Dynamic Multi-Objective None-Sorted Genetic Algorithm III Deep Long Short-Term Memory Incorporating Dropout for Collision Detection in Internet of Vehicles

Salisu Abdullahi[1], Dr. Abdulsalam Y. Gital[2], Dr. Fatima Umar Zambuk[3], Saleh Waziri Mustapaha[4], Dr. Usman Ali Abdullahi[5]

[1]Federal Polytechnic, Bauchi, Department of Computer Science

[2,3,4] Abubakar Tafawa Balewa University, Faculty of Computing, Department of Computer Science, Bauchi, Nigeria

[5] Federal College of Education Technical, Gombe, Department of Computer Science

DOI: https://doi.org/10.51583/IJLTEMAS.2025.1410000059

Received: 12 October 2025; Accepted: 23 October 2025; Published: 10 November 2025

Abstract. The Internet of Vehicles (IoV) is an emerging field with significant research and commercial potential. A central challenge in IoV is to aggregate the vast volumes of data generated by interconnected vehicles and transform it into actionable knowledge for intelligent decision-making, such as detecting vehicle collisions. While various classifiers have been developed for collision detection, there is a lack of rigorous research on selecting the most impactful features for these detections. Many existing studies rely on single algorithms and simple correlation coefficients for feature identification, which can be a limitation. To address this, this paper proposes a dynamic model that combines a Non-dominated Sorting Genetic Algorithm III (NSGA-III) with a deep Long Short-Term Memory (LSTM) network incorporating dropout to detect vehicle collisions in IoV environments. The model aims to simultaneously minimize the feature subset, reduce computational time, and maximize collision detection accuracy. NSGA-III evolutionary algorithm efficiently explores the feature space to optimize multiple objectives, while the deep LSTM is well-suited for capturing temporal dependencies in the sequential data generated by vehicles. The model was trained and evaluated using a dataset generated in a VISSIM traffic simulation environment, which recreated various urban driving scenarios. The experimental results demonstrate that the NSGA-III + Deep LSTM-RD model significantly outperforms baseline algorithms like deep LSTM, DRNN, GANN, and ANN, especially when using a reduced set of features. The model achieved high accuracy and lower error rates while maintaining fast execution times, highlighting the importance of combining advanced optimization techniques with deep learning for robust and adaptable predictive models.

Keywords: Deep Long Short-Term Memory, Non-dominated Sorting Genetic Algorithm III, Internet of Vehicles, Vehicle Collision Detection.

I. Introduction

The Internet of Vehicles (IoV) is an emerging field where vehicles function as interconnected entities on the internet, enabling real-time data sharing and communication. IoV holds significant commercial and research potential, drawing interest from both industry and academia. Through inter-vehicle and intra-vehicle communication, along with real-time road condition monitoring, IoV generates vast volumes of data useful for analyzing behavioral patterns, environmental conditions, and device performance.

However, the core challenge of IoV goes beyond simply routing, exchanging, or uploading data from local sensors to the internet. The real issue lies in aggregating this data and transforming it into meaningful information and actionable knowledge to support intelligent decision-making. For example, in the vehicle insurance industry, companies can tailor insurance premiums based on driving behavior by examining correlations between driver errors, traffic violations, and accident involvement (Chen et al., 2019). Amid this digital automotive revolution, the ability to extract valuable insights from massive data streams becomes essential. Feature selection, a crucial aspect of data analysis, plays a central role in identifying the most relevant variables in IoV environments. Effectively navigating this sea of information requires a deep understanding of the interrelated dynamics of vehicular data, including driving behavior, environmental context, and system performance (Bolón-Canedo et al., 2015).

In the domain of machine learning, feature selection presents a significant challenge. Researchers have made considerable efforts to develop classifiers for vehicle collision detection within the IoV. Some studies focus specifically on identifying which features most strongly influence collision risk. Despite growing interest, there remains a shortage of rigorous research dedicated to selecting impactful features for collision prediction in IoV. The few existing studies largely depend on correlation coefficients to identify relevant features (Almutairi et al., 2023). One of the critical challenges in IoV-enabled mobility is vehicle collisions, which may result in injuries, disabilities, vehicle damage, traffic congestion, and loss of lives and property. Being able to predict the likelihood of such collisions can empower both human-driven and autonomous vehicles to take preventive action. As IoV research progresses, the essential subset of features required to build a robust vehicle collision alarm system remains an open question (Almutairi et al., 2023).

This paper explores the importance of feature selection within the IoV framework and highlights methodologies for deriving meaningful insights from the vast array of data generated by connected vehicles. By examining the intricacies of feature selection in IoV, this study aims to enhance decision-making, improve road safety, and contribute to the development of a more intelligent transportation ecosystem. As we navigate the data-rich highways of the IoV, selecting the right features is critical to driving the future of connected mobility. Preventing vehicle collisions in this context requires the development of predictive classifiers capable of early detection and timely intervention (Sun et al., 2022).

Several studies have explored different methods for collision detection. For instance, Lima et al. (2020) utilized Long Short-Term Memory (LSTM**)** networks, while Almutairi et al. (2023) employed the Non-dominated Sorting Genetic Algorithm III (NSGA-III) to identify key features influencing vehicle collisions. However, both studies and much of the existing literature rely on a single algorithm for solving the vehicle collision problem. Recent findings suggest that **hybrid approaches**, which combine multiple intelligent algorithms, often outperform individual algorithms in predictive tasks (Almutairi et al., 2023b).

In light of this, the present study proposes a hybrid model combining **NSGA-III** and deep **LSTM** to predict vehicle collisions in IoV. The goal is to simultaneously minimize the feature subset, reduce computational time and maximize prediction accuracy. LSTM is well-suited for capturing temporal dependencies in sequential data, making it ideal for time-series tasks such as predicting collision likelihood (Yu et al., 2019). In parallel, NSGA-III's evolutionary algorithm efficiently explores the feature space, optimizing multiple objectives concurrently (Cui et al., 2019). Thus, this hybrid approach leverages the strengths of both methods offering a robust solution for feature selection and accurate collision prediction in IoV scenarios.

Related works

This section presents works related to collision detection in the Internet of Vehicles (IoV), providing state-of-the-art insights on the topic. Numerous studies on collision detection have been published in the literature, including research on collision warning systems (Baek et al., 2020), collision detection in cooperative inland vessels (Hammedi et al., 2022), and human vehicle collision detection (Qu et al., 2020), among others. However, the focus of this research is specifically on collision detection within the IoV environment.

Therefore, the related works presented in this section emphasize collision detection in IoV. One of the key services in the IoV ecosystem is vehicle collision detection. To ensure low-latency communication and real-time responsiveness for collision avoidance, this service is typically deployed at the edge using multi-access edge computing (Brik & Ksentini, 2021). The following paragraphs review studies that have applied machine learning techniques for collision detection in IoV.

Chang et al. (2021) proposed an IoV-based collision detection framework that leverages YOLOv4 to predict the future positions of vehicles and prevent potential collisions. The framework incorporates both vehicle dynamics and machine learning techniques, with YOLOv4 specifically used for forecasting vehicle trajectories. The results demonstrated that the proposed method provided more accurate and stable predictions of future vehicle positions. However, the study employed a single algorithm without incorporating hybrid approaches. Previous research (e.g., Chiroma et al., 2020) has shown that hybrid algorithms often outperform individual models. Furthermore, the study only compared its results with existing works in the literature and did not benchmark the model against other classical algorithms on the same dataset, limiting the robustness of the evaluation.

To estimate vehicle density and compute the resources required for effective collision detection, Brik and Ksentini (2021) proposed a deep learning-based framework deployed at the mobile edge computing layer. The model was trained and evaluated using a dataset representing urban mobility patterns. The findings indicated that the framework could successfully predict both the necessary computing resources and system requirements for collision detection in an IoV environment. However, a notable limitation of this study is its omission of environmental factors particularly adverse weather conditions such as fog which significantly affect vehicle movement. This oversight may hinder the practical deployment of the system under real-world weather conditions. Chang et al. (2019) introduced an IoV-based system utilizing squeeze-and-excitation networks and densely connected convolutional networks to detect vehicular collisions. Their deep learning model was trained for traffic collision detection and hosted on a cloud server as part of a cloud-based management platform. Similarly, Chen et al. (2018) proposed a deep learning architecture within the IoV framework that includes collision detection sensors, a cloud-based training server, and a web-based service platform. The framework was designed to detect traffic accidents using deep learning, and experimental results demonstrated high accuracy in identifying traffic collisions.

Nevertheless, both Chang et al. (2019) and Chen et al. (2018) did not compare their proposed methods against other established algorithms. Without such benchmarking, it becomes difficult to assess the performance and advantages of their frameworks relative to traditional or competitive approaches. Hybrid approaches are generally more effective than individual algorithms, as demonstrated in previous studies (e.g., Chiroma et al., 2020). However, the referenced study only compared the performance of the convolutional neural network (CNN) with traditional methods namely Honda, Berkeley, and MCWAw ithout benchmarking it against other deep learning models from the same algorithmic family. In a related work, Chen, Xiang et al. (2018) utilized a genetic algorithm (GA) to train an artificial neural network (ANN), resulting in a GA-ANN model for detecting rear-end collisions in IoV environments. Despite this effort, several challenges were noted. One major limitation is the decline in GA-ANN performance as the dataset size increases significantly. As Fong et al. (2018) noted, some researchers argue that using metaheuristic methods like GA for ANN optimization is not always necessary.

This is because local minima often emerge from small, progressive changes in the hidden layer's node-weight interactions, and directly minimizing error can often yield reliable solutions. Moreover, excessive attempts to find a global optimum may overextend the neural network, reduce its flexibility and potentially cause overfitting. Overfitted ANNs, as observed by Liu (2008), may lack generalization ability unlike models trained with gradient descent that can effectively settle into beneficial local minima to maintain a balance between accuracy and generalization (Fong et al., 2018).

The presented framework incorporates a physical distancing notification system by harnessing cutting-edge technologies, specifically Deep Learning (DL) and the Internet of Vehicles (IoV). Each vehicle is outfitted with a switching camera system employing both thermal and vision imaging capabilities. Subsequently, the Faster-RCNN algorithm is utilized to gauge and identify instances of physical distancing violations between objects of the same class. The performance of the proposed architecture is then evaluated through Vehicle-to-Infrastructure communication. The results obtained affirm the applicability and efficiency of our proposal in delivering timely notifications concerning social distancing violations.(C.-j. Li, Qu, Wang, & Liu, 2021) proposes a VGG16 model advancement on Faster R-CNN by introducing a cross-layer fusion multi-object detection and recognition. From their experimental and data analysis, the result shows that the improved Faster RCNN architecture combines low-level and high-level image semantic attributes.

This has improved image object classification and recognition. A mixed dataset was used, that is manually labelled for good detection outcome. The improved R-CNN model for object image classification and recognition has advanced the mean accuracy from the labelled cityscapes and KITTI datasets.(Tian & Hu, 2019) proposed a deep convolutional neural network to handle the issue concerning poor foreground extraction in dynamic background images, an image foreground target detection technique. The foreground can accurately detect both in the complex background and foreground occlusion. VGG16 based network was used for the feature extraction map. They Applied the deconvolution technique with the pyramid pooling technique to upgrade the problem of VGG16 on classifying the entire image.

The proposed model uses TensorFlow in training the network. The proposed model is compared to the traditional target model achieved good results in terms of solid robustness in a complex scene.(Hou, Li, Zhang, Wang, & Jiao, 2019) proposed a model for object detection and tracking technique built on deep convolutional neural networks for wide-swath high-resolution optical remote sensing videos. The proposed model divides the video frames into sub-sample to avoid the challenges that occur due to data size. To detect images at the sliding window effectively, they used an overlapping sliding window sampling method. The proposed network was designed based on the region of interest (ROIs) to track images from the previous frames in the video and used directly on the current frame. The result shows the valid and general use of their dataset for the proposed algorithm.

II. Methodology

The methodology section details the approach taken in the study, which combines a simulated environment for data generation with a multi-objective optimization model for feature selection and collision detection. The process began by creating a virtual traffic environment using VISSIM to collect data under diverse conditions that reflect real-world traffic dynamics. The core of the methodology lies in its multi-objective feature selection strategy making deep LSTM dynamic with feature selection, which utilizes the Non-dominated Sorting Genetic Algorithm III (NSGA-III). This approach simultaneously aims to minimize the feature subset and computational time while maximizing the accuracy of collision detection. To evaluate the effectiveness of the selected features, the study employed five machine learning classifiers: Support Vector Machine (SVM), Random Forest Classifier (RFC), Gaussian Naïve Bayes (GNB), Decision Tree Classifier (DTC), and K-Nearest Neighbor (KNN). This multi-faceted approach allows for a rigorous comparison and validation of the proposed hybrid model against baseline methods.

Datasets

To generate data for vehicle collision analysis within the IoV framework, a simulated environment was first developed using the VISSIM traffic simulation tool. This environment was specifically tailored to enable data collection under varying conditions that reflect real-world traffic dynamics. By recreating multiple traffic scenarios, the simulation provided a robust foundation for capturing the necessary data points related to vehicular collisions. The simulated IoV environment was designed to include various driving scenarios. It featured a road layout spanning an area of 1500 by 1500 meters, constructed with a single-lane road that incorporated bends, multiple lanes, and distinct speed zones. These road elements allowed the simulation to mimic complex urban driving conditions, which are essential for studying interactions that could lead to collisions. The road setup also included areas with speed limitations and multiple signalized intersections. Signal controllers were placed strategically, controlling the flow of traffic with standard green, amber, and red lights. This inclusion helped to simulate stop-and-go traffic behavior and its effect on vehicle interactions. By recreating a controlled signal environment, the simulation could better reflect real-world driving behaviors at intersections.

Speed dynamics in the simulated IoV environment were diverse. The minimum speed recorded was 50 km/h, while the average and maximum speeds were set at 95 km/h and 140 km/h, respectively. These variations helped in capturing how vehicles behave across different speed limits, contributing valuable insights into collision risks under varying velocity conditions. Traffic density within the simulation was also variable, ranging from 30 to 180 vehicles in motion at any given time. This range was selected to reflect both light and heavy traffic scenarios, allowing for an analysis of how vehicle density affects collision likelihood. As vehicles navigated through different traffic zones, their behavior in terms of acceleration, deceleration, and stopping was tracked. To simulate real-world complexity further, the IoV environment was reconfigured repeatedly to test various traffic conditions. Special attention was given to low-speed zones where speed limits varied between 5 km/h and 50 km/h. These zones are typically prone to sudden braking and close following distances, which are key contributors to collisions. Observing vehicle behavior in such zones provided critical collision-related data.

A wide range of traffic features were tracked during the simulation. These included the driving state of each vehicle, vehicle clearance, inter-vehicle spacing, gross and net safety distances, and both gross and net following distances. Interaction states, delay times, and lane usage were also monitored. These features were essential for evaluating the dynamics that lead to potential collisions. Furthermore, speed measurements for individual vehicles were recorded with different traffic events. The detailed observation of speed in combination with following distance and lane change behavior allowed for a comprehensive understanding of how driver actions contribute to crash risks.

In addition to these technical variables, certain environmental and human behavioral factors were also considered. Key features used in the study included the number of lanes, the status or behavior of the driver, environmental conditions, vehicle speed, inter-vehicle distance, and braking capability. These attributes helped create a holistic view of the factors influencing vehicle collisions in the IoV environment. The configuration of all these variables and how they interact within the VISSIM simulation environment. This configuration laid the groundwork for extracting meaningful insights about vehicle collision behavior in a controlled yet realistic IoV setting, supporting the study's goal of improving road safety through intelligent vehicular systems.

The NSGA-III + Deep LSTM-RD

NSGA-III + Deep LSTM-RD (where R = ReLU and D = Dropout) is a hybrid model that combines a **multi-objective optimization algorithm** (NSGA-III) with a **deep learning architecture** of LSTM enhanced by ReLU activation and dropout regularization. Here's a breakdown of each component and how they work together:

NSGA-III (Non-dominated Sorting Genetic Algorithm III):

- NSGA-III is a powerful **evolutionary multi-objective optimization algorithm** designed to solve complex problems with **three or more conflicting objectives**.
- It improves upon earlier versions (like NSGA-II) by introducing a **reference-point-based selection** strategy, which ensures a well-distributed set of solutions across the objective space.
- In the context of machine learning or deep learning, **NSGA-III is often used to optimize hyperparameters** (e.g., number of layers, learning rate, batch size) or feature subsets for better model performance across multiple criteria such as **accuracy, training time, and generalization**.

Deep LSTM-RD (Long Short-Term Memory with ReLU and Dropout):

- **LSTM** is designed to capture long-term dependencies in sequential data, making it suitable for collision detection in IoV environments.
- **"R" – ReLU (Rectified Linear Unit):** ReLU is an activation function used in the LSTM network to introduce non-linearity and accelerate training. It outputs zero for negative values and passes positive values as-is, helping avoid the vanishing gradient problem.
- **"D" – Dropout:** Dropout is a regularization technique applied during training to prevent overfitting. It randomly deactivates a fraction of the neurons in a layer, forcing the network to learn more robust features.

Combined NSGA-III + Deep LSTM-RD:

- The NSGA-III algorithm is used to **automatically select subset vehicle collision features** before passing to optimize **Deep LSTM-RD model,** such as the number of LSTM layers, dropout rate, learning rate, number of hidden units, neurons and momentum.
- The **LSTM-RD** network, once optimized, is responsible for **learning temporal patterns** in vehicle movements or collision indicators, with **ReLU ensuring effective training** and **dropout maintaining generalization.**
- The hybrid approach leverages **NSGA-III's strength in finding a balanced trade-off** between multiple objectives including optimal feature selection, accuracy and enhance **LSTM's ability to model complex vehicles collision behavior**, resulting in a robust and efficient prediction of vehicle collision in IoV.

ReLU Activation (Algorithm 1): This algorithm applies the ReLU function to hidden neurons, passing only positive values and setting negative values to zero, effectively deactivating certain neurons. This selective approach allows the model to focus on positive feature contributions, simplifying the network and making it computationally efficient.

Algorithm 1 *ReLu at the hidden neurons*

```
For an input features:
    1. If features are positive, $f(x)f(x)f(x)$ outputs feature selected,
Allowing Positive values to pass unchanged.
If xxx is negative or zero,
f(x)f(x)f(x) outputs 000,
Deactivate neuron's contribution to the hidden network layer.
```

Dropout (Algorithm 2): Dropout is implemented by generating a mask based on a specified probability. During training, this mask randomly deactivates neurons (sets their output to zero) to prevent overfitting. At inference time, the model’s output is scaled to maintain consistency with the training phase.

Algorithm 2 Dropout

```
Input activations: xxx
Dropout rate: ppp (probability of "dropping" a neuron)
Generate Mask
    M=rand(x.shape)≥pM = \text{rand}(x.\text{shape}) \geq pM=rand(x.shape)≥p
    (Mask with 1 where neurons are kept, 0 where dropped)
Apply Mask (Training)
    xdrop=x×Mx_{\text{drop}} = x \times Mxdrop=x×M
Scale Output (Inference)
    xinf = x×(1−p)x_{\text{inf}} = x \times (1 - p)xinf=x×(1−p)
```

Combined ReLU with Dropout (Algorithm 3)

This combined approach first applies ReLU to the neurons and then introduces dropout to further regularize the model. By dropping neurons probabilistically, this combination allows for sparse activation, which is beneficial in reducing overfitting while allowing the network to learn complex patterns.

Algorithm 3 Dropout Plus ReLu

```
import numpy as np
  def relu_with_dropout(x, dropout_prob):
      Step 1: Apply ReLU
          x = np.maximum(0, x)
  Step 2: Apply Dropout
   mask = (np.random.rand(*x.shape) >= dropout_prob).astype(float) # Create dropout    mask
        x *= mask # Apply mask
     Step 2:  # During inference, scale by
            (1 - dropout_prob) to adjust for dropped neurons
return x
```

For each layer with neurons x_i:

1. **ReLU Activation**: Apply x_i =max (0, x_i) x_i = \max (0, x_i) x_i =max (0, x_i) to each neuron’s output.
2. **Dropout**: Generate a random mask for each neuron based on the dropout probability ppp:
   - If a random number for neuron xix_ixi is less than ppp, set x_i =0 *xi* = 0 x_i =0 (drop that neuron).
   - If the random number is greater than or equal to ppp, keep the neuron’s output.
3. **Scale Output**: During training, dropout is applied with a probability ppp, so to maintain the expected output range during inference, multiply each neuron by (1−p)(1 - p)(1−p) during inference.

NSGA-III + Deep LSTM (Algorithm 4)

NSGA-III, a multi-objective genetic algorithm, is used here to select the optimal subset of input features before feeding them into the LSTM. It aims to balance objectives like feature reduction and classification accuracy. The LSTM is then set up with ReLU activation functions and dropout, leveraging the selected feature subset. The steps include initializing feature subsets, evaluating them based on objectives, and iteratively refining the population using crossover, mutation, and selection based on non-dominated sorting.

Algorithm 4: NSGA-III + Deep LSTM with ReLU and Dropout (NSGA-III + Deep LSTM-RD)

```
1. Feature set collusion detection features
2. LSTM parameters: hidden size, layers, dropout rate y
3. NSGA-III objectives: minimize feature count and maximize accuracy
4. NSGA-III for Feature Selection
    a. Initialize population of feature subsets collusion detection features from collusion detection features
       4.2 Evaluate objectives on each subset in x
       (e.g., accuracy, feature count)
    b. Apply non-dominated sorting on collusion detection features
       4.4 Select next generation with NSGA-III operator
       (crossover, mutation, elitism and diversity preserved)
    c. 4.5 Iterate until termination criterion is met
       (e.g., max generations)
5. Select Optimal Feature Subset
   5.1 Choose best subset FoptF_{\text{opt}}Fopt from final population
6. Deep LSTM Initialization
     6.1 Define LSTM layers with ReLU activation
      6.2 Apply Dropout with rate x after each LSTM layer
7. Input Layer
   4.1 Set FoptF_{\text{opt}} Fopt as input to LSTM
    a. LSTM Forward Pass
       7.1 Compute LSTM cell output with ReLU and dropout
```

Experiment

This section provided the detailed description of the experiment including multiple data partitioning**,** feature usage**,** and parameter settings for modeling NSGA-III + Deep LSTM-RD to predict **vehicle collisions** in IoV environment as developed in the preceding sections:

Experimental Setup and Configuration

The experiments were conducted on a Mac system powered by the M2 chip. The hardware configuration includes an 8-core CPU—comprising performance and efficiency cores—a 10-core GPU, a 16-core Neural Engine, and a memory bandwidth of 100 GB/s. On the software side, the experiments were executed in a Python environment within the Anaconda platform. The parameter settings for the NSGA III and NSGA2 algorithms, shown in below, were adapted from Kumar and Yadav (2018).

Parameter Settings for NSGA Family Algorithms

- Reference Points Number: 0.9
- Crossover Probability: 0.9
- Crossover Distribution Index: 0.25
- Mutation Probability: 0.2
- Hypervolume: [0,0]– [1,1]
- Generations: 500
- Population Size: 200
- Population Initialization: Random
- Objectives: 2

Multi-objective Feature Selection Strategy

This study applies a multi-objective feature selection approach, which evaluates groups of features simultaneously. NSGA III is utilized alongside Gain Ratio (GR) as the filter evaluation criterion and a bi-directional selection strategy to identify the most informative subset of features related to vehicle collisions in IoV (Internet of Vehicles) environments. For comparative purposes, NSGA2 is also employed using the same GR filter and bi-directional selection across filter-based, wrapper-based, and hybrid-based methods.

Execution Methodology

Binary chromosomes represent candidate feature subsets for selection within the NSGA framework. Since the problem is multi-objective, the model aims to minimize the number of selected IoV features while maximizing vehicle collision detection accuracy. Both NSGA2 and NSGA III select subsets of features, which are then ranked using the GR-based filter method. Classification performance is used to order the subsets, and five well-known machine learning algorithms SVM, RFC, GNB, DTC, and KNN are used to evaluate the discriminant power of the selected features. These algorithms were chosen due to their robustness in scenarios involving moderate-sized datasets.

Classifier Evaluation and Performance Analysis

Each selected subset of features was used to train all five classifiers, and their performance in detecting vehicle collisions was assessed. The detection accuracy returned by each classifier reflects the effectiveness of the selected feature subsets. The evaluation was carried out using both NSGA III and NSGA2 to determine which approach produced superior feature selection in terms of predictive performance.

Algorithm 1: Algorithm for the vehicle collision subset feature selection in IoV

```
    Begin
    SET hardware: = Mac with M2 chip (8-core CPU, 10-core GPU, 16-core Neural Engine, 100 GB/s memory bandwidth)
    SET software: = Anaconda Python environment
    // NSGA Parameter Configuration
    SET params: = {
    reference_points_number: = 0.9,
    crossover_probability: = 0.9,
    crossover_distribution_index: = 0.25,
    mutation_probability: = 0.2,
    hypervolume: = [0,0] to [1,1],
    generations: = 500,
    population_size: = 200,
    population_initialization: = "random",
    objectives: = 2
    }
// Define Dataset
    LOAD IoV dataset
    DEFINE features: = all available features in the dataset
    DEFINE labels: = collision class labels
    // Binary Encoding of Feature Subsets
    FOR each chromosome IN population
    INITIALIZE chromosome as binary vector of length = number of features
    1 => feature selected, 0 => feature not selected
    END FOR
```

```
// Feature Selection and Evaluation Loop
    24. FOR algorithm IN {NSGA2, NSGA III}
    25. INITIALIZE population using random chromosomes
    26. FOR generation IN 1 TO params. generations
    27. // Decode Chromosomes and Select Features
    28. FOR each chromosome IN population
    29. SELECT features where chromosome[i] == 1
    30. COMPUTE Gain Ratio for selected features
    31. COMPUTE objectives:
            a. Objective 1: minimize number of selected features
            b. Objective 2: maximize classification accuracy
            c. Objective 3: Minimize computational time
    32. END FOR

// Apply Genetic Operators
    33. PERFORM selection, crossover, and mutation
    34. UPDATE population using NSGA III logic:
    35. IF algorithm == NSGA2:
            a. APPLY crowding distance and non-dominated sorting
    36. ELSE IF algorithm == NSGA III:
            a. APPLY reference-point-based diversity preservation
    37. END IF


    38. END FOR

// Rank and Evaluate Final Feature Subsets
    39. FOR each Pareto-optimal feature subset
    40. FOR classifier IN {SVM, RFC, GNB, DTC, KNN}
    41. TRAIN classifier using selected features
    42. TEST classifier and record accuracy
    43. END FOR
    44. STORE average accuracy for each feature subset
    45. END FOR

// Wrapper-Based Bi-Directional Feature Selection
    46. FOR each chromosome in final population
    47. INIT selected_features: = []
    48. WHILE not converged
    49. PERFORM forward selection using GR and accuracy
    50. IF any existing feature is irrelevant
            a. REMOVE using backward elimination
    51. END IF
    52. UPDATE chromosome accordingly
    53. END WHILE
    54. EVALUATE using classifiers and record performance
    55. END FOR
    56. STORE algorithm performance results
    57. END FOR

// Comparative Analysis
    58. COMPARE NSGA2 vs NSGA3 based on:
                    - Detection Accuracy
                    - Number of selected features
                    - Diversity of selected subsets

// External Validation
    59. COMPARE results with other MOO feature selection methods (e.g., SPEA2, MOPSO)
    60. EVALUATE fairness using same dataset
    61. End
```

Wrapper Method with Bi-directional Feature Selection

The experiment was further extended using the wrapper-based bi-directional feature selection method. Unlike simple forward selection, this approach evaluates the relevance of existing features before new ones are added. If a current feature is deemed non-significant, it is eliminated through backward selection. This dual approach ensures an optimal balance between adding and removing features. NSGA2 and NSGA III algorithms integrate this selection logic, with NSGA2 employing its crowding distance operator to maintain diversity and NSGA III focusing on solution diversity across generations. Ultimately, both algorithms feed their selected feature subsets into the bi-directional wrapper model. Algorithm 1 illustrates the complete process of the proposed hybrid NSGA III framework. To validate the novelty and effectiveness of this method, the results were compared with other established multi-objective feature selection algorithms outside the NSGA family using the same dataset for fairness and consistency, the algorithm is provided in algorithm 2.

Algorithm 2 Wrapper with Bi-directional feature selection

```
function BiDirectionalFeatureSelection(dataset, learning_algorithm, max_iterations):
   // Inputs:
   // - dataset: The full dataset with features and target variable
   // - learning_algorithm: The classifier or regressor used to evaluate feature subsets (e.g., SVM, Decision Tree)
   // - max_iterations: Maximum number of iterations to prevent infinite loops
   // Outputs:
   // - best_feature_set: The optimal subset of features
   // - best_performance: The performance score of the best_feature_set
   // 1. Initialization
   all_features = set (all feature names in dataset)
   selected_features = set () // Start with an empty set
```

```
candidate_features = all_features.copy() // All features are initially candidates for addition
best_feature_set = selected_features.copy()
best_performance = -infinity // Initialize with a very low score
iterations_with_no_improvement = 0
max_no_improvement_iterations = 5 // A stopping condition
// 2. Main Iteration Loop
for i from 1 to max_iterations:
    current_performance = -infinity
    feature_to_add = null
    feature_to_remove = null
    // 3. Forward Step: Find the best feature to ADD
    // This phase aims to greedily add the most impactful feature from the candidates
    if candidate_features is not empty:
        best_forward_gain = -infinity
        for feature in candidate_features:
            temp_features = selected_features + {feature}
            model = train (learning_algorithm, dataset with temp_features)
            performance = evaluate (model, test_data) // Evaluate on a held-out test set or via cross-validation
            if performance > best_forward_gain:
                best_forward_gain = performance
                feature_to_add = feature
        // Add the best feature if it improves performance
        if best_forward_gain > current_performance:
            selected_features.add(feature_to_add)
            candidate_features.remove(feature_to_add)
            current_performance = best_forward_gain
   // 4. Backward Step: Find the worst feature to REMOVE
    // This phase aims to greedily remove the least impactful feature from the selected set
    if selected_features has more than 1 feature:
        best_backward_score = -infinity // A score to track the best removal
        feature_to_remove = null
        for feature in selected_features:
            temp_features = selected_features - {feature}
            model = train (learning_algorithm, dataset with temp_features)
            performance = evaluate (model, test_data)
            // We want to remove the feature whose removal causes the smallest performance drop
            if performance > best_backward_score:
                best_backward_score = performance
                feature_to_remove = feature
        // Remove the worst feature if it improves or maintains performance
        if best_backward_score > current_performance or best_backward_score == current_performance:
            selected_features.remove(feature_to_remove)
            candidate_features.add(feature_to_remove)
            current_performance = best_backward_score
    // 5. Check for Improvement and Update Best Feature Set
    if current_performance > best_performance:
        best_performance = current_performance
        best_feature_set = selected_features.copy()
        iterations_with_no_improvement = 0 // Reset counter on improvement
    else:
        iterations_with_no_improvement = iterations_with_no_improvement + 1
    // 6. Stop Condition
    if iterations_with_no_improvement >= max_no_improvement_iterations:
        break // Exit loop if no improvement for several iterations
// 7. Return Result
return best_feature_set, best_performance
```

III. Results and discussions

Parameter Setting

LSTM

i. Number of Hidden Layers: 3
ii. Number of Neurons at each Hidden Layer: 100
iii. Number of epochs: 100
iv. Activation Function: Relu and Sigmoid
v. Number of steps: 6
vi. Number of Input features: 3, 4, 5 and 6
vii. Number of Dropout: 0

DRNN

i. Number of Hidden Layers: 3
ii. Number of Neurons at each Hidden Layer: 100
iii. Number of epochs: 100
iv. Activation Function: Relu and Sigmoid
v. Number of steps: 6
vi. Number of Input features: 3, 4, 5 and 6
vii. Number of Dropout: 0

GANN

i. Number of Population per Generation: 1000
ii. Number of Generations: 10000
iii. Fitness Function: 1
iv. Mutation Rate: 2%
v. Number of Best Solutions per Generation: 100

vi. Number of Hidden Layers: 1

vii. Number of Neurons at Hidden Layer: 100

viii. Number of epochs: 100

ix. Activation Function: Relu and Sigmoid

x. Number of steps: 6

xi. Number of Input features: 3, 4, 5 and 6

ANN

i. Number of Hidden Layers: 1

ii. Number of Neurons at Hidden Layer: 100

iii. Number of epochs: 100

iv. Activation Function: Relu and Sigmoid

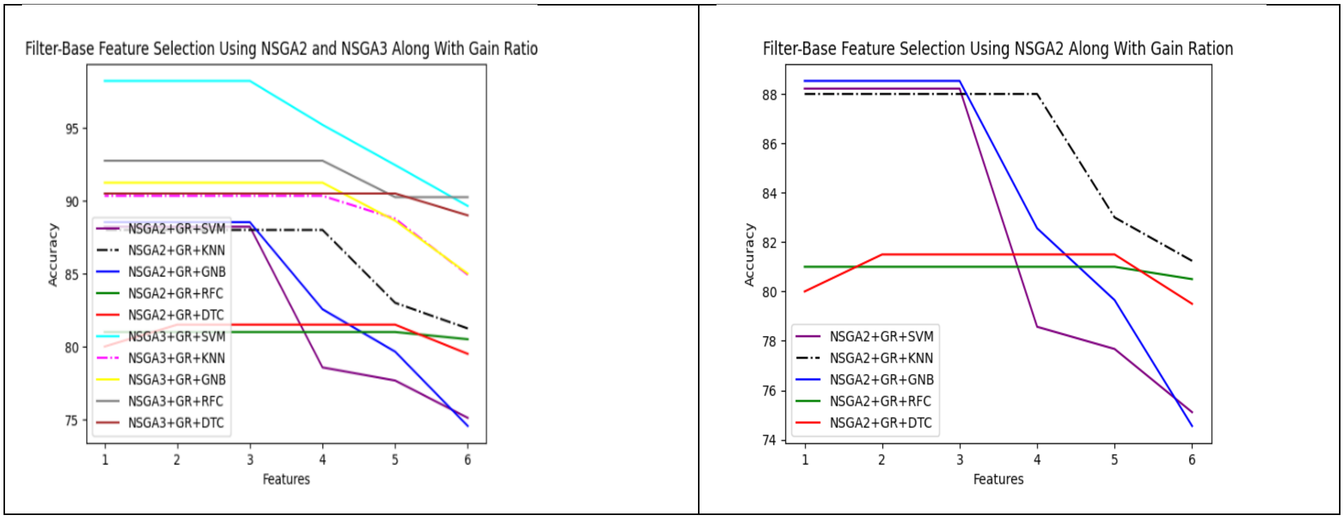
v. Number of steps: 6

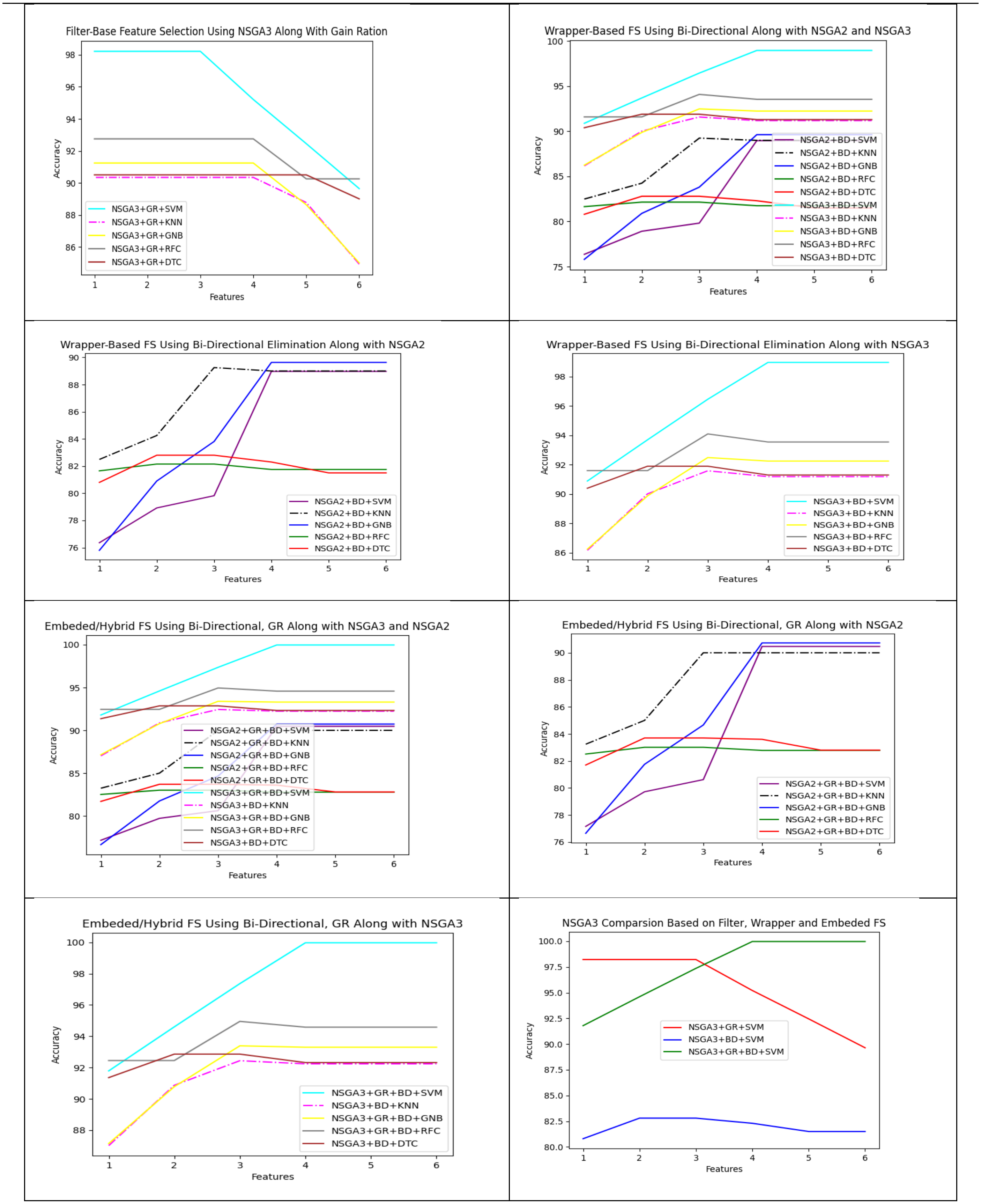
vi. Number of Input features: 3, 4, 5 and 6

The wrapper-based feature selection method employs a bi-directional elimination strategy, also known as step-wise selection, which is an enhancement over simple forward selection. Unlike forward selection, which only considers the inclusion of new features, bi-directional elimination actively evaluates both the addition and removal of features throughout the process. This dual approach ensures that the model remains optimized by continuously reassessing the relevance of all features at each step.

At the core of this method is the integration of forward selection and backward elimination in a single loop. Forward selection begins by identifying features not currently in the model and evaluating whether they should be included based on a predetermined level of statistical significance. Once a feature meets the inclusion criteria, it is added to the model. However, before finalizing the addition, the process checks if any of the already selected features have lost their statistical relevance. Backward elimination plays a complementary role in this combined approach. As each new feature is considered for inclusion, the algorithm also re-evaluates all previously added features. If any of them are found to have become statistically insignificant, they are promptly removed. This dynamic balancing act ensures that the model only retains features that continue to contribute meaningfully to prediction accuracy.

The procedure begins by defining threshold values, typically known as the significance levels for entry and removal commonly set at 0.05 for both. These thresholds represent the criteria by which a feature is judged either worthy of inclusion or subject to elimination. A feature is added to the model only if its statistical p-value falls below the inclusion threshold, and any feature whose p-value rises above the elimination threshold is discarded. Once these thresholds are set, the process iteratively performs a forward selection step, followed by a backward elimination step. This loop continues until no further features meet the criteria for entry or removal. The repeated evaluation ensures that the model evolves into a highly refined and efficient structure, free from redundancy or statistical noise. Ultimately, bi-directional elimination guides the feature selection process toward identifying an optimal subset of features. By merging the strengths of both forward and backward selection, this method helps construct a more parsimonious and interpretable model that maintains high predictive performance while avoiding overfitting.





Filter-Base Feature Selection Using NSGA3 Along With Gain Ration

Wrapper-Based FS Using Bi-Directional Along with NSGA2 and NSGA3

Wrapper-Based FS Using Bi-Directional Elimination Along with NSGA2

Wrapper-Based FS Using Bi-Directional Elimination Along with NSGA3

Embeded/Hybrid FS Using Bi-Directional, GR Along with NSGA3 and NSGA2

Embeded/Hybrid FS Using Bi-Directional, GR Along with NSGA2

Embeded/Hybrid FS Using Bi-Directional, GR Along with NSGA3

NSGA3 Comparsion Based on Filter, Wrapper and Embeded FS

Figure1: Comparing feature subset selection

The analysis of the experimental results, particularly as visualized in Figure 4, provides compelling evidence regarding the efficacy of the wrapper NSGA III + GR in optimizing feature selection and classification performance. The discussion centers on the performance of the wrapper NSGA III + GR, evaluated using a gain ratio-based entropy measure across the dataset. The wrapper NSGA III + GR, when configured, demonstrated a significant capability for feature reduction. Across the datasets, this configuration managed to decrease the total number of features by nearly 70%. Despite this substantial reduction in feature size, the algorithm was able to achieve a classification performance that was superior to the performance attained using the full-length feature set (12). This outcome highlights the potential of the wrapper NSGA III + GR approach to distill large feature spaces into a more manageable and computationally efficient form without sacrificing predictive power.

The wrapper NSGA III + GR consistently performed well. Its primary strength lies in its ability to find single solutions that not only utilize a significantly smaller number of features but also achieve a higher level of performance compared to the complete feature set. This suggests a highly efficient search mechanism for identifying a minimal, yet highly predictive, feature subset. The wrapper NSGA III + GR achieved a better classification error rate. This variant's focus appears to be on maximizing classification accuracy, even while performing a drastic feature reduction. For instance, the feature size was reduced by exactly 50%, from 12 features to 6, which concurrently resulted in a significant drop in the error rate from to a more favorable level. The wrapper NSGA III + GR consistently outperformed baseline feature selection approaches in terms of error rate, with the notable exception of the In every dataset evaluated, wrapper NSGA III + GR selected fewer features This reinforces the notion that wrapper NSGA III + GR is more effective at identifying and isolating the most critical features for a given for detecting vehicle collision in IoV.

The experimental results reveal that integrating optimization strategies with machine learning classifiers significantly enhances the performance of collision detection models. Specifically, the use of advanced multi-objective optimization techniques such as NSGA2 and NSGA III in combination with feature refinement methods leads to clear improvements over the baseline configurations where all features are used indiscriminately as in the case of original LSTM. Among the evaluated classifiers, Support Vector Machines (SVM) and Random Forest (RF) consistently stood out in terms of performance, especially when used in tandem with hybrid optimization strategies. These classifiers are inherently capable of capturing complex patterns in high-dimensional spaces, and when aided by appropriate feature selection, they can effectively model the dynamic and intricate nature of vehicular interactions. An interesting observation is that models trained using all available features did not always yield the highest accuracy. This suggests the presence of redundant or less-informative features within the dataset. In contrast, models that utilized a carefully selected subset of features primarily those related to velocity, distance between vehicles, braking capability, etc. tended to produce more reliable predictions across all classifiers. The optimization technique that incorporated NSGA III and Bi-Directional search combined with a guided refinement approach consistently demonstrated the most robust performance. This method appears particularly effective in identifying and preserving features that contribute meaningfully to the learning process, while filtering out those that introduce noise or unnecessary complexity.

Feature subsets comprising three to four attributes were found to be sufficient in most cases to drive high-performing models. These typically included vehicle speed, the gap between adjacent vehicles, and the vehicle’s braking response. Occasionally, contextual features such as the number of lanes or the nature of the driving environment also contributed, depending on the classifier and optimization method used. The comparative analysis further highlights that different classifiers respond uniquely to various feature selection strategies. For instance, while Naive Bayes benefited significantly from feature reduction, Decision Trees showed moderate gains. On the other hand, ensemble methods like Random Forest maintained strong performance even with the full set of features but still showed improvements when irrelevant features were excluded. The integration of Bi-Directional search mechanisms allowed the models to explore both forward and backward feature spaces, enabling a more thorough selection process. This bidirectional exploration appears to enhance the model’s ability to generalize across varying traffic scenarios, making it particularly valuable in real-time vehicular networks where environmental conditions and driver behavior can change rapidly.

Furthermore, the results demonstrate the merit of combining evolutionary algorithms with data-driven refinement techniques. The hybridization of global search (from NSGA III) and local search (from guided refinement) creates a balance that leverages both exploration and exploitation in the feature space, which is essential for developing robust and adaptable predictive models. The consistency in performance improvements across all classifiers when optimized features are used reflects the importance of feature engineering in intelligent transportation systems. It reinforces the notion that more data is not always better; rather, the relevance and quality of the features play a more critical role in determining the success of predictive modeling. In conclusion, this study underscores the effectiveness of hybrid optimization strategies in improving collision detection accuracy in vehicular environments. By narrowing down to the most informative features and aligning them with the strengths of specific classifiers, the models not only become more accurate but also computationally efficient, laying a solid foundation for scalable and real-time deployment in IoV systems.

Table 1a Vehicle collision feature subset selected by wrapper NSGA III + GR

Method	SVM		KNN		Naive Bayes		Random Forest		Decision Tree	
	Selected Features	**Accuracy**	**Selected features**	**Accuracy**	**Selected features**	**Accuracy**	**Selected features**	**Accuracy**	**Selected features**	**Accuracy**
NSGA2	All Features (6)	75.11	All features	81.24	All features	74.55	All features	80.51	All features	79.5

	3	88.22	4	88.00	3	88.51	4	81.00	4	81.5
NSGA3 +GR	All Features	79.64	All features	64.92	All features	85.00	All features	80.25	All features	89.0
	3	89.22	4	80.34	4	87.28	4	82.75	5	90.5
Wrapper NSGA3 + GR	All Features	80.2	All features	89.75	All features	89.55	All features	85.65	All features	82.50
	6	78.44	5	75.00	4	87.00	5	76.00	4	92.50
NSGA3	All Features	83.25	All features	71.05	All features (6)	81.75	All features	82.25	All features	80.65
	3	78.97	4	81.59	3	82.25	3	83.55	3	91.30
NSGA2 +Bi	All Features (12)	82.55	All features	70.75	All features	81.05	All features	89.12	All features	89.77
	5	83.00	4	81.75	3	81.75	3	80.05	3	90.50
Wrapper NSGA3 + GR	All Features (12)	79.25	All features (6)	72.05	All features (12)	83.15	All features (12)	84.25	All features (12)	70.65
	6	96.89	7	72.24	7	83.30	8	89.58	3	82.32

Table 1 The summary of 80-20 splits compared to baseline algorithm for collision detection

Model	No. of Inputs	AES (Seconds)	MSE	RMSE	Accuracy (%)
LSTM	12	279s (15ms/step)	0.1284	0.3583	83
DRNN	12	141s (4ms/step)	0.1329	0.3645	78
DRNN	6	127s (7ms/step)	0.0085	0.0921	99
GANN	12	100s (6ms/step)	0.1817	0.4262	75
GANN	6	10s (12ms/step)	0.1598	0.3997	79
ANN	12	13s (1ms/step)	0.2241	0.4733	65
ANN	6	38s (2ms/step)	0.2488	0.4987	65
NSGA-III + Deep LSTM-RD	**6**	**5s (1ms/step)**	**0.000898**	**0.0299**	**99**

Table 1 compare the results obtained from using 6 input features versus 12 input features for vehicle collision detection in the IoV, it is evident that models performed significantly better with 6 inputs. The NSGA-III + Deep LSTM-RD, DRNN, and GANN models demonstrated higher accuracy and reduced error rates when the features were reduced. This improvement highlights the importance of collision-influencing parameters, as the models were better able to learn complex patterns relevant to predicting collisions with less input features. Among the models evaluated, the ANN showed limited improvement in performance, even when the number of input features reduces. This underperformance is attributed to the shallow nature of the ANN, which lacks the depth required to effectively capture the underlying patterns in the expanded feature space. In contrast, the Genetic Algorithm component in the GANN architecture played a critical role in enhancing the performance of the ANN by optimizing weights and structure, thereby yielding better results than the ANN alone. In terms of the rate of vehicle collision detection per unit time, or AES, all models experienced an increase as the number of input features reduce from 12 to 6. This may appear counterintuitive but is likely due to the models' enhanced ability to generalize from richer input data, leading to quicker and more confident predictions despite the added computational complexity. However, this improvement was not uniform across all models.

A closer analysis of model performance based on key metrics such as MSE, RMSE, and accuracy revealed a consistent ranking. The NSGA-III + Deep LSTM-RD model emerged as the best performer, showing superior precision and reliability in vehicle collision detection. This hierarchy underscores the importance of both model architecture and input dimensionality in achieving effective predictions.

Interestingly, when examining AES specifically, NSGA-III + Deep LSTM-RD achieved the lowest (fastest) execution time. This indicates a trade-off between accuracy and computational efficiency. While deeper models like the LSTM-based architecture offered the highest detection performance, they demanded more processing time, whereas models like GANN and ANN offered quicker predictions at the expense of some accuracy.

Table 2: The findings for 60 – 40 splits comparing the NSGA-III + Deep LSTM-RD with baseline algorithms in detecting vehicle collision

Model	**No. of Inputs**	**AES (Seconds)**	**MSE**	**RMSE**	**Accuracy (%)**
LSTM	12	150s 12ms/step	0.1080	0.3286	83
DRNN	12	290s 23ms/step	0.1152	0.3394	81
DRNN	6	28s 22ms/step	0.0000165	0.00406	100
GANN	12	200s 6ms/step	0.1817	0.4262	75
GANN	6	10s 12ms/step	0.1598	0.3997	79
ANN	12	214s 3ms/step	0.2245	0.4738	65
ANN	6	40s 3ms/step	0.2244	0.4737	65
NSGA-III + Deep LSTM-RD	**6**	**284s (1ms/step)**	**0.000898**	**0.0299**	**99**

Increasing the number of training epochs from 50 to 100 is expected to lead to performance improvements across all the evaluated models. Extended training allows the models to better adjust their weights and biases, enabling them to learn more complex patterns within the data. Consequently, the results presented in Table 2 improve when the models are trained over a longer duration. Among the models, the NSGA-III + Deep LSTM-RD exhibits the most notable improvement when the number of epochs is increased. It shows rapid enhancement in terms of accuracy, reduction in error, and improvement in computational efficiency. This demonstrates the model's ability to effectively leverage deeper training iterations to refine its prediction of vehicle collision in IoV, the GANN and ANN models display a limited response to the increased number of epochs. While there is a slight improvement in their Average Execution Speed (AES), their accuracy and error rates remain largely unchanged. This suggests that these models may have already reached a plateau in learning or lack the architectural complexity to benefit significantly from additional training. The minimal gains observed in GANN and ANN further highlight the importance of model depth and optimization mechanisms. While the genetic algorithm in GANN provides some advantage over the basic ANN, it still falls short when compared to more sophisticated models like Deep LSTM-RD. This indicates that advanced architectures are better suited for complex tasks such as vehicle collision detection in IoV environments.

A key strength of the NSGA-III + Deep LSTM-RD model is its ability to identify and utilize the most relevant subset of features for collision prediction. This feature selection process helps the model focus on the most informative data, which not only enhances detection accuracy but also reduces the computational time. As a result of this efficient feature selection, the NSGA-III + Deep LSTM-RD model demonstrates a significant reduction in processing time. Despite the increased number of epochs, the model maintains fast execution, reflecting its ability to streamline computation without compromising performance. Overall, the findings affirm that extending the training period benefits all models to varying degrees, with the NSGA-III + Deep LSTM-RD leading in terms of both efficiency and predictive capability. Its combination of deep learning with evolutionary optimization proves particularly effective in the context of real-time vehicle collision detection in IoV systems.

Table 3: The summary of comparison analysis for 70 – 30 splits in detecting collision in IoV

Model	No. of Inputs	AES (Seconds)	MSE	RMSE	Accuracy (%)
LSTM	3	35s 17ms/step	0.1025	0.3201	83
DRNN	3	102s 7ms/step	0.1341	0.3661	78
DRNN	6	88s 6ms/step	0.2260	0.4753	65
GANN	3	40s 6ms/step	0.1817	0.4262	75
GANN	6	50s 12ms/step	0.1860	0.4312	77
ANN	3	49s 1ms/step	0.2244	0.4737	65
ANN	6	50s 3ms/step	0.2244	0.4737	65
NSGA-III + Deep LSTM-RD	**6**	**265s 18ms/step**	**0.0000138**	**0.0037**	**98**

Table 3 indicate that the NSGA-III + Deep LSTM-RD model demonstrated superior performance compared to the baseline algorithms, including DRNN, GANN, LSTM, and ANN. Its integration of multi-objective optimization and deep learning enabled it to consistently outperform the others across various performance metrics. This model effectively captured complex temporal patterns in the data, leading to higher accuracy and lower error rates in vehicle collision detection within the IoV environment. One of the key strengths of the NSGA-III + Deep LSTM-RD lies in its ability to reduce the number of collision detection features from 12 to 6 without compromising performance. In fact, it maintained and even improved upon its prior results with fewer input features. This reduction significantly contributed to decreased computational time, proving the model's efficiency in selecting the most relevant features for accurate and timely detection vehicle collision.

When the data partition ratio was altered from 60-40 to 70-30, a variation in performance was observed among the models. Specifically, the LSTM model benefited from this change, showing a reduction in error. In contrast, the DRNN experienced a decline in performance, as indicated by an increase in error. Meanwhile, both the GANN and ANN models exhibited no change in error rates, suggesting limited sensitivity to the partition ratio adjustment. In terms of accuracy, the LSTM maintained its performance despite the altered partition ratio. However, the DRNN's accuracy further deteriorated, indicating instability under this configuration. The GANN model retained its accuracy with 12 input features but underperformed when the input features were reduced to 6, showing a dependency on higher dimensional input. On the other hand, the ANN's accuracy remained unchanged under all configurations, reinforcing its limited learning capacity in this context.

The increase in the partition ratio to 70-30 also influenced the models' Average Execution Speed (AES). Notably, LSTM, DRNN, and ANN all experienced improvements in AES, reflecting enhanced computational efficiency due to the increased training set size. However, GANN's AES remained constant, possibly due to the additional overhead introduced by the genetic algorithm component, which offset any potential gains from the larger training data.

These variations highlight how different architectures respond to changes in data availability and input complexity. While deep learning models such as deep LSTM can adapt and improve with more training data, simpler or hybrid models may not always benefit, and in some cases, performance can degrade, especially when essential features are removed or the model lacks sufficient depth to generalize effectively. The NSGA-III + Deep LSTM-RD continues to stand out as the most robust and adaptable model for vehicle collision detection in IoV systems. Its ability to maintain performance with fewer features, adapt well to different data partitions, and deliver high accuracy with lower error and computational time underscores its suitability for real-time intelligent transportation applications.

Table 5: The 50 – 50 slip ratios for the detection of vehicle collision in IoV

Model	**No. of Inputs**	**AES (Seconds)**	**MSE**	**RMSE**	**Accuracy (%)**
LSTM	3	18s 17ms/step	0.0988	0.3143	84
LSTM					
DRNN	3	23s 22ms/step	0.1137	0.3371	82
DRNN	6	23s 22ms/step	0.00101	0.31	65
GANN	3	0s 6ms/step	0.1817	0.4262	75
GANN	6	0s 12ms/step	0.1860	0.4312	77
ANN	3	4s 4ms/step	0.2240	0.4732	66
ANN	6	4s 3ms/step	0.2240	0.4732	66
NSGA-III + Deep LSTM-RD	**6**	**18s 18ms/step**	**0.0000096**	**0.0030**	**97**

Table 4: Present the performance of the NSGA-III + Deep LSTM-RD model under the 50-50 data partition ratio was unexpectedly superior to that observed with the 80-20, 70-30, and 60-40 partition ratios. This outcome appears analytically counterintuitive, as models are typically expected to perform better with more training data. This raises a critical question: what could possibly explain this performance? To address this question, it is important to first understand the implications of a 50-50 data partition. In this case, half of the available dataset is used for training the model, while the remaining half is used for testing or prediction. Typically, having less training data might be seen as a limitation, but under certain conditions, it may inadvertently simplify the learning process for the model.

One plausible explanation is that the first 50% of the dataset used for training may consist of tuples with similar or repetitive patterns. These consistent data sequences could allow the Deep LSTM component to effectively learn temporal dependencies during forward and backward propagation. As a result, the weight updates during training become more stable and predictable, leading to faster convergence and lower error rates. The NSGA-III algorithm, which assists in feature selection and optimization, may also benefit from the uniformity in the training data. With more homogenous input patterns, the multi-objective optimization process may more easily identify dominant features that are relevant for vehicle collision detection. This synergy between optimized feature selection and efficient learning could contribute to the exceptional performance observed with the 50-50 split.

In contrast, increasing the training ratio to 60-40, 70-30, or 80-20 introduces a broader and more diverse set of training tuples. While this typically supports better generalization, in this case, the increased randomness and complexity may have challenged the model. The introduction of varied and less predictable input patterns likely required more epochs or adaptive strategies for the model to learn effectively, which may not have been adequately provided during training. Consequently, the learning rate might have decreased, and the model may have struggled to maintain low error margins. The LSTM component, sensitive to sequence continuity, could find it harder to learn consistent temporal relationships in such varied data. Similarly, NSGA-III's optimization may face difficulties identifying consistently dominant features in a more heterogeneous training set, leading to a slight performance drop. Generally, the seemingly anomalous superior performance at the 50-50 partition ratio is likely due to the reduced complexity and greater consistency in the training data, which simplified the learning process. While larger training sets are generally advantageous, the quality and uniformity of the data matter significantly. These findings suggest that beyond the volume of training data, its internal structure and pattern distribution can play a critical role in model performance.

To practically investigate the unexpected performance anomaly observed with the 50-50 data partition ratio, a deliberate adjustment was made to the original dataset, which comprised 206,775 records. Specifically, the first 77,830 data entries believed to have significantly influenced the models' performance were removed. These entries had previously formed part of the training set across all models when fine-tuning their parameters and were also included in the 50% training portion under scrutiny. By removing this substantial portion of the dataset, the total remaining data was reduced to 128,945 entries. This adjustment effectively eliminates the suspected repetitive or pattern-heavy segments that may have simplified learning for the deep LSTM and NSGA-III components of model, potentially explaining the earlier anomaly. The intent is to test the models' performance under more randomized and possibly more representative training conditions. This dual-setup will help determine whether the previously observed performance advantage under the 50-50 ratio persists, or whether it was indeed a result of the early training data structure. The results will provide clearer insights into how the nature and position of training data influence model accuracy and generalization.

Possible reasons why the dynamic LSTM outperform the baseline algorithms

The model is a hybrid that combines a genetic algorithm (NSGA-III) with a deep learning network (Deep LSTM). This approach leverages the strengths of both methods, as hybrid models often outperform individual algorithms in predictive tasks. The evolutionary algorithm (NSGA-III) is efficient at exploring the feature space to optimize multiple objectives, while the Deep LSTM is well-suited for capturing temporal dependencies in sequential data, which is ideal for predicting collision likelihood. The model's ability to identify and utilize the most relevant subset of features is a key strength. It successfully reduced the number of collision detection features from 12 to 6, which not only maintained or improved performance but also significantly decreased computational time. The other models either showed limited improvement with reduced features (like ANN) or, in the case of GANN, exhibited a dependence on a higher-dimensional input. The NSGA-III component, through its multi-objective optimization, helps the model focus on the most informative data, enhancing detection accuracy. The "Deep" nature of the LSTM component is crucial. Unlike the shallow ANN, which lacks the depth to effectively capture underlying patterns, the Deep LSTM-RD model has the architectural complexity to benefit significantly from training and learn more complex patterns. The NSGA-III + Deep LSTM-RD model consistently demonstrated superior precision and reliability in vehicle collision detection across different data partition ratios (50-50, 60-40, 70-30, and 80-20). While other models like DRNN and GANN showed instability or no change in performance under different data configurations, the NSGA-III + Deep LSTM-RD model proved to be the most robust and adaptable. The model's efficient feature selection process contributes to its fast execution, maintaining a fast Average Execution Speed (AES) despite the increased number of training epochs. This makes it particularly suitable for real-time applications where computational efficiency is a primary concern. The hybridization of global search (from NSGA-III) and local search (from guided refinement) creates a balance that leverages both exploration and exploitation in the feature space. This results in a robust and adaptable predictive model.

Implication for theory and practice

The study provides a theoretical basis for the synergy between evolutionary algorithms and deep learning models for complex, real-time problems. Specifically, it demonstrates that combining NSGA-III, a multi-objective genetic algorithm, with a Deep LSTM network can lead to superior performance in feature selection and predictive accuracy compared to using either algorithm in isolation. This challenges the reliance on static LSTM that lack the ability to select features noted in previous literature and validates the theoretical superiority of hybrid approaches. The paper formalizes feature selection for vehicle collision detection as a multi-objective optimization problem. The objectives are to minimize the number of selected features while maximizing classification accuracy and minimizing computational time. This theoretical framing is more holistic than methods that only rely on correlation coefficients, which have been a common approach in existing studies. The use of NSGA-III to address these competing objectives concurrently provides a robust theoretical model for handling similar complex optimization tasks in other domains. The research introduces a crucial theoretical finding that challenges the common assumption that more training data always leads to better model performance. The superior results observed with the 50-50 data split, compared to larger training ratios (60-40, 70-30, and 80-20), suggest that the internal structure, consistency, and quality of the training data can sometimes outweigh its sheer volume. This finding has significant implications for machine learning theory, suggesting a need to re-evaluate how data is partitioned and preprocessed for optimal training, moving beyond simple volume-based assumptions. The study reinforces the theoretical principle that a model's architectural depth is crucial for learning complex patterns. The Deep LSTM-based models consistently outperformed shallower models like the basic ANN, which showed limited improvement even with a reduced number of input features. The research theoretically attributes this underperformance to the ANN's lack of depth, which prevents it from effectively capturing underlying patterns in the expanded feature space.

The primary practical implication is the development of a more accurate and reliable predictive classifier for vehicle collisions. By using the NSGA-III + Deep LSTM + RD model, developers can create collision alarm systems that empower both human and autonomous vehicles to take preventive action with higher confidence. The high accuracy and low error rates achieved by the model make it a strong candidate for real-world deployment. The research demonstrates that effective feature selection can drastically reduce computational time without compromising accuracy. The NSGA-III + Deep LSTM-RD model, with its ability to reduce features from 12 to 6, showed a significant reduction in processing time and the fastest execution time (AES) compared to other models. This is a critical factor for IoV applications, where real-time responsiveness and low latency are essential for collision avoidance. The findings provide a practical blueprint for IoV data management. Instead of indiscriminately using all available data, which can include redundant or noisy features, practitioners can use multi-objective feature selection to identify the most informative data streams. This not only improves model performance but also reduces the data transmission and processing load on network infrastructure, a key consideration for edge computing deployments in IoV. The research provides clear guidance for practitioners on which model architectures and feature selection strategies are most effective. The consistent strong performance of SVM and Random Forest, particularly when combined with hybrid optimization, provides a practical starting point for engineers designing IoV-based predictive models. The study also offers insights into the importance of model depth and the limitations of simpler models like ANN in complex IoV environments. The practical application of the bi-directional feature selection method, integrated within the NSGA framework, offers a powerful tool for practitioners. This approach ensures an optimal balance between adding and removing features, creating a more refined and efficient model that avoids overfitting and maintains high predictive performance. This method is a valuable addition to the toolkit for anyone working on feature engineering for real-world machine learning applications.

IV. Conclusion and Future Direction

This study highlights the effectiveness of dynamic optimization strategies in improving the accuracy of collision detection models within vehicular environments. The research reinforces that the relevance and quality of features are more critical than the sheer volume of data. The proposed hybrid model, which integrates a multi-objective optimization technique (NSGA-III) with deep learning (Deep LSTM), consistently demonstrated superior performance. By effectively identifying and utilizing a small subset of key features such as vehicle velocity, inter-vehicle distance, and braking capability, the model not only achieved high accuracy but also became more computationally efficient, making it suitable for real-time deployment in IoV systems. The study also revealed that while larger training datasets are generally beneficial for generalization, the internal structure and consistency of the data can significantly influence model performance, as evidenced by the unexpected superior results with a 50-50 data partition ratio. For future research, the following avenues could be explored: Investigate the performance of the hybrid model using real-world IoV datasets rather than simulated data. This would provide a more robust validation of the model's performance under authentic and unpredictable traffic conditions. Explore additional hybrid models that combine other evolutionary algorithms with different deep learning architectures to further optimize feature selection and prediction accuracy. Extend the model to incorporate and evaluate the impact of a wider range of environmental factors, such as diverse weather conditions (e.g., fog, heavy rain) and road surface variables, which were noted as a limitation in some previous studies.

References

1. Aboulola, O., Khayyat, M., Al-Harbi, B., Muthanna, M. S. A., Muthanna, A., Fasihuddin, H., &Alsulami, M. H. (2021). Multimodal feature-assisted continuous driver behavior analysis and solving for edge-enabled internet of connected vehicles using deep learning. *Applied Sciences*, *11*(21), 10462.
2. Almutairi, M. S., Almutairi, K., &Chiroma, H. (2023b). Selecting Features That Influence Vehicle Collisions in the Internet of Vehicles Based on a Multi-Objective Hybrid Bi-Directional NSGA-III. *Applied Sciences*, *13*(4), 2064.
3. Almutairi, M. S., Almutairi, K., & Chiroma, H. (2023a). Hybrid of deep recurrent network and long short term memory for rear-end collision detection in fog based internet of vehicles. *Expert Systems with Applications*, *213*, 119033.
4. Akcay, S., Kundegorski, M. E., Willcocks, C. G., & Breckon, T. P. (2018). Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. *IEEE transactions on information forensics and security*, *13*(9), 2203-2215.
5. Alzubaidi, F., Makuluni, P., Clark, S. R., Lie, J. E., Mostaghimi, P., & Armstrong, R. T. (2022). Automatic fracture detection and characterization from unwrapped drill-core images using mask R–CNN. *Journal of Petroleum Science and Engineering*, *208*, 109471.
6. Baisa, N. L. (2019). *Online multi-object visual tracking using a GM-PHD filter with deep appearance learning.* Paper presented at the 2019 22th international conference on information fusion (FUSION).
7. Baek, M., Jeong, D., Choi, D., & Lee, S. (2020). Vehicle trajectory prediction and collision warning via fusion of multisensors and wireless vehicular communications. Sensors, 20(1), 288.

8. Brik, B., & Ksentini, A. (2021). Toward optimal MEC resource dimensioning for a vehicle collision avoidance system: A deep learning approach. IEEE Network, 35(3), 74–80.
9. Bills, T. C. (2019). The great transformation: the future of the data-driven transportation workforce. In *Empowering the New Mobility Workforce* (pp. 31-62). Elsevier.
10. Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-based systems, 86,* 33-45.
11. Boudaoud, L. B., Maussang, F., Garello, R., & Chevallier, A. (2019). *Marine bird detection based on deep learning using high-resolution aerial images.* Paper presented at the OCEANS 2019-Marseille.
12. Cao, Y., Niu, X., & Dou, Y. (2016). *Region-based convolutional neural networks for object detection in very high resolution remote sensing images.* Paper presented at the 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD).
13. Chandan, G., Jain, A., & Jain, H. (2018). *Real time object detection and tracking using Deep Learning and OpenCV.* Paper presented at the 2018 international conference on inventive research in computing applications (ICIRCA).
14. Chen, M. S., Hwang, C. P., Ho, T. Y., Wang, H. F., Shih, C. M., Chen, H. Y., & Liu, W. K. (2019). Driving behaviors analysis based on feature selection and statistical approach: a preliminary study. *The Journal of Supercomputing*, *75*, 2007-2026.
15. Chang, W. J., Chen, L. B., & Su, K. Y. (2019). DeepCrash: A deep learning-based Internet of vehicles system for head-on and single-vehicle accident detection with emergency notification. IEEE Access, 7, 148163–148175.
16. Chang, C. C., Ooi, Y. M., & Sieh, B. H. (2021). IoV-based collision avoidance architecture using machine learning prediction. IEEE Access, 9, 115497–115505.
17. Chen, L. B., Su, K. Y., Mo, Y. C., Chang, W. J., Hu, W. W., Tang, J. J., & Yu, C. T. (2018, September). An implementation of deep learning based IoV system for traffic accident collisions detection with an emergency alert mechanism. In 2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin) (pp. 1-2). IEEE.
18. Chen, C., et al. (2018). A rear-end collision prediction scheme based on deep learning in the Internet of Vehicles. Journal of Parallel and Distributed Computing, 117, 192–204.
19. Chiroma, H., Ezugwu, A. E., Jauro, F., Al-Garadi, M. A., Abdullahi, I. N., & Shuib, L. (2020). Early survey with bibliometric analysis on machine learning approaches in controlling COVID-19 outbreaks. PeerJ Computer Science, 6, e313.
20. Colaboratory, G. (2018). “Wecome To Colaboratory.” 2021, from https://colab.research. google.com.
21. Cui, Z., Chang, Y., Zhang, J., Cai, X., & Zhang, W. (2019). Improved NSGA-III with selection-and-elimination operator. *Swarm and Evolutionary Computation*, *49*, 23-33.
22. Chen, J., Liu, Z., Wang, H., & Liu, K. (2017). *High-speed railway catenary components detection using the cascaded convolutional neural networks.* Paper presented at the 2017 IEEE International Conference on Imaging Systems and Techniques (IST).
23. Cheng, B., Li, Z., Wu, Q., Li, B., Yang, H., Qing, L., & Qi, B. (2019). Multi-class objects detection method in remote sensing image based on direct feedback control for convolutional neural network. *IEEE Access, 7*, 144691-144709.
24. Cheng, G., Han, J., Zhou, P., & Guo, L. (2014). Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS journal of photogrammetry and remote sensing*, *98*, 119-132.
25. Cheng, G., Si, Y., Hong, H., Yao, X., & Guo, L. (2020). Cross-scale feature fusion for object detection in optical remote sensing images. *IEEE Geoscience and Remote Sensing Letters, 18*(3), 431-435.
26. Cheng, G., Zhou, P., & Han, J. (2016a). Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(12), 7405-7415.
27. Cheng, G., Zhou, P., & Han, J. (2016b). *Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
28. Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H., & Hu, S.-M. (2014). Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence, 37*(3), 569-582.
29. Chiroma, H., Abdulhamid, S. I. M., Hashem, I. A., Adewole, K. S., Ezugwu, A. E., Abubakar, S., & Shuib, L. (2021). Deep learning-based big data analytics for internet of vehicles: taxonomy, challenges, and research directions. *Mathematical Problems in Engineering*, *2021.*
30. Deng, J., Bi, H., Yin, Y., Lu, X., & Liang, W. (2021). *Sparse SAR Image Based Automatic Target Recognition by YOLO Network.* Paper presented at the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS.
31. Dong, Z., Wang, M., Wang, Y., Zhu, Y., & Zhang, Z. (2019). Object detection in high resolution remote sensing imagery based on convolutional neural networks with suitable object scale features. *IEEE Transactions on Geoscience and Remote Sensing, 58*(3), 2104-2114.
32. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). *CARLA: An open urban driving simulator.* Paper presented at the Conference on robot learning.
33. Fang, F., Li, L., Zhu, H., & Lim, J.-H. (2019). Combining faster R-CNN and model-driven clustering for elongated object detection. *IEEE Transactions on Image Processing, 29*, 2052-2065.

34. Feki, M. A., et al. (2013). “The internet of things: the next technological revolution.” Computer(2): 24-25. Fong, S., et al. (2018). How meta-heuristic algorithms contribute to deep learning in the hype of big data analytics. Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, Springer: 3-25.
35. Fu, K., Chang, Z., Zhang, Y., Xu, G., Zhang, K., & Sun, X. (2020). Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing, 161*, 294-308.
36. Fayyad, J., Jaradat, M. A., Gruyer, D., & Najjaran, H. (2020). Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors, 20*(15), 4220.
37. Hammedi, W., Brik, B., & Senouci, S. M. (2022). Toward optimal MEC-based collision avoidance system for cooperative inland vessels: A federated deep learning approach. IEEE Transactions on Intelligent Transportation Systems.
38. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735–1780.
39. Geiger, A., Lenz, P., & Urtasun, R. (2012). *Are we ready for autonomous driving? the kitti vision benchmark suite.* Paper presented at the 2012 IEEE conference on computer vision and pattern recognition.
40. Guo, W., Yang, W., Zhang, H., & Hua, G. (2018). Geospatial object detection in high resolution satellite images based on multi-scale convolutional neural network. *Remote Sensing, 10*(1), 131.
41. Han, J., Zhang, D., Cheng, G., Guo, L., & Ren, J. (2014). Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Transactions on Geoscience and Remote Sensing, 53*(6), 3325-3337.
42. Han, J., Zhou, P., Zhang, D., Cheng, G., Guo, L., Liu, Z., . . . Wu, J. (2014). Efficient, simultaneous detection of multi-class geospatial targets based on visual saliency modeling and discriminative learning of sparse coding. *ISPRS journal of photogrammetry and remote sensing, 89*, 37-48.
43. Han, Y., Ma, S., Xu, Y., He, L., Li, S., & Zhu, M. (2020). Effective Complex Airport Object Detection in Remote Sensing Images Based on Improved End-to-End Convolutional Neural Network. *IEEE Access, 8*, 172652-172663.
44. Hashemi, H., & Abdelghany, K. (2018). End-to-end deep learning methodology for real-time traffic network management. *Computer-Aided Civil and Infrastructure Engineering*, *33*(10), 849-863.
45. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780.
46. Hou, B., Li, J., Zhang, X., Wang, S., & Jiao, L. (2019). *Object Detection and Trcacking Based on Convolutional Neural Networks for High-Resolution Optical Remote Sensing Video.* Paper presented at the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium.
47. Hu, Y., Li, X., Zhou, N., Yang, L., Peng, L., & Xiao, S. (2019). A sample update-based convolutional neural network framework for object detection in large-area remote sensing images. *IEEE Geoscience and Remote Sensing Letters, 16*(6), 947-951.
48. Ishibuchi, H., Imada, R., Setoguchi, Y., & Nojima, Y. (2016, July). Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3045-3052). IEEE.
49. Islam, A., Hossan, M. T., & Jang, Y. M. (2018). Convolutional neural networkscheme–based optical camera communication system for intelligent Internet of vehicles. *International journal of distributed sensor networks*, *14*(4), 1550147718770153.
50. Jaramillo-Avila, U., & Anderson, S. R. (2019). *Foveated image processing for faster object detection and recognition in embedded systems using deep convolutional neural networks.* Paper presented at the Conference on Biomimetic and Biohybrid Systems.
51. Joshi, A. V. (2019). Machine Learning and Artificial Intelligence. Springer.
52. Kaiwartya, O., et al. (2016). Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects. IEEE Access, 4, 5356–5373.
53. Kim, S. H., et al. (1999). An experimental investigation of a CW/CA system for automobiles. SAE Technical Paper.
54. Khryashchev, V., Ostrovskaya, A., Pavlov, V., & Semenov, A. (2018). *Optimization of convolutional neural network for object recognition on satellite images.* Paper presented at the 2018Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO).
55. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*, 1097-1105.
56. Lei, J., Luo, X., Fang, L., Wang, M., & Gu, Y. (2020). Region-enhanced convolutional neural network for object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing, 58*(8), 5693-5702.
57. Li, C.-j., Qu, Z., Wang, S.-y., & Liu, L. (2021). A method of cross-layer fusion multi-object detection and recognition based on improved faster R-CNN model in complex traffic environment. *Pattern Recognition Letters, 145*, 127-134.
58. Li, N., & Wang, Z. (2020). *Hyperspectral Image Ship Detection Based Upon Two-Channel Convolutional Neural Network and Transfer Learning.* Paper presented at the 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP).
59. Li, Z., Dong, M., Wen, S., Hu, X., Zhou, P., & Zeng, Z. (2019). CLU-CNNs: Object detection for medical images. *Neurocomputing, 350*, 53-59.

60. Lima, M., Silva, R., de Souza Mendes, F. L., de Carvalho, L. R., Araujo, A., & de Barros Vidal, F. (2020, November). Inferring about fraudulent collusion risk on Brazilian public works contracts in official texts using a Bi-LSTM approach. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 1580-1588)
61. Liu, Y., Liu, B., & Chen, Y. (2019). *Research on Image Recognition of Supermarket Commodity Based on Convolutional Neural Network*. Paper presented at the 2019 12th International Symposium on Computational Intelligence and Design (ISCID).
62. Liu, Y., Zhang, Z., Zhong, R., Chen, D., Ke, Y., Peethambaran, J., . . . Sun, L. (2018). Multilevel building detection framework in remote sensing images based on convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11*(10), 3688-3700.
63. Labriji, I., et al. (2021). Mobility aware and dynamic migration of MEC services for the Internet of Vehicles. IEEE Transactions on Network and Service Management, 18(1), 570–584.
64. Lin, T., et al. (1998). How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies. Neural Networks, 11(5), 861–868.
65. Liu, J. (2008). “Wavelet Basis Function Neural Networks for Sequential Learning.” IEEE Access.
66. Loh, D. R., Yong, W. X., Yapeter, J., Subburaj, K., & Chandramohanadas, R. (2021). A deep learning approach to the screening of malaria infection: Automated and rapid cell counting, object detection and instance segmentation using Mask R-CNN. *Computerized Medical Imaging and Graphics, 88*, 101845.
67. Long, H., Chung, Y., Liu, Z., & Bu, S. (2019). Object detection in aerial images using feature fusion deep networks. *IEEE Access, 7*, 30980-30990.
68. Lu, Q., Liu, C., Jiang, Z., Men, A., & Yang, B. (2017). G-CNN: Object detection via grid convolutional neural network. *IEEE Access, 5*, 24023-24031.
69. Mahmood, Z. (2020). Connected vehicles in the IoV: Concepts, technologies and architectures. In *Connected vehicles in the internet of things: concepts, technologies and frameworks for the IoV* (pp. 3-18). Cham: Springer International Publishing.
70. Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies, 54,* 187–197. https://doi.org/10.1016/j.trc.2015.03.014
71. Maxwell, S., Kilcher, M., Benasutti, A., Siebert, B., Seto, W., Shanley, O., & Pearlstein, L. (2018). *Automated Detection of Colorspace Via Convolutional Neural Network*. Paper presented at the 2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR).
72. Mueller, M., Smith, N., & Ghanem, B. (2016). *A benchmark and simulator for uav tracking*. Paper presented at the European conference on computer vision.
73. Pang, J., Li, C., Shi, J., Xu, Z., & Feng, H. (2019). $\mathcal {R}^ 2$-CNN: Fast Tiny Object Detection in Large-scale Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing, 57*(8), 5512-5524.
74. Pan, J., Shan, H., Li, R., Wu, Y., Wu, W., & Quek, T. Q. (2021). Channel estimation based on deep learning in vehicle-to-everything environments. *IEEE Communications Letters, 25*(6), 1891-1895.
75. Pi, Y., Nath, N. D., & Behzadan, A. H. (2020). Convolutional neural networks for object detection in aerial imagery for disaster response and recovery. *Advanced Engineering Informatics, 43*, 101009.
76. Rashid, M., Khan, M. A., Sharif, M., Raza, M., Sarfraz, M. M., & Afza, F. (2019). Object detection and classification: a joint selection and fusion strategy of deep convolutional neural network and SIFT point features. *Multimedia Tools and Applications, 78*(12), 15751-15777.
77. Saetchnikov, I. V., Tcherniavskaia, E. A., & Skakun, V. V. (2020). Object detection for unmanned aerial vehicle camera via convolutional neural networks. *IEEE Journal on Miniaturization for Air and Space Systems, 2*(2), 98-103.
78. Schilling, H., Bulatov, D., Niessner, R., Middelmann, W., & Soergel, U. (2018). Detection of vehicles in multisensor data via multibranch convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11*(11), 4299-4316.
79. Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. *Procedia computer science, 132,* 377-384.
80. Shen, J., Liu, N., Sun, H., & Zhou, H. (2019). Vehicle detection in aerial images based on lightweight deep convolutional network and generative adversarial network. *IEEE Access, 7*, 148119-148130.
81. Shih, K.-H., Chiu, C.-T., Lin, J.-A., & Bu, Y.-Y. (2019). Real-time object detection with reduced region proposal network via multi-feature concatenation. *IEEE Transactions on Neural Networks and Learning Systems, 31*(6), 2164-2173.
82. Singh, B., Najibi, M., & Davis, L. S. (2018). Sniper: Efficient multi-scale training. *arXiv preprint arXiv:1805.09300*.
83. Song, Y., He, B., & Liu, P. (2019). Real-time object detection for AUVs using self-cascaded convolutional neural networks. *IEEE Journal of Oceanic Engineering*.
84. Serradilla, O., Zugasti, E., Ramirez de Okariz, J., Rodriguez, J., & Zurutuza, U. (2021). Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data. *Applied Sciences*, *11*(16), 7376.
85. Sun, Y., Yin, L., Ma, Y., & Wang, C. (2022). IoV-SDCM: An IoV Secure Data Communication Model Based on Network Encoding and Relay Collaboration. *Security and Communication Networks*, *2022*.

86. Sun, H., Sun, X., Wang, H., Li, Y., & Li, X. (2011). Automatic target detection in high-resolution remote sensing images using spatial sparse coding bag-of-words model. *IEEE Geoscience and Remote Sensing Letters, 9*(1), 109-113.
87. Tahir, M. W., Zaidi, N. A., Rao, A. A., Blank, R., Vellekoop, M. J., & Lang, W. (2018). A fungus spores dataset and a convolutional neural network based approach for fungus detection. *IEEE transactions on nanobioscience, 17*(3), 281-290.
88. Tao, X., Zhang, D., Wang, Z., Liu, X., Zhang, H., & Xu, D. (2018). Detection of power line insulator defects using aerial images analyzed with convolutional neural networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50*(4), 1486-1498.
89. Tayara, H., & Chong, K. T. (2018). Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network. *Sensors, 18*(10), 3341.
90. Tian, J., & Hu, J. (2019). *Image Target Detection Based on Deep Convolutional Neural Network.* Paper presented at the 2019 International Conference on Communications, Information System and Computer Engineering (CISCE).
91. Wang, D., He, Y., Liu, Y., Li, D., Wu, S., Qin, Y., & Xu, Z. (2019). 3D Object Detection Algorithm for Panoramic Images With Multi-Scale Convolutional Neural Network. *IEEE Access, 7*, 171461-171470.
92. Wang, J., Zheng, Y., Wang, M., Shen, Q., & Huang, J. (2020). Object-scale adaptive Convolutional Neural Networks for high-spatial resolution remote sensing image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14*, 283-299.
93. Wang, R., You, Y., Zhang, Y., Zhou, W., & Liu, J. (2018). *Ship detection in foggy remote sensing image via scene classification R-CNN.* Paper presented at the 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC).
94. Wang, X., Ma, H., Chen, X., & You, S. (2017). Edge preserving and multi-scale contextual neural network for salient object detection. *IEEE Transactions on Image Processing, 27*(1), 121-134.
95. Wang, Y., Hou, J., Hou, X., & Chau, L.-P. (2021). A self-training approach for point-supervised object detection and counting in crowds. *IEEE Transactions on Image Processing, 30*, 2876-2887.
96. Womg, A., Shafiee, M. J., Li, F., & Chwyl, B. (2018). *Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection.* Paper presented at the 2018 15th Conference on Computer and Robot Vision (CRV).
97. Wu, M., Yue, H., Wang, J., Huang, Y., Liu, M., Jiang, Y., . . . Zeng, C. (2020). Object detection based on RGC mask R-CNN. *IET Image Processing, 14*(8), 1502-1508.
98. Xiao, J., Zhang, S., Dai, Y., Jiang, Z., Yi, B., & Xu, C. (2020). Multiclass Object Detection in UAV Images Based on Rotation Region Network. *IEEE Journal on Miniaturization for Air and Space Systems, 1*(3), 188-196.
99. Xu, S., Fang, T., Li, D., & Wang, S. (2009). Object classification of aerial images with bag-of-visual words. *IEEE Geoscience and Remote Sensing Letters, 7*(2), 366-370.
100. Yang, W., Li, Z., Wang, C., & Li, J. (2020). A multi-task Faster R-CNN method for 3D vehicle detection based on a single image. *Applied Soft Computing, 95*, 106533.
101. Yang, X., Wu, T., Zhang, L., Yang, D., Wang, N., Song, B., & Gao, X. (2019). CNN with spatio-temporal information for fast suspicious object detection and recognition in THz security images. *Signal Processing, 160*, 202-214.
102. Yao, Q., Hu, X., & Lei, H. (2020). Multiscale convolutional neural networks for geospatial object detection in VHR satellite images. *IEEE Geoscience and Remote Sensing Letters, 18*(1), 23-27.
103. You, Y., Cao, J., Zhang, Y., Liu, F., & Zhou, W. (2019). Nearshore ship detection on high-resolution remote sensing image via scene-mask R-CNN. *Ieee Access, 7*, 128431-128444.
104. Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation, 31*(7), 1235-1270.
105. Zhang, D., Han, J., Cheng, G., Liu, Z., Bu, S., & Guo, L. (2014). Weakly supervised learning for target detection in remote sensing images. *IEEE Geoscience and Remote Sensing Letters, 12*(4), 701-705.
106. Zhang, Y., Yuan, Y., Feng, Y., & Lu, X. (2019). Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. *IEEE Transactions on Geoscience and Remote Sensing, 57*(8), 5535-5548.
107. Zhao, W., Ma, W., Jiao, L., Chen, P., Yang, S., & Hou, B. (2019). Multi-scale image block-level F-CNN for remote sensing images object detection. *IEEE Access, 7*, 43607-43621.
108. Zheng, S., Guo, J., Cui, X., Veldhuis, R. N., Oudkerk, M., & Van Ooijen, P. M. (2019). Automatic pulmonary nodule detection in CT scans using convolutional neural networks based on maximum intensity projection. *IEEE transactions on medical imaging, 39*(3), 797-805.