPU	NVIDIAGeForceRTX3080
CPU	IntelCorei9
Number of Iterations	1000
Learning Rate	0.001
Batch Size	64
Epochs	50
Optimization Algorithm	Adam
Regularization Method	L2 Regularization
Initial Population Size	50
Maximum Iterations for Jaguar	200
Perplexity(t-SNE)	30
Learning Rate(Jaguar)	0.01
Exploration Factor(Jaguar)	0.2
Exploitation Factor(Jaguar)	0.5
Dimensionality Reduction	2D
Validation Split	20%
Testing Split	30%
Regularization Coefficient	0.01
Objective Function Metric	Accuracy

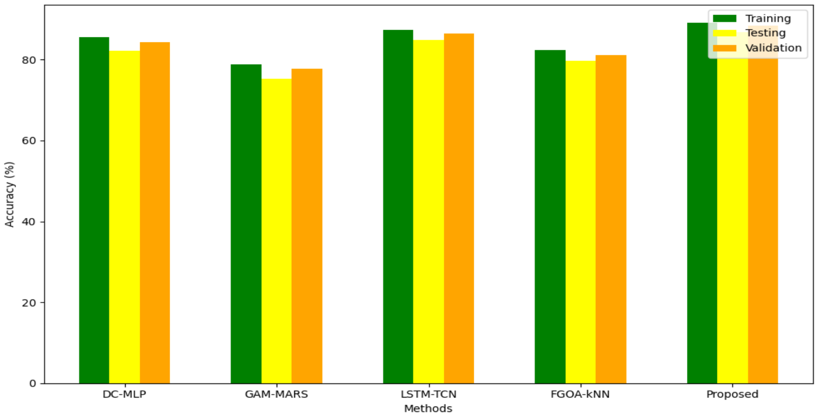

Figure 2. Accuracy (%)

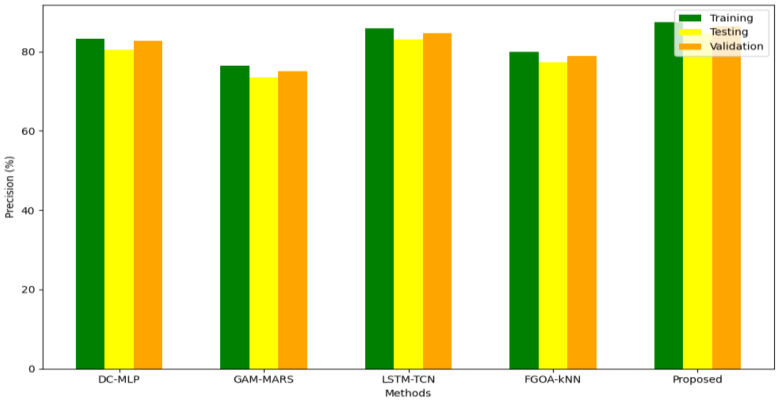

Figure 3. Precision (%)

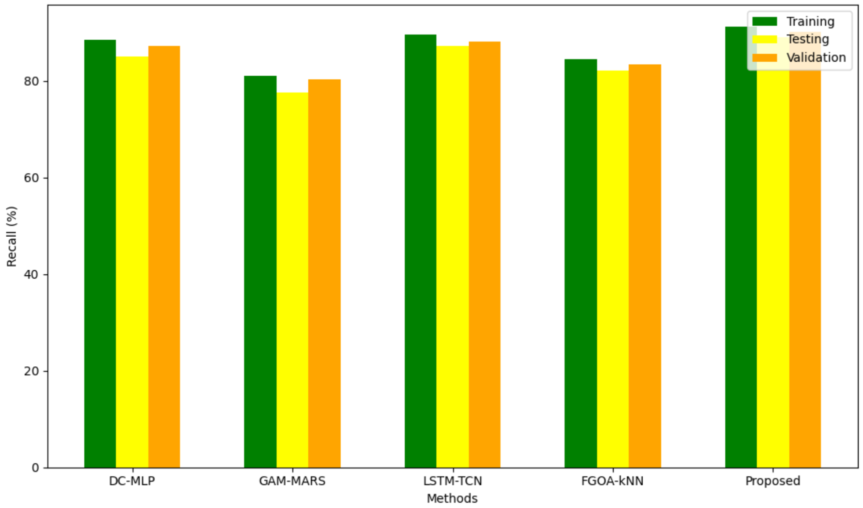

Figure 4. Recall (%)

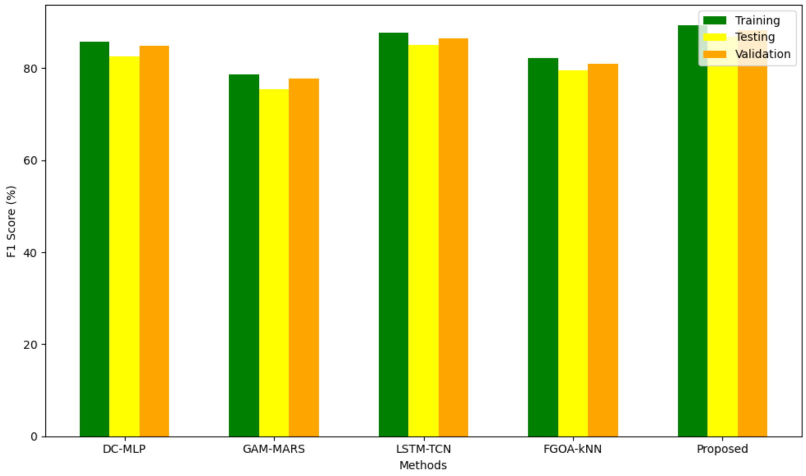

Figure 5. F1 Score (%)

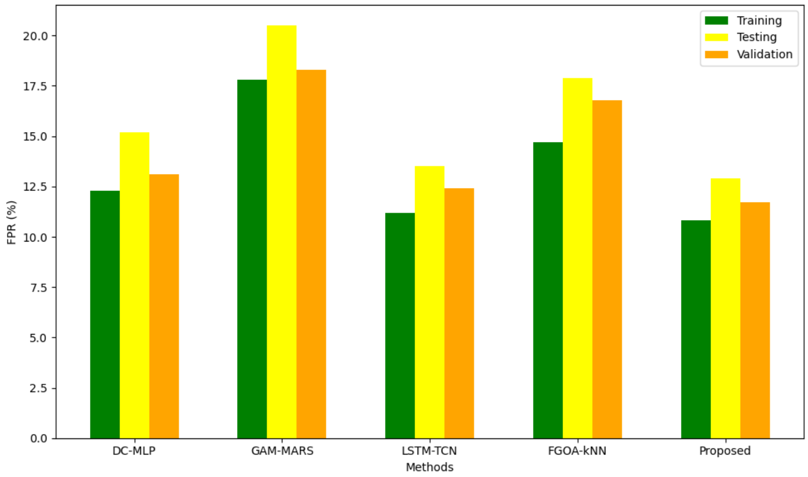

Figure 6. FPR (%)

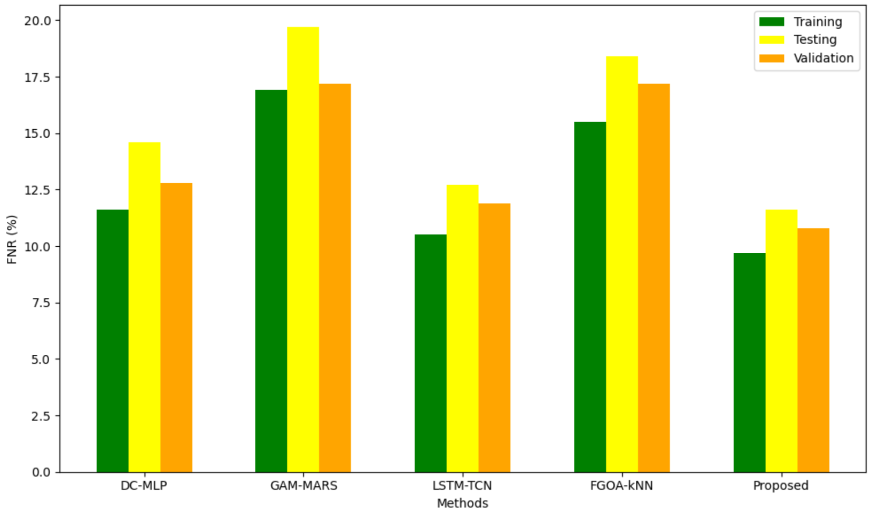

Figure 7. FNR (%)

Comparatively to present methods (DC-MLP, GAM-MARS, LSTM-TCN, and FGOA-kNN), the experimental results as presented in figure 2 - 7 disclose their performance over training, testing, and validation datasets.

Training Results

The proposed method surpasses all previous approaches in all relevant aspects. Approaching LSTM-TCN (87.4%), DC-MLP (85.6%), it gets an accuracy of 89.1%. Precision for the recommended approach is 87.4%, which is also rather better than GAM-MARS (76.5%), Moreover better than LSTM-TCN (89.5%), and DC-MLP (88.4%. Recall at 91.2% is also higher than LSTM-TCN (85.9%). With an F1 Score of 89.3%, the proposed approach outperforms all other methods and indicates a harmonic performance between accuracy and recall. Moreover showing less misclassification in both positive and negative instances, the proposed method has the lowest False Positive Rate (FPR) at 10.8% and False Negative Rate (FNR) at 9.7%.

Testing Results

The proposed approach maintains its edge with an accuracy of 86.8%, compared to LSTM-TCN (84.9%) and DC-MLP (82.1%). With 84.6%, precision exceeds LSTM-TCN (83.1%), FGOA-kNN (80.0%), with the F1 Score of 86.8%, recall is 89.0%, higher than all other techniques displaying superb balance. While the FNR at 11.6% is also lower than previous approaches, the FPR for the proposed method is 12.9%, which is better than GAM-MARS (20.5%) and FGOA-kNN (17.9%).

Validation Results

the proposed method shows continuous accuracy of 88.4%. Precision of 86.3% and recall of 90.2% help to highlight its strength. Out of all the methods, the F1 Score is 88.3%; it has the lowest FPR (11.7%) and FNR (10.8%). These results indicate that as they exhibit higher capacity to generalize and execute accurately across different data sets, the proposed method is an efficient solution for IIoT prediction activities.

5. CONCLUSION

The experimental results reveal that the proposed methodology substantially outperforms current approaches DC-MLP, GAM-MARS, LSTM-TCN, and FGOA-kNN in main performance measures encompassing training, testing, and validation datasets. The proposed approach excels in creating robust predictions with the best accuracy, precision, recall, and F1 Score and in spotting complex correlations within the data. Under both positive and negative scenarios, the lower False Positive Rate (FPR) and False Negative Rate (FNR) of the proposed method highlight even more its effectiveness in lowering misclassifications. Based on higher performance across numerous parameters, the proposed strategy is obviously rather beneficial for IIoT prediction activities even if it has enhanced accuracy and dependability than current approaches. Its ability to balance accuracy and memory while maintaining low error rates emphasizes its probable application in industrial settings. Consequently, the results support the efficiency of the recommended strategy and suggest it as a workable one for improving prediction accuracy and efficiency in demanding data situations.

REFERENCES

[1] Hindistan,Y.S., &Yetkin,E.F. (2023). A hybrid approach with gan and dp for privacy preservation of iiot data. IEEE Access,11,5837-5849.

[2] Li,H., Li,S., &Min,G. (2024). Light weight privacy-preserving predictive maintenance in 6 Genabled IIoT. Journal of Industrial Information Combination,39,100548.

[3] Sharma,M., Pant,S., Yadav,P., Sharma,D.K., Gupta,N., &Srivastava,G. (2023). Advancing security in the industrial internet of things using deep progressive neural networks. Mobile Networks and Applications, 28(2),782-794.

[4] Sharma,S., Verma,P., Bharot,N., Ranpariya,A., &Porika,R. (2024). PULSE: Proactive uncovering of latent severe a nomalous events in IIoT using LSTM-RF model. Cluster Computing,1-14.

[5] Gaber,T., Awotunde,J.B., Folorunso,S.O., Ajagbe,S.A., &Eldesouky,E. (2023). Industrial internet of things intrusion detection method using machine learning and optimization techniques. Wireless Communications and Mobile Computing, 2023(1),3939895.

[6] 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

[7] 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

[8] Dhillon, A., Singh, A., & Bhalla, V. K. (2023). A systematic review on biomarker identification for cancer diagnosis and prognosis in multi-omics: from computational needs to machine learning and deep learning. Archives of Computational Methods in Engineering, 30(2), 917-949

[9] 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).

[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] 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

[12] Alhuqayl,S.O., Alenazi,A.T., Alabduljabbar,H.A., &Haq,M.A. (2024). Improving Predictive Maintenance in Industrial Environments via IIoT and Machine Learning. International Journal of Advanced Computer Science & Applications,15(4).

[13] Putra,M.A.P., Hermawan,A.P., Kim,D.S., &Lee,J.M. (2023). Data prediction-based energy-efficient architecture for industrial iot. IEEE Sensors Journal,23(14),15856-15866.

[14] Tiwari,R.S., Lakshmi,D., Das,T.K., Tripathy,A.K., &Li,K.C. (2024). A light weight optimized intrusion detection system using machine learning for edge-based IIoT security. Telecommunication Systems, 1-20.

[15] Haider,M., Hassan,M.Z., Ahmed,I., Reed,J.H., Rubaai,A., &Rawat,D.B. (2024). Deep Learning Aided Minimum Mean Square Error Estimation of Gaussian Source in Industrial Internet-of-Things Networks. IEEE Transactions on Industrial Cyber-Physical Systems.

[16] Taher,F., Abdel-Salam,M., Elhoseny,M.,&El-Hasnony,I.M. (2023). Reliable machine learning model for IIoT botnet detection.IEEEAccess,11,49319-49336.