Article

An Efficient Real-Time Vehicle Classification from a Complex Image Dataset Using eXtreme Gradient Boosting and the Multi-Objective Genetic Algorithm

Pemila Mani 1, Pongiannan Rakkiya Goundar Komarasamy 2,*, Narayanamoorthi Rajamanickam 1, Roobaea Alroobaea 3, Majed Alsafyani 3 and Abdulkareem Afandi 4,*

1 Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai 603203, India; pm5291@srmist.edu.in (P.M.); narayanr@srmist.edu.in (N.R.)
2 Department of Computing Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai 603203, India
3 Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; r.robai@tu.edu.sa (R.A.); alsufyani@tu.edu.sa (M.A.)
4 General Subject Department, University of Business and Technology, Jeddah 23435, Saudi Arabia
* Correspondence: pongiank@srmist.edu.in (P.R.G.K.); a.afandi@ubt.edu.sa (A.A.)

Abstract: Recent advancements in image processing and machine-learning technologies have significantly improved vehicle monitoring and identification in road transportation systems. Vehicle classification (VC) is essential for effective monitoring and identification within large datasets. Detecting and classifying vehicles from surveillance videos into various categories is a complex challenge in current information acquisition and self-processing technology. In this paper, we implement a dual-phase procedure for vehicle selection by merging eXtreme Gradient Boosting (XGBoost) and the Multi-Objective Optimization Genetic Algorithm (Mob-GA) for VC in vehicle image datasets. In the initial phase, vehicle images are aligned using XGBoost to effectively eliminate insignificant images. In the final phase, the hybrid form of XGBoost and Mob-GA provides optimal vehicle classification with a pioneering attribute-selection technique applied by a prominent classifier on 10 publicly accessible vehicle datasets. Extensive experiments on publicly available large vehicle datasets have been conducted to demonstrate and compare the proposed approach. The experimental analysis was carried out using a myRIO FPGA board and HUSKY Lens for real-time measurements, achieving a faster execution time of 0.16 ns. The investigation results show that this hybrid algorithm offers improved evaluation measures compared to using XGBoost and Mob-GA individually for vehicle classification.

Keywords: vehicle classification; machine learning; eXtreme gradient boost algorithm; multi-objective genetic algorithm

1. Introduction

In modern times, swift progression is occurring with remarkable consequences in the vehicle manufacturing industry. Currently, automobiles in these industries are produced with different intensities, designs, and external factors, which have a huge impact on VC [1,2]. Especially during congested traffic conditions, vehicle mobility monitoring is a serious task in cases of traffic violations, toll plaza monitoring, and missing vehicle tracking. Variations in illumination, occlusion, imperfect detection, camera position, and its properties have powerful consequences on effective classification [3,4]. Machine learning (ML) algorithms resolve these issues, helping us to recognize meaningful features in image classification. These algorithms are robust in real-world situations, since they are used in decision support systems. Differentiating subordinate images through visual categories is very difficult, and the issue is addressed by a subset of ML algorithms using Convolutional...
Neural Networks (CNNs) [5, 6]. Similarly, the imbalanced data are classified using high-level global descriptors. Along with the CNN, the transfer learning process is included for VC, which provides more training efficiency and a prediction accuracy of 98% [7, 8]. Vehicle image segmentation, identification, location, and classification are performed based on ML algorithms. These algorithms learn from the features, which help in image classification.

The automatic localization of vehicles provides calculation by avoiding error-prone and manual methods, which are the conventional approaches [9, 10]. Nonlinearity and variable interdependency are also considered by ML algorithms. Unsupervised learning uses unlabeled data to make predictions, while supervised learning builds a model using labeled data (target) to predict the output value for a new set of data. The most commonly used supervised algorithms are Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Discriminant Analysis (DA), Naive Bayes (NB), Random Forests (RFs), decision trees (DTs), and K-Nearest Neighbors (KNNs) [11–13]. The selection of a suitable ML algorithm is of prime importance in solving the difficult task of finding a solution to a given classification problem [14, 15]. The selection of network/model parameters, such as the input, target, size of the training dataset, and regularization parameters, determines the performance of the ML algorithm. In intelligent transportation systems, the use of ML algorithms has greatly contributed to enhancing the accuracy of the detection and classification of vehicles. Figure 1 illustrates the general block diagram of VC.

Figure 1. General block diagram for vehicle classification using machine learning.

2. Related Work and Problem Description

The major steps of the supervised learning technique are feature extraction, representation, and classification for automatic vehicle detection processes [16, 17]. The classifier algorithms allocate labels to the data to be classified and transform them into determined data. These classifiers are utilized for VC under varying factors like shape, color, and environment. The heights and the number of axles for different classes of vehicles were classified using various classifiers like RFs, SVMs, XGBoost, and CatBoost [18, 19]. Using roadside sensors, vehicles were classified using a linear regression (LR) classifier, with an average accuracy rate of 93.4% [20, 21]. The stumbling blocks in the LR classifier include a low bias and high variance. Similarly, by employing the velocity of the automobiles, vehicles were classified with an accuracy of 75% using the Naive Bayes classifier [22, 23]. In the case of the KNN classifier, only cars were identified and localized, with an accuracy of 84% [24, 25]. The major drawback of this classifier is that its accuracy relies on the quality of the data. In the case of the SVM classifier, Gabor extraction was utilized for feature extraction and the SVM was utilized for VC [26, 27].

The improper background complexities of VC were easily handled by this classifier, with an accuracy of 87.67% [28, 29]. With the help of the acoustic signal of the vehicle, the vehicle was classified using GA optimization along with the SVM classifier, achieving a classification accuracy of over 70% [30, 31]. The pitfall of this classifier is its inability to handle a huge dataset. However, a huge dataset can be handled by an RF classifier for classifying vehicles. The Random Forest classifier can evaluate missing data and provide
accurate values [32,33]. Two stages of the Random Forest classifier were incorporated for autonomous parking [34,35]. Similarly, decision trees are also prone to noise and were used for both the regression and classification of vehicle images [36,37]. Another classifier, the ANN, validated the response of the prediction over the categories of vehicles, ranging from medium to high [38,39]. Among the different kinds of vehicles, it is tough to identify a particular one from massive traffic. To solve this issue, the most prevalent method practiced in the area of ML is attribute selection. It nominates a subset of attributes with a strong correlation to minimize the dimensionality of the dataset from the original one. The term attribute is otherwise known as a feature.

Generally, a dataset contains high-dimensional problems. To overcome this problem and also to boost the classification performance, the attribute-selection method is essential. In recent years, the most frequent methods carried out for attribute selection were the ensemble, filter, wrapper, and embedded techniques. These methods are used as preprocessing techniques to remove or eliminate unwanted attributes, which are free from repeated learning rates. The classifier is trained with the examined properties of the data. The correlation between the classes and attributes, along with analytical details of the training periods, are utilized to estimate the performance of the attribute subset. In the modern era, filter methods like t-tests, f-tests, and regression measures are used. The induction algorithm is used to investigate the attribute subset with the help of the wrapper technique. The wrapper method is considered to provide better classification performance. During classifier training, the objective function is optimized to select attributes using the embedded technique. Embedded techniques are independent of the loop partitioning of trained datasets into subcategories. Since there is no need for an iterative training processes, the selection of optimal attributes is performed faster. Even dual stages of attribute selection are experimented with. At first, the hybrid method is utilized, which combines all of the top attributes, followed by the genetic algorithm for the hyperparameters.

As mentioned, the filter and wrapper methods are widely used for attribute-selection techniques. There are various hybrid attribute-selection methods depending on the filter and wrapper methods utilized for improving the accuracy of the model. Moreover, the investigation based on embedded design is deficient. Since XGBoost is an enhancement of gradient decision trees, it has tremendous advantages in attribute selection. XGBoost focuses on calibrating model training, which can also be applied to the attribute selection. This algorithm is also useful for the evaluation of clustering and resampling data [40,41]. Similarly, XGBoost could handle tedious conditions and various pharmaceutical data to meet the basic requirements of a fast time diagnosis prediction, with greater accuracy in the classification [42,43]. Heart disease was predicted using an optimized XGBoost algorithm with an accuracy of 94.7% [44,45]. A hybrid method, principal component analysis (PCA) with XGBoost, was used for the classification of coal gangue with a 98.33% accuracy rate [46,47]. Vehicles were classified using XGBoost and CatBoost [48,49] with statistical features. Few tasks have been performed on attribute selection and classification in vehicle datasets, and those that have mainly concerned the XGBoost and genetic algorithms.

To be more accurate in risk prediction, several selection methods are used to enhance the interpretability, facilitate easy modeling, reduce the learning time, and improve the generalization. PCA and GA were implemented to train vehicle door failure classification with an accuracy of 99.8% [50,51]. Deep neural network evolution was used as a model for VC with a 78% accuracy. However, it had limitations, such as inflexible linkage levels, resulting in only minimal accuracy for the prediction assessment [52,53]. The paper by [54] proposed a hybrid attribute-selection algorithm combining the deep CNN and GA to reduce unwanted attributes across eight different classes. The major drawback of this paper was the limited dataset, considering only 8000 images. From the literature survey, it is observed that maximum label considerations, slow training processes, and imprecise vehicle classifications are common issues. Hence, in this paper, a hybrid attribute-selection method is performed using XGBoost and a Multi-Objective Genetic Algorithm, named XGBoost-MOB-GA, to address the stated problems. This procedure initially ranks the
primary attributes based on XGBoost, eliminating low-range parameters and acquiring the initial attributes. Consequently, the attained attribute subset is processed using the MOB-GA to acquire the ideal attribute subset. Depending on both attribute-selection procedures, the ideal attribute subset can be nominated from the real dataset. An equivalence and evaluation of the investigation results substantiate the merits of XGBoost-MOB-GA in terms of numerous parameters and computation times, improving the classification performance. In all of the above-stated methods, identifying the optimal number of selected attributes as a framework is necessary. Each is energized by the potential of XGBoost for attribute selection and the universal exploration potential of the genetic algorithm. From the outcome of this analysis, hybrid attribute selection is proposed to select the attributes of vehicle divisions from large surveillance vehicle datasets. The major contribution of this research work is the consideration of minimal labels from a huge dataset to classify vehicles more precisely.

The structure of this paper is organized as follows: Section 3 summarizes the postulates of XGBoost and MOB-GA in detail. Section 4 presents the implementation of a hybrid algorithm on 10 publicly accessible vehicle datasets. Section 5 presents the experiment and result analysis of the large datasets. Section 6 includes the conclusion of the work.

3. Materials and Methodology

In this section, the hybrid vehicle attribute-selection technique named XGBoost-MOB-GA is initially put forward. Sections 3.1 and 3.2 offer the postulation of XGBoost-MOB-GA consecutively.

3.1. eXtreme Gradient Boosting Algorithm

XGBoost (eXtreme Gradient Boosting) is a gradient boosting algorithm widely used in ML for huge datasets to achieve the performance of classification and regression [20–22]. Its library is written in the C++ language. Decision trees are generated in the subsequent structures and weights are added to autonomous variables, which are fed as inputs to the decision trees to provide the prediction results. Wrongly predicted weighted variables have increments in the tree and are given to the second decision tree. This unique classifier is structured as an ensemble to produce an efficient model. The regularization of the objective function depends on real factors that will limit the overfitting and regulate the convergence. XGBoost is highly advantageous over other algorithms in attribute selection. These algorithms can easily select the most relevant attribute at the beginning of the selection process. The unknown attributes are categorized based on the predicted and determined parameters. The given dataset is considered in the form of dimensions. These CART (Classification and Regression Tree) sets are produced in the form of decision trees. The sample map is given to the leaf node of the organized tree. The node count and score are then noted, which need to be arranged into an optimal model so to detect its own factors. These models are then used for the XGBoost modeling. The objective function contains the errors and complexities of the model, which are expressed as follows:

\[ \text{Obj}(t) = \sum_{i=1}^{n} B(x_i, x'_i) + \sum_{i=1}^{j} \Omega(y_i) \]

\[ \Omega(y_i) = \alpha T + \frac{1}{\beta} \sum_{i=1}^{W_j} W_j^2 \]

In Equation (1), the term \((x_i, x'_i)\) is the deviation of the square loss function with the real and predicted values. Similarly, \(\Omega(y_i)\) indicates the regularization term. \(\alpha\) and \(\beta\) express the difficulties in the co-efficient of the splitting tree for the tree formation and to reduce the overfitting issues. Once the iteration is completed, the predicted value can be given by the following:

\[ C'_j(t) = C'_j(t - 1) + d_i(b_i) \]

The objective function is expressed as follows:

\[ \text{Obj}(t) = \sum_{i=1}^{n} B(x_i, x'_i) + f_i(b_i) + \Omega(y_i) \]
The loss function relies on Taylor’s formula, with a higher convergence rate and a higher accuracy than real decision trees. The final objective function is manifested as follows:

$$\text{Obj}(t) = \sum_{i=1}^{n} \left[ E_i - \left[ E_i'(t-1) + f(t(x_i)) \right] \right]^2 + h$$

Equation (5) evaluates the nodes and minimizes the loss function.

### 3.2. Multi-Objective Genetic Algorithm

A genetic algorithm (GA) is a chronic adaptive technique that provokes an issue so to clear up the genetic development method. Afterward, it creates another generation solution via duplication, crossover, and mutation, which minimizes the fitness value. Generally, the GA is used for attribute extraction [55,56]. The GA is also used to concurrently find various spot data. The purpose of this investigation is to avoid the iteration of spot dropping into a neighborhood ideal solution. Mob-GA is relatively more able to solve optimization issues than other algorithms. Under its unique code, the input data are encoded in the form of discrete and floating-point data formats.

Lastly, the vehicle dataset values are in discrete and binary forms, where the parallel encrypting successions are obtained. But, in this paper, several attributes were organized for various attributes acquired in the first stage of the attribute selection through the XGBoost algorithm. Fitness was individually calculated with huge value probabilities for the forthcoming generation existence. Using the Mob-GA technique, the fitness value was manipulated. The distance, average, and classification methods can be designated with the Mob-GA processes for attribute selection. The purpose of this investigation is to avoid the iteration of spot dropping into a neighborhood ideal solution. Mob-GA is relatively more able to solve optimization issues than other algorithms. Under its unique code, the input data are encoded in the form of discrete and floating-point data formats.

Fitness = \(KNN_{acc}(P_t)\)  \(\text{(6)}\)

where \(P\) represents the patterns, \(t\) is the pattern label, and \(acc\) shows the accuracy. Mob-GA utilizes attribute selection with probability for the next existence and Darwin’s theory of survival of the fittest. A fitness-proportionate selection method was performed to select individuals randomly. The fitness function was allotted to the favorable solutions, in which each scheme N of individuals was selected out of the offspring population. The process was repeated until the original population size was recovered. Following the selection operation, the crossover was carried out. In this paper, Davi’s order (OX1) crossover method was implemented to form two new individuals based on the crossover probability.

3.3. eXtreme Gradient Boosting Algorithm and Multi-Objective Genetic Algorithm-Based Hybrid Algorithm

In the architecture of the ML prototype for building the model selection of specific factors, the neural network and kernel parameters instantaneously have influence in the accuracy and generalization of the model. Attribute selection is the predominant method used in the establishment of the model, which is unable to store data and represent the accurate real data. Ensemble learning is the most important task for the classification, since it needs to combine the models most productively. The factors which affect the attribute selection for the image classification are the segmentation of attributes with a low speed, as
more attributes need to be considered, the greater execution time, difficulty in handling the objectives, constraints, encoding, scaling, the unreliability of estimation, metric non-availability, handling large datasets, missing values, low performance, and the accuracy of the classification. These listed factors are overcome by the XGBoost and Mob-GA-based hybrid algorithm. A detailed explanation of this hybrid algorithm is presented in the forthcoming section.

4. Implementation of the Hybrid Algorithm Using eXtreme Gradient Boosting Algorithm and Multi-Objective Genetic Algorithm-Based Hybrid Algorithm

In this phase, a dual-phase attribute-selection method, eXtreme Gradient Boosting merged with the Multi-Objective Genetic Algorithm, namely, XGBoost-Mob-GA for VC in vehicle datasets, is proposed. The schematic flow diagram of the hybrid XGBoost and Mob-GA is presented in Figure 2. In the initial phase, the label is categorized through XGBoost, and a score greater than zero retains the labels. This phase can detach unwanted labels and combine them with those most pertinent to the class. In the second phase, XGBoost-Mob-GA finds the most pertinent group for the optimal label subset by using Mob-GA.

![Flowchart for vehicle classification using XGBoost and Mob-GA.](image)

This further brings out the superfluous label while still extending the VC. In Algorithm 1, the XGBst() function computes the score of each label, similarly to the pn() function, thus building the embryonic population. In Table 1, we provide the parameter settings of XGBoost and Mob-GA, respectively. Prior to the training process, a few hyperparameters were identified with definite values, and are utilized for the entire research work, as these hyperparameters play an important role in the vehicle selection outcome of XGBoost, as demonstrated in Table 1. Preceded by this, the XGBoost-Mob-GA computational complexity is explained.

<table>
<thead>
<tr>
<th>Table 1. Parameters of XGBoost and MOB-GA.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XGBoost</strong></td>
</tr>
<tr>
<td>Term</td>
</tr>
<tr>
<td>Learning rate</td>
</tr>
<tr>
<td>Maximum depth of trees</td>
</tr>
<tr>
<td>Subsampling rate</td>
</tr>
</tbody>
</table>
The time of the computational problem is based on the initial label selection and the identification of the optimum subset of the label. In the case of the label selection, the logarithmic of the label is taken with respect to the samples and similarly, while the second computational problem is calculated by an embedded method based on iteration and the population size from the label subset. For the vehicle dataset, the sample size will be smaller when compared to the label and its population size. XGBoost, Python, the sci-kit-feature, and learn are the publicly available software products and languages utilized for the implementation of the hybrid Mob-GA classification algorithm.

Algorithm 1: eXtreme Gradient Boosting and Multi-Objective Genetic Algorithm

```
input: A(B₁, B₂, B₃, B₄, ..., Bₙ, C)
output: Afinest
initialize a variable with an arrays
chromosome is zero
B-significance is equivalent to XGBst(A)
for(int i = 0; i < n; i++)
  if B-significance[i] > 0
    increase feature i into initialize data
chromosome increased by the variable 1
finest population with the array of variable
finest-score are same as variable of array
for(int j = 0; j < recurrence; j++)
  Initialize the population which is equal to pn(Initialize data, c)
  Sum initialize of population into finest population
  Initialize variable-score = ANN + KNN–fitness (initialize population)
  Add initialize variable-score into finest-score
  If (initialize variable–score ≥ allotted variable-score)
    Or
    Select = fitness candidate
    crossover ()
    mutation ()
    initialise population of crossover (), mutation (), chromosome
    Outcome = maximum (finest-score)
    Afinest = finest population of outcome
return Afinest
```

5. Experiments

The experiments were performed on the suggested procedure to authenticate the performance. The dataset is essential for the experimental investigation, and the VC procedure, the similarities in the attribute-selection methods, the outcome, and the analysis are furnished in this section. The developed hybrid Mob-GA algorithm was run on a personal computer outfit with a corei7-12600K and 3.7 G of memory.

5.1. Datasets

The experiment was conducted on 10 publicly accessible vehicle datasets, along with the real data shown in Figure 3. The properties of the 10 vehicle datasets include the number of attributes, samples and multi-classes. The number of attributes ranged from 1000 to 50,456, and the number of samples was 4250. The dataset of the comp cars contained a tri-class dataset, while the rest of the datasets were binary. In some of the vehicle datasets containing multi-classes and particular cases, the attributes were binary with misplaced data. The vehicle dataset contained lost data, and some of the interposed data were incorporated to sort out the issue of misplaced data. In this paper, attribute scaling was carried out to restore the lost data. Each sample of lost data was assigned by attribute scaling in the training phase. Once the lost value was refined, the advanced systematized data were captured to route the second data value (0, 1).
5.1. Datasets

The experiment was conducted on 10 publicly accessible vehicle datasets, along with the real data shown in Figure 3. The properties of the 10 vehicle datasets include the number of attributes, samples and multi-classes. The number of attributes ranged from 1000 to 50,456, and the number of samples was 4250. The dataset of the comp cars contained a tri-class dataset, while the rest of the datasets were binary. In some of the vehicle datasets containing multi-classes and particular cases, the attributes were binary with misplaced data. The vehicle dataset contained lost data, and some of the interposed data were incorporated to sort out the issue of misplaced data. In this paper, attribute scaling was carried out to restore the lost data. Each sample of lost data was assigned by attribute scaling in the training phase. Once the lost value was refined, the advanced systemized data were captured to route the second data value (0, 1).

Figure 3. Collection of different publicly accessible vehicle datasets (A–J) utilized for the training and testing of the real data (K).

5.2. Hardware/Software Experimentation

The experiments were performed with a double classifier including various archetypes, i.e., K-Nearest Neighbors (KNNs) and Artificial Neural Networks (ANNs), to estimate the selected attributes for each attribute-selection method. Subsequently, the 10-fold cross-validation for the classification output was monitored. XGBoost, Python, the ssci-kit-feature, and learn were the publicly available software products and languages utilized for the classification algorithms. In this paper, the KNN classifier was used to find the class of an image based on weighted frequency analysis; hence, it is a parametric-less classifier. The ANN classifier is a disbursed collateral processor which contains three layers. Depending on the weight and biases, the input and the output were recorded. The efficiency was calculated by the mean square error (MSE). Based on various metrics, it provides different outputs. There were various evaluation measures employed based on the accuracy (Ac), F1 score (Sc), precision (Pr), and recall (Re) [57,58]. An identical weight was added to each class in the multi-classification issue to compute the above measures.

5.2.1. Software Experimentation

The average value of the measure is the final outcome. To impose the performance of the classification algorithm, KNN and ANN classifiers were used. A 10-fold cross-validation was performed with the classifiers on different labels to produce a moderate outcome. The components of XGBoost-Mob-GA were compared to authenticate the persuasiveness of the combination. In addition to this, the different attribute-selection approaches were nominated to determine the merit of this hybrid method. The number of labels was selected by three different methods. With the reference, the number of labels was taken from the original dataset. The number of labels adopted in this work was lower when compared to the prototypical dataset [18].
5.2.2. Hardware Experimentation

XGBoost opts for fewer labels from the raw dataset. XGBoost-Mob-GA was chosen for its low number of labels when compared to the individual algorithms [40]. In Figure 4, the vehicle is classified under various conditions using the hybrid algorithm with a given experimental setup. Then, in each and every experiment, an average of the outcomes was attained for the huge dataset computation in order to perform an easy comparison. The experiment was carried out, including the FPGA implementation, using the myRIO board to sense the color and track the vehicle images in complex environments. The specifications include a Xilinx Z-7010 processor with a frequency of 667 MHz and a non-volatile memory of 256 MB, and a double data rate third generation (DDR3) of 512 MB with a frequency 533 MHz and containing 16 bits. Figure 5 shows the final prototype of the XGBoost and MOBGA classifier implemented on the myRIO board. In addition, the classification performance was calculated using receiver operating characteristic (ROC) analysis to validate the results, as shown in Figure 6. As depicted in Figure 7, the performance of the VC is defined.

![Figure 4](image1.jpg)

(a) (b) (c)

**Figure 4.** Classification of the vehicles using XGBoost and Mob-GA under various conditions: (a) fog; (b) cloudy; (c) dark.

![Figure 5](image2.jpg)

**Figure 5.** Embedded processor package of the NI myRIO for the classification of vehicles.
Processes 2024, 12, x FOR PEER REVIEW 13 of 22

Re 91.06 61.40 91.01 61.40 95.85 92.06 97.00 93.06
MVVTR
Ac 70.00 47.56 68.78 48.56 82.56 54.56 81.22 55.56
Sc 69.61 40.49 68.32 40.02 82.40 41.03 80.71 41.33
Pr 70.81 43.36 70.20 44.08 83.46 41.02 82.45 41.21
Re 70.07 46.48 68.72 46.32 82.64 51.14 81.00 53.32

Vehicle-Rear
Ac 94.78 80.60 94.78 82.10 100.00 95.08 100.00 96.21
Sc 93.20 70.59 93.20 72.07 100.00 93.86 100.00 95.31
Pr 95.56 72.05 95.56 74.62 100.00 94.72 100.00 96.40
Re 93.16 71.40 93.06 73.16 100.00 94.72 100.00 95.56

Figure 5. Embedded processor package of the NI myRIO for the classification of vehicles.

Figure 6. Simulation output using Xilinx Z-7010 with the frequency of 667 MHz: (a) labels; (b) classes of the vehicles under a traffic environment.

Figure 7. ROC curves based on the calculated preliminary array values using CDF.

Table 2 shows the number of labels nominated by Mob-GA, XGBoost, and XGBoost-Mob-GA. The major purpose was to furnish the table with datasets in a way which was not sporadic, as these datasets were not highly focused in other studies. The Vehicle Re-identification (VeRi) dataset [59,60] was built for vehicle re-identification from an original surveillance scene, which was labeled with various attributes and which contains 50 k images captured from different camcorders covering a distance of 1 km. The sample of labeled parameters with spatiotemporal factors includes boxes of plates, strings of plates, and timestamps of different vehicles. In the case of the comprehensive cars (CompCars) dataset [61,62], it contains 1,367,26 images of cars captured in a web and surveillance context.

The labeled part includes boxes and viewpoints enclosing the attributes of the maximum speed, displacement, door and seat counts, and the types of cars. The VehicleX dataset [63,64] is a synthetic dataset containing 3D models with complete modifiable attributes of 1362 vehicles. This dataset includes all-inclusive real-world and re-identification...
data to reduce the problem complexity. The UFPR–ALPR [65,66] dataset includes 45,000 images that are completely annotated in real-world scenarios, with pixel sizes of 1920 × 1080, taken from go pro hero 4 silver, Huawei P9 Lite, and iPhone 7 plus equipment.

Table 2. Number of selected labels by the real data, XGBoost, Mob-GA, and XGBoost-MO-GA.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Real</th>
<th>Mob-GA</th>
<th>XGBoost</th>
<th>XGBoost-MO-GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>VeRi</td>
<td>50,000</td>
<td>10,000</td>
<td>36</td>
<td>10</td>
</tr>
<tr>
<td>CompCars</td>
<td>27,618</td>
<td>7156</td>
<td>132</td>
<td>68</td>
</tr>
<tr>
<td>VehicleX</td>
<td>1362</td>
<td>324</td>
<td>58</td>
<td>25</td>
</tr>
<tr>
<td>UFPR-ALPR</td>
<td>4500</td>
<td>536</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>Tsinghua-Tencent</td>
<td>10,000</td>
<td>5632</td>
<td>97</td>
<td>31</td>
</tr>
<tr>
<td>Stanford car</td>
<td>16,185</td>
<td>6543</td>
<td>61</td>
<td>28</td>
</tr>
<tr>
<td>TRANCOS</td>
<td>1244</td>
<td>207</td>
<td>33</td>
<td>15</td>
</tr>
<tr>
<td>Indian Vehicle</td>
<td>35,000</td>
<td>8056</td>
<td>80</td>
<td>54</td>
</tr>
<tr>
<td>MVVTR</td>
<td>4793</td>
<td>436</td>
<td>22</td>
<td>7</td>
</tr>
<tr>
<td>Vehicle-Rear</td>
<td>2500</td>
<td>501</td>
<td>06</td>
<td>5</td>
</tr>
</tbody>
</table>

The labels of the portrayed attributes include the manufacturer, models and years of the cars and motorcycles, as well as the position, identification, and characters of the license plates (LP) of the vehicles. The Tsinghua–Tencent [67,68] dataset contains 100 K images with 30 K traffic signs. From these, 10 K images of vehicles are considered for our VC research work. Bounding boxes and pixel masks are considered for the labels of the attributes of the vehicles. In the Stanford car dataset [33], 16,185 images with 196 different classes of cars are accommodated. The makes, models, and years of the cars are considered as the labels, and the classes are equally split for the training and testing. For the testing, either the real data or the raw data were fed into the classifier.

In the TRANCOS dataset [69,70], 1244 images with 46,796 vehicles are annotated and captured from a CCTV system in Spain. The region of interest (ROI) of the road region identification, the location, and the count of the vehicles are illustrated. The Indian Vehicle dataset includes around 50 K images of Indian vehicles captured from both urban and rural areas, with pixel sizes of 1920 × 1080 and above. Only seven classes of vehicles are noted in this dataset. In our research work, out of the 50 K images, only 35 K images are considered for the VC. In the multi-view vehicle-type recognition (MVVTR) [71,72] dataset, seven major vehicle types and 4793 images are included, while 1000 images were chosen for the VC in our work. Up–left, up–right, down–left, down–right, center, and mirrored vehicles comprise the classification of vehicles as labeled parts in this dataset. The Vehicle–Rear [73,74] dataset contains 3 h of high-resolution surveillance video. Labels include the makes, models, colors, and years of the vehicles, and the accurate location and detection of LP information is also provided. Furthermore, the numbers of labels in the real dataset are given for reference. It was inferred that the number of labels opted by Mob-GA was less than half of the real dataset, while XGBoost [75,76] chose even fewer labels from the real dataset.

Table 3 postulates the average performances of the 10 publicly accessible datasets for the two classifiers based on the evaluation metrics. Furthermore, it displays a list of the proposed approaches and benchmark methods with detailed descriptions. Additionally, the performance of the KNN and ANN methods is also used as a comparison with our proposed method. In the proposed method, extensive experiments on publicly available large vehicle datasets—VeRi and VehicleX—have determined the superiority over the MVVTR dataset, with gain percentages of 44.44%, 58.67%, 58.79%, and 46.68% in the accuracy, F1 score, precision, and recall.
Table 3. Number of selected labels by the real data, XGBoost, Mob-GA, and XGBoost-Mob-GA.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metrics</th>
<th>Real KNN</th>
<th>ANN</th>
<th>Mob-GA KNN</th>
<th>ANN</th>
<th>XGBoost KNN</th>
<th>ANN</th>
<th>XGBoost-Mob-GA KNN</th>
<th>ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td>VeRi</td>
<td>Ac</td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>Comp Cars</td>
<td>Ac</td>
<td>62.71</td>
<td></td>
<td>69.00</td>
<td></td>
<td>61.52</td>
<td></td>
<td>65.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>57.64</td>
<td></td>
<td>53.70</td>
<td></td>
<td>56.34</td>
<td></td>
<td>50.43</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>59.25</td>
<td></td>
<td>51.56</td>
<td></td>
<td>57.3</td>
<td></td>
<td>48.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>71.00</td>
<td></td>
<td>59.31</td>
<td></td>
<td>71.00</td>
<td></td>
<td>56.40</td>
<td></td>
</tr>
<tr>
<td>VehicleX</td>
<td>Ac</td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>UFPR-ALPR</td>
<td>Ac</td>
<td>67.12</td>
<td></td>
<td>69.00</td>
<td></td>
<td>69.00</td>
<td></td>
<td>73.22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>58.21</td>
<td></td>
<td>64.58</td>
<td></td>
<td>57.80</td>
<td></td>
<td>64.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>58.12</td>
<td></td>
<td>68.40</td>
<td></td>
<td>61.66</td>
<td></td>
<td>66.47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>61.01</td>
<td></td>
<td>64.00</td>
<td></td>
<td>62.64</td>
<td></td>
<td>65.14</td>
<td></td>
</tr>
<tr>
<td>Tsinghua-Tencent</td>
<td>Ac</td>
<td>84.32</td>
<td></td>
<td>79.60</td>
<td></td>
<td>97.42</td>
<td></td>
<td>82.10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>82.81</td>
<td></td>
<td>69.48</td>
<td></td>
<td>97.33</td>
<td></td>
<td>72.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>86.47</td>
<td></td>
<td>72.05</td>
<td></td>
<td>98.00</td>
<td></td>
<td>74.62</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>84.00</td>
<td></td>
<td>71.40</td>
<td></td>
<td>97.11</td>
<td></td>
<td>73.06</td>
<td></td>
</tr>
<tr>
<td>Stanford car</td>
<td>Ac</td>
<td>84.62</td>
<td></td>
<td>57.02</td>
<td></td>
<td>82.70</td>
<td></td>
<td>58.41</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>82.82</td>
<td></td>
<td>55.18</td>
<td></td>
<td>68.66</td>
<td></td>
<td>56.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>86.47</td>
<td></td>
<td>58.31</td>
<td></td>
<td>70.43</td>
<td></td>
<td>59.64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>84.00</td>
<td></td>
<td>61.81</td>
<td></td>
<td>80.56</td>
<td></td>
<td>63.47</td>
<td></td>
</tr>
<tr>
<td>TRANCOS</td>
<td>Ac</td>
<td>97.46</td>
<td></td>
<td>97.22</td>
<td></td>
<td>98.33</td>
<td></td>
<td>96.11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>97.33</td>
<td></td>
<td>95.20</td>
<td></td>
<td>97.73</td>
<td></td>
<td>93.14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>98.00</td>
<td></td>
<td>97.05</td>
<td></td>
<td>98.58</td>
<td></td>
<td>95.50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>97.22</td>
<td></td>
<td>95.22</td>
<td></td>
<td>97.12</td>
<td></td>
<td>93.22</td>
<td></td>
</tr>
<tr>
<td>Indian Vehicle</td>
<td>Ac</td>
<td>91.08</td>
<td></td>
<td>61.62</td>
<td></td>
<td>91.08</td>
<td></td>
<td>61.65</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>91.01</td>
<td></td>
<td>57.04</td>
<td></td>
<td>91.81</td>
<td></td>
<td>57.15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>91.81</td>
<td></td>
<td>64.42</td>
<td></td>
<td>91.06</td>
<td></td>
<td>63.50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>91.06</td>
<td></td>
<td>61.40</td>
<td></td>
<td>91.01</td>
<td></td>
<td>61.40</td>
<td></td>
</tr>
<tr>
<td>MVVTR</td>
<td>Ac</td>
<td>70.00</td>
<td></td>
<td>47.56</td>
<td></td>
<td>68.78</td>
<td></td>
<td>48.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>69.61</td>
<td></td>
<td>40.49</td>
<td></td>
<td>68.32</td>
<td></td>
<td>40.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>70.81</td>
<td></td>
<td>43.36</td>
<td></td>
<td>70.20</td>
<td></td>
<td>44.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>70.07</td>
<td></td>
<td>46.48</td>
<td></td>
<td>68.72</td>
<td></td>
<td>46.32</td>
<td></td>
</tr>
<tr>
<td>Vehicle-Rear</td>
<td>Ac</td>
<td>94.78</td>
<td></td>
<td>80.60</td>
<td></td>
<td>94.78</td>
<td></td>
<td>82.10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sc</td>
<td>93.20</td>
<td></td>
<td>70.59</td>
<td></td>
<td>93.20</td>
<td></td>
<td>72.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pr</td>
<td>95.56</td>
<td></td>
<td>72.05</td>
<td></td>
<td>95.56</td>
<td></td>
<td>74.62</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re</td>
<td>93.16</td>
<td></td>
<td>71.40</td>
<td></td>
<td>93.06</td>
<td></td>
<td>73.16</td>
<td></td>
</tr>
</tbody>
</table>
The MVVTR dataset has faced challenges due to the occlusion of the vehicles. This has led to minimum accuracy in vehicle classification. Compared with the real dataset, XGBoost, and Mob-GA, the proposed XGBoost-Mob-GA technique provides better results in terms of the accuracy, F1 score, precision, and recall in most of the criteria. The proportion of the hybrid algorithm has an optimal accuracy of 9/10 and 9/10 for the above-considered classifiers, respectively. Specifically, XGBoost-Mob-GA attained the prime performance on the KNN classifier. As shown in Table 3, XGBoost-Mob-GA achieved 100% on the five datasets of the KNN classifier based on four evaluation cases.

In the case of the XGBoost approach with the KNN classifier, three datasets gained a 100% classification performance. The performances of the VeRi and Vehicle X datasets reached 100% in the four methods based on the KNN classifier. XGBoost and XGBoost-Mob-GA equally achieved a 100% classification performance on the three and four datasets with the ANN classifier. The results of the four evaluation criteria demonstrate that XGBoost-Mob-GA performed the best, thereby validating the efficiency of XGBoost-Mob-GA.

In order to simulate the vehicle color extraction, a color extraction model was mounted on an NI myRIO. Figure 5 shows the hardware experimental setup with the NI myRIO package for the vehicle classification. The NI myRIO consists of a processor, the FPGA, software, and an I/O port. This system includes timing control with a high-performance I/O hardware circuit. After testing the good condition of the myRIO, the LabVIEW 2015 programming environment was used to identify the myRIO project to organize the particular algorithm using it as a virtual instrument. The color of the vehicles was the major attribute considered for the representation and tracking. The model had a computation speed of 0.16 ns. In the test environment, the vehicle images were captured under different illumination conditions such as dark, fog, etc. In Figure 6, the hardware simulation output is shown, where the attributes are displayed in the form of the nodes and types of the vehicles, which are portrayed with the evaluation metrics of each vehicle.

The results of the processes included populated arrays with different class test probabilities, denoting the true-positive rate and false-positive rate for the given vehicle dataset. Figure 7 depicts the ROC curve depending upon the calculated primary array values by means of the cumulative distribution function (CDF). For each iteration over a different value of the threshold, the ROC is formed. At the starting stage, the ROC is not generated due to the lack of the presence of minimum data points shifting from the original to the new array. Figure 7 shows the threshold value of 0.64 at the early iteration of the ROC curve, and both positive and negative arrays are less than 30.

Figure 8 portrays the accuracy rates of each proposed technique concerning the various sizes of the vehicle images. Here, the output shows that XGBoost-Mob-GA has the highest accuracy rate of 99.8%. Mob-GA and XGBoost follow with an 80–89% accuracy. The accuracy rate is the same as the Python and hardware implementations on the FPGA. The results of the software implementation are provided directly. Although finding the best vehicle classification attribute value results is a crucial procedure, precautions should be taken with the interpretation of these outcomes when designing the hardware, because the architecture of the design of the attributes requires more pixels and consumes larger areas in the FPGA. Thus, the Xilinx used in the cases with complex architectures are unreal. The XGBoost-Mob-GA exceeds the LUT by up to $32 \times 32$ pixels in the KNN and ANN.

Similarly, the confusion matrix for the classification of the 2 W, 3 W, and 4 W vehicles of the AlexNet network are portrayed in Figure 9. The results in Figure 9 show that the texture and shape are easily predicted as the strongest matches by the network, which may be due to the confined labels of the images from the dataset. A total of 90% of the dataset was used for training, while the remaining 10% was used for testing the model; these testing models provide the accuracy rate of the utilization. A total of 203 general images and 537 vehicle images were used for testing. The results of the classification can be observed in the confusion matrix in the AlexNet models. As demonstrated by the comprehensive investigation of the classification metrics, the performance of the system is perceptible, as shown in Figure 9.
Figure 7. ROC curves based on the calculated preliminary array values using CDF.

Figure 8. Classification accuracy over 10 datasets based on different attributes.

Figure 9. Confusion matrices illustrating the vehicle classification based on the XGBoost and MOB-GA classifiers.

Figure 10 demonstrates the classification algorithm running times of the vehicle images on the software and hardware. The fastest hardware design is the KNN at 42.34 ns, followed by the ANN at 52.78 ns and XGBoost-Mob at 300.62 ns. The classifier time for detection in Python is 7834 ns, while for the KNN and for the ANN and decision tree, it is 62,535 ns. The most prominent analysis is that the classification algorithms implemented on the FPGA are quicker than Python. Figure 11 indicates the LUT limit on the FPGA for the design with the 8 × 8, 16 × 16, and 32 × 32 pixel attributes. The XGBOOST-MOB-GA gained highest accuracy percentage with the minimum pixel attribute of 8 × 8, whereas XGBoost and MOB-GA gained the highest accuracies with the 16 × 16 pixel attribute. Figure 12 indicates the stimulation results of the types of vehicles and evaluation metrics. Table 3 examines the XGBOOST-MOB-GA hardware design success in the LUT in detail.

Figure 9 illustrates the confusion matrices for vehicle classification with the image processing techniques and the machine learning algorithm. The color in each square indicates the range of the cell. The brighter colors show the maximum matches. The values along the diagonal of the confusion matrix show the matches, whereas the off-diagonals indicate fewer match types for the vehicles. The following are the mathematical formulae used to calculate the different metrics in the proposed work.

- **Mean Square Error (MSE):** The MSE measures the average of the squares of the errors, which are the differences between the predicted and actual values.

  \[
  MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2
  \]  

  (7)

  where \( n \) is the number of observations, \( y_i \) is the actual value, and \( \hat{y}_i \) is the predicted value.
• **Accuracy**: Accuracy is the ratio of the number of correctly predicted instances to the total number of instances.

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]

where \(TP\) is the number of true positives, \(TN\) is the number of true negatives, \(FP\) is the number of false positives, and \(FN\) is the number of false negatives.

• **Precision**: Precision is the ratio of the number of true-positive instances to the total number of instances predicted as positive.

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

• **Recall**: Recall (also known as the sensitivity or the true-positive rate) is the ratio of the number of true-positive instances to the total number of actual positive instances.

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

---

**Figure 10.** Comparison of the detection times between the hardware and simulation.

**Figure 11.** Comparison of the detection times among different algorithms.

Figure 10 shows the comparison of different hardware simulations for the vehicle classification in different datasets. The orange color bar chart indicates the MATLAB simulation versus the classification time of the vehicle images. Similarly, the blue color indicates the FPGA simulation versus the classification time.

Figure 11 shows the resource utilization time of the proposed technique using the FPGA implementation for vehicle classification under complex environments.
Figure 12 shows the different types of vehicle classifications from the testing data. In Figure 12b, the evaluation metrics are shown for various types of vehicles under tedious conditions. Hence, the proposed hybrid approach combining eXtreme Gradient Boosting (XGBoost) and the Multi-Objective Optimization Genetic Algorithm (Mob-GA) to enhance the vehicle classification (VC) from image datasets addresses the challenges in the detection and classification of vehicles from surveillance videos; the proposed method focuses on effective attribute selection and the elimination of insignificant images. Initially, XGBoost aligns the vehicle images, followed by a hybrid phase where XGBoost and Mob-GA optimize the attribute selection for improved classification. Evaluated on 10 publicly accessible vehicle datasets, including VeRi and VehicleX, the method achieved significant improvements in the accuracy (44.44%), F1 score (58.67%), precision (58.79%), and recall (46.68%) compared to the MVVTR dataset. A real-time implementation using an FPGA board demonstrated a fast execution time of 0.16 ns, indicating low computation costs and rapid processing capabilities. This study highlights the hybrid approach’s ability to handle large datasets efficiently, providing superior performance metrics over existing methods. However, limitations include the reliance on specific hardware, the limited dataset scope, and the need for further investigation into the scalability and real-time performance under diverse conditions. Overall, this research contributes a robust method for vehicle classification, demonstrating significant advancements in the accuracy and efficiency which are suitable for intelligent transportation systems.

5.3. Summary Discussion

The main objective of this paper was to propose an XGBoost–Mob-GA approach towards realizing an efficient embedded vehicle class classification system from a complex dataset. This paper aimed to contribute to the existing challenges and bottlenecks in the latest literature. This paper was highly focused on converging the arduous constraints of embedded systems development with a higher accuracy rate. The real-time implementation in the FPGA platform made the proposed approach more suitable for VC with a shorter detection time. In this research, the proposed method has following merits when compared to conventional ML approaches:

(i) It can furnish more robust and accurate results, and it is hard to overfit when compared to traditional approaches such as SVMs, ANNs, and RFs.

(ii) When compared to single-objective optimization, our proposed method can optimize the classification accuracy and efficiency.

(iii) Our proposed method was superior to XGBoost, since the optimized parameters led to a considerable potency in the performance of the classification results.
(iv) The proposed method provides a lower computation cost when compared to complex GUIs.
(v) Minimum labels were taken from the complex dataset and used to classify vehicles under complex environments.
(vi) This hybrid method provides notable accuracy for vehicle classification.

5.4. Limitations and Applications

Vehicle classification plays a vital role in image processing. The proposed method is used to classify vehicles from complex datasets. The limitations of this paper include the following:

- This study only considers 10 publicly accessible vehicle datasets. While these datasets provide a good starting point, the performance of the proposed method on other datasets, especially those with different characteristics or larger volumes, was not evaluated. This could limit the generalizability of the results to different real-world scenarios.
- The implementation and performance evaluation were carried out using specific hardware, namely, the myRIO FPGA board and HUSKY Lens. The reliance on specialized hardware might limit the accessibility and applicability of the proposed method to environments where such hardware is unavailable or impractical to use.
- The proposed method primarily focuses on the attribute selection process and its impact on vehicle classification. While this is crucial, other aspects of the machine-learning pipeline, such as data preprocessing, model interpretability, and robustness to noise or variations in the data, are not addressed in detail, which could affect the overall performance and usability.
- Although the study claims fast execution times and improved performance, the scalability of the proposed method to handle even larger datasets or to operate in real-time under varying traffic conditions is not thoroughly explored. Future work should investigate the method’s performance under different scales and real-time constraints to ensure its practical applicability.

The following are the major applications of the proposed work:

- Traffic Monitoring and Management: Real-time vehicle classification can help in monitoring traffic flow, identifying congestion points, and optimizing traffic signal timings to improve traffic management.
- Automated Toll Collection: The system can be used to classify vehicles for automated toll collection systems, ensuring accurate toll charges based on vehicle type and size.
- Smart Parking Systems: In smart parking systems, real-time vehicle classification can assist in managing parking spaces by directing different types of vehicles to appropriate parking areas and ensuring efficient space utilization.
- Law Enforcement and Surveillance: Law enforcement agencies can use the system for real-time surveillance, identifying and tracking specific types of vehicles involved in criminal activities or traffic violations.
- Fleet Management: Companies with large fleets can use this technology to monitor and manage their vehicles in real-time, optimizing routes, scheduling maintenance, and improving overall operational efficiency.

6. Conclusions

In this study, we implemented a hybrid attribute-selection approach using XGBoost and Mob-GA for vehicle classification (VC) in vehicle datasets, as well as utilizing Python 3.11.5, the sci-kit-feature, and learn. The hardware implementation on an FPGA board demonstrated a computation speed of 0.16 ns. We considered 10 publicly accessible vehicle datasets, and by applying the hybrid attribute-selection technique, we identified the most significant labels. These labels were transformed and processed at a high speed using an embedded approach, leading to an improved VC performance. Extensive experiments on large vehicle datasets, including VeRi and VehicleX, showed that the proposed method outperformed the MVVTR dataset, achieving gains of 44.44%, 58.67%, 58.79%, and 46.68%.
in the accuracy, F1 score, precision, and recall, respectively. The FPGA-based implementation resulted in lower computation costs, faster training processes, and a more advanced model for classifying vehicles from large datasets. Our advanced evaluation metrics, such as the accuracy, F1 score, and recall, confirmed the superior performance of the hybrid XGBoost-MOB-GA approach for vehicle classification.

**Author Contributions:** Conceptualization, P.M. and P.R.G.K.; methodology, N.R.; software, P.M.; validation, N.R., P.M. and M.A.; formal analysis, A.A.; investigation, P.M.; resources, N.R.; data curation, P.R.G.K.; writing—original draft preparation, P.M.; writing—review and editing, N.R.; visualization, P.R.G.K.; supervision, P.R.G.K.; project administration, R.A.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Taif University, Taif, Saudi Arabia (TU-DSPP-2024-17).

**Data Availability Statement:** Data are contained within the article.

**Acknowledgments:** The author extend their appreciation to Taif University, Saudi Arabia, for supporting this work through project number (TU-DSPP-2024-17).

**Conflicts of Interest:** The authors declare no conflicts of interest.

**Nomenclature**

**Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC</td>
<td>Vehicle classification</td>
</tr>
<tr>
<td>XGBoost</td>
<td>eXtreme gradient boosting</td>
</tr>
<tr>
<td>XGBoost-Mob-GA</td>
<td>Multi-objective optimization genetic algorithm</td>
</tr>
<tr>
<td>ML</td>
<td>Machine learning</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolution neural network</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial neural network</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machine</td>
</tr>
<tr>
<td>DA</td>
<td>Discriminant analysis</td>
</tr>
<tr>
<td>RF</td>
<td>Random forest</td>
</tr>
<tr>
<td>NB</td>
<td>Naive Bayes</td>
</tr>
<tr>
<td>DT</td>
<td>Decision trees</td>
</tr>
<tr>
<td>KNN</td>
<td>K-nearest neighbor</td>
</tr>
<tr>
<td>LR</td>
<td>Linear regression</td>
</tr>
<tr>
<td>CB</td>
<td>CatBoost</td>
</tr>
<tr>
<td>PCA</td>
<td>Principle component analysis</td>
</tr>
<tr>
<td>CART</td>
<td>Classification and regression trees</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic algorithm</td>
</tr>
<tr>
<td>ROCs</td>
<td>Receiver operating characteristics</td>
</tr>
<tr>
<td>VeRi</td>
<td>Vehicle Re-identification</td>
</tr>
<tr>
<td>CompCars</td>
<td>Comprehensive cars</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of interest</td>
</tr>
<tr>
<td>MVVTR</td>
<td>Multi-view vehicle-type recognition</td>
</tr>
<tr>
<td>NI</td>
<td>National Instruments</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative distribution function</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-programmable gate array</td>
</tr>
</tbody>
</table>

**Variable Definition**

- $Obj(t)$: Objective function for the model
- $n$: Number of trees
- $y_i$: Functional space
- $x_i$: Real data
- $x'_i$: Predicted data
- $\Omega(y_i)$: Regularization parameter
- $\beta$: Co-efficient of splitting trees
- $P_t$: Label of the pattern
- $y$: Testing data


4. Yin, Y.; Guo, Y.; Su, Q.; Wang, Z. Task Allocation of Multiple Unmanned Aerial Vehicles Based on Deep Transfer Reinforcement Learning. Drones 2022, 6, 215. [CrossRef]


29. Bhatt, P.; Maclean, A.L. Comparison of high-resolution NAIP and unmanned aerial vehicle (UAV) imagery for natural vegetation communities classification using machine learning approaches. *GisCI. Remote Sens.* 2023, 60, 2177448. [CrossRef]


34. Tang, Y.; Wang, C.; Gao, J.; Hua, P. Intelligent Detection Method of Forgings Defects Detection Based on Improved EfficientNet and Memetic Algorithm. *IEEE Access* 2022, 10, 79553–79563. [CrossRef]


69. Feng, J.; Wang, Y.; Liu, Z. Joint impact of service efficiency and salvage value on the manufacturer’s shared vehicle-type strategies. Rairo-Oper. Res. 2024, 58, 2261–2287. [CrossRef]


