EPiC Series in Computing

Volume 102, 2024, Pages 33–43

Proceedings of Conference on Cognitive and Computational Aspects of Situation Management 2023

Comparison of different hyperparameter optimization methods on driving behavior recognition

Ruth David[1] and Dirk Söffker[2]

[1] University of Duisburg-Essen, Duisburg, Germany
ruth.david@uni-due.de
[2] University of Duisburg-Essen, Duisburg, Germany
soeffker@uni-due.de

Abstract

The prediction and recognition models of driving behaviors are often based on machine learning approaches. These models are required for the growth of advanced driving assistance systems. The performance of the model depends on the optimal parameters, hyperparameters, and model structure. In the present study, hyperparameters of a previously developed model (neural network-based state machine model) are optimized for the lane changing recognition. Two methods are considered for the hyperparameter optimization: Bayesian optimization and Genetic algorithm (GA). Three lane changing behaviors are estimated. Real human driving data generated using a driving simulator are used for the parameterization. The aim is to compare the model's recognition performance based on the two methods. Furthermore, comparisons between the models with optimized hyperparameters and the original model (without hyperparameter optimization) are performed. The results show that the performance based on the Bayesian optimization is better than GA, while the original model still outperforms others.

1 Introduction

In recent years, the use of Advanced Driving Assistance Systems (ADAS) have increased due to the systems' ability to realize potential driving behaviors and provide hints in different situations. In situations where hints are not sufficient, a complete takeover by the system is performed to maneuver safely. The driving behavior prediction and recognition models play an integral role in the development of ADAS. Many research contributions have addressed the development of these models and their integration to ADAS [28]. Most models are developed using machine learning (ML)-based methods, such as Artificial Neural Network (ANN) [25] and Support Vector Machine (SVM) [27]. However, developing an optimal model is challenging due to the difficulty of selecting optimal parameters.

A common solution to tackle this issue is by combining two or more ML-based algorithms to develop a prediction and recognition model, as in [20], whereby SVM is combined with ANN to develop a lane changing prediction model. The ANN predicts the trajectory of the vehicle ahead, while the SVM predicts if a lane change will occur based on the predictions of the ANN. Feature selection is another method to tackle this issue. Behavior estimations are dependent on features

K. Baclawski, M. Kozak, K. Bellman, G. D'Aniello, A. Ruvinsky and C. Da Silva Ferreira Barreto (eds.), CogSIMA 2023 (EPiC Series in Computing, vol. 102), pp. 33–43

like environmental features, as driving behaviors are influenced by the driver's characteristics and environmental conditions [11], [8]. Hence, appropriate selection of features is important for developing an optimal model. Optimization of model parameters during the training process, such as using Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [11] is also done to improve the performance. Optimization of hyperparameters in ML-based approaches is also another method [31]. Hyperparameters are parameters that control the learning process of an ML algorithm, thus impacting the selection of optimal model parameters and estimations. Hyperparameter optimization reduces efforts to develop suitable hyperparameters manually and it makes models reproducible [31]. Various methods are used to optimize the hyperparameters, such as Bayesian optimization [17] and random search [15]. In [17], hyperparameters of a Long short-term memory (LSTM) model are optimized using Bayesian optimization for the prediction of aggressive driving behavior, while in [15], hyperparameters of different ML-based models are optimized using random search for the classification of drowsy behavior. Nevertheless, only a few research contributions have considered hyperparameter tuning in the development of driving behavior estimation models, particularly for lane changing behavior estimations.

In this work, an ANN-based state machine model developed in [6] for lane changing behavior recognition is used for the application of the hyperparameter optimization. Two hyperparameter optimization methods are considered: Bayesian optimization and a Genetic algorithm (GA). The objective is to compare the recognition performance between the models developed based on both methods. Comparison to the model without hyperparameter optimization is performed as well. In Section 2, the methodology of the recognition model and the optimization techniques are described. The application of the methods are described in Section 3. In Section 4, the estimation performance based on different hyperparameter optimization methods are presented. Finally, a conclusion is given in Section 5.

2 Methodology

2.1 Artificial Neural Network-Based State Machine Model

In recent years, state machines have been used in various applications for modeling behaviors using discrete states [16]. The state machine can define the switching of system states from one state to another or remain in the same state based on a set of conditions and inputs. In this work, the transition between states or remaining in the same state describes the output of the model. In addition, designers determine the parameters, transition conditions, input, and output variables in this specific model. State machine models posses many benefits such as its flexibility, non-complicated design process, and quick state reachability [9], [1]. Existing literature uses the state machine to model degradation behavior of technical systems [4], plant growth control [23], and driving behaviors [7], [6]. The ANN-based state machine model developed in [6] (as a new ML-based model) is utilized here to realize lane changing behavior recognition. Three lane changing behaviors are defined: lane change to the right (LCR), lane keeping (LK), and lane change to the left (LCL)[11].

The model has three states which represent the different lane changing maneuvers: LCR (State 1), LK (State 2), and LCL (State 3). Each state transitions between each other or remain in the same state for the estimation of lane changing behaviors. Transitions between states or remaining in the same state are based on specific conditions defined using the estimations of individual ANN models (Fig. 1). Thus for each behavior, an ANN model is developed: ANN (right), ANN (keep), and ANN (left) [6]. The estimation of the ANN corresponding to the current estimated state is used to define the transitioning or remaining conditions (Fig. 1).

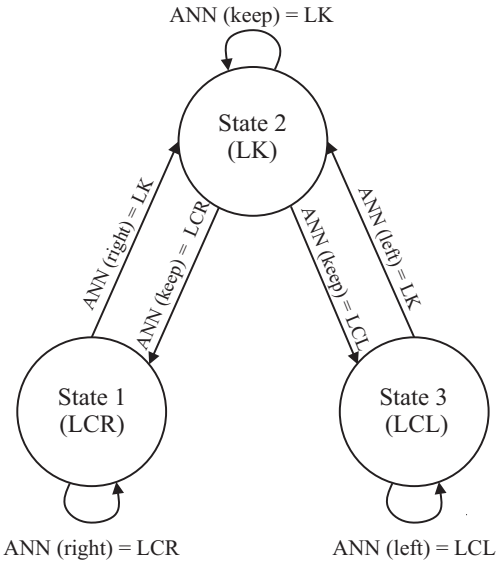

Figure 1: ANN-based state machine topology [6]

Each ANN model has one hidden layer of ten neurons and an output layer of three neurons representing the behaviors. Based on Fig. 1, LK transitions to LCR or LCL, if the estimation of ANN (keep) is LCR or LCL respectively, to estimate the next behavior. While LCR or LCL transitions to LK, if the estimation of ANN (right) and ANN (left) is LK [6]. The state remains in the same state if the estimation of ANN is same as the current state. In Table 1, the possible estimations of the ANN models are presented.

ANN models	Estimations
ANN(right)	LCR, LK
ANN(keep)	LCR, LK, and LCL
ANN(left)	LK, LCL

Table 1: Estimations of different ANN models

3 Model Parameters and Hyperparameter Optimization

Hyperparameters are parameters that control the training process, while model parameters are unknown parameters that affect the prediction and recognition of a model directly. In general, hyperparameters are distinguished into two types: model and algorithm hyperparameters [12]. The model hyperparameters are related to the structure of the model, while the algorithm hyperparameters are related to the way the model is trained [12]. Hyperparameters are used by the training algorithm to develop design parameter values, thus affecting the lane changing behavior recognition performance. Therefore, optimization of the hyperparameters is required to find the optimal configuration of the ML-based model for training. The optimization of model parameters is also required to develop optimal estimations and improved performance. Usually, hyperparameters are optimized prior to training, unlike the model parameters which are optimized during the training process.

In this contribution, the Bayesian optimization and Genetic algorithm (GA) are used for the hyperparameter optimization of the ANN models. The ANN model structure (based on hyperparameter optimization) is then combined with the state machine to develop the proposed model. Subsequently, the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is employed to optimize model parameters, which are used for the recognition of behaviors.

3.1 Bayesian Optimization

Bayesian optimization is an iterative approach for optimizing parameters of an objective function f_1. This technique considers the past evaluations to select the next set of hyperparameter values defined by the acquisition function [29], [24], Bayesian optimization optimizes the acquisition function (cheap function) to develop an optimal f_1 (expensive function). Generally, there are two types of acquisition functions: confident-based criteria and improvement-based criteria [21],[24]. Here, an improvement-based criteria known as 'expected-improvement-per-second-plus' function is chosen. Selecting parameter combinations based on past evaluations, allows the optimization to focus on the parameter space which generates the most accurate validation. Exploration (space of high uncertainty) and exploitation (space with high objective values- area of current best hyperparameter values) are considered when searching the parameter space for the next point [24]. One of the main benefits of this method is it requires less iterations to reach the optimal parameter set [24], [29]. The objective function for this optimization technique shows the measure of loss and is given as

$$f_1 = \sum \frac{1}{n}(y_i \neq \hat{y}_i), \tag{1}$$

whereby n is the total number of observations, y_i is the actual lane changing behavior, and $\hat{y}_i$ is the estimated behavior. The hyperparameters optimized are the number of hidden layer neurons, activation function of the ANN's first layer, learning rate, and number of epochs. These variables are selected as they are known to affect the selection of model parameters [31].

3.2 Genetic Algorithm

The Genetic algorithm is a heuristic optimization method. This technique is a stochastic global search optimization technique that changes the population of individual solutions repeatedly [18]. The solutions are the hyperparameters to be optimized. Two existing solutions can be combined to develop a new solution (known as crossover) [18]. A new solution can also be obtained by making random changes to an individual existing solution (known as mutation) [18]. A similar objective function (known as fitness function) as used in Bayesian optimization is selected here. The fitness function describes how well a solution fits the optimal solution [19], [5]. Thus, the GA process can be described as follows [14]:

1. Initialize the population and generation.
2. Calculate the fitness function for each individual solution.
3. Select individual solutions based on the fitness function.
4. Perform the crossover or mutation between the solutions.
5. Repeat steps (2) to (4) until convergence (or the iteration limit has been reached).

To generate an optimal ANN model, the same hyperparameters used in the Bayesian optimization are considered. Some of the benefits of this method include good parallelization properties, the use of probabilistic rules instead deterministic rules as well as its ability to work with discrete and continuous problems [2], [22].

3.3 Model Parameter Optimization

In the ANN-based state machine model, the design parameters (bias and weights of ANN) affect the recognition performance of the model. Hence, these parameters are defined by optimization during the training process using NSGA-II [10]. The NSGA-II is a well-known method for solving multi-objective problems and for its quick convergence [10]. Three suitable objective functions, each representing the different maneuvers, are minimized through the selection of parameters. Metrics well-known for the evaluation of driving behavior estimations such as accuracy (ACC), detection rate (DR), and false alarm rate (FAR) are used to evaluate the model's performance [26]. The metrics are based on comparisons between the actual and estimated behavior. Hence, the parameter values are optimized such that the model achieves high ACC, DR, and low FAR. In addition, the objective functions are defined using the metrics [7], [6], given as

$$f_2 = (1 - DR_{right}) + FAR_{right}, \tag{2}$$

$$f_3 = (1 - DR_{keep}) + FAR_{keep}, \text{and} \tag{3}$$

$$f_4 = (1 - DR_{left}) + FAR_{left}. \tag{4}$$

4 Application of the Techniques

4.1 Design of experiment

To collect the driving data, a driving simulator ($SCANeR^{TM}$ [3]) is utilized in a laboratory environment at the University of Duisburg-Essen. Real human driving are performed by seven drivers. Each driver performed a manual drive of 40 minutes to generate data for the training process and another 10 minute drive for the test process [11]. A highway environment-based scenario with four lanes in two directions is simulated. The interactive scenario consists of both the ego vehicle and other vehicles. The drivers are able to perform different maneuvers like driving straight and overtaking[11].

4.2 Selection of Data and Data Processing

The input variables for the models (individual neural networks and developed model) only consist of environmental variables. Environmental variables show the relationship between the ego vehicle and surrounding vehicles. While some contributions have considered eye-tracking or physiological variables, previous contributions such as [13] state that environmental variables have a greater impact on the estimation abilities as the driver's decisions mainly depend on environmental factors. There are two groups of environmental variables: state of ego/surrounding vehicles and driver's operational information [11]. Here, variables in the first group are the time to collision to the vehicle in the front TTC_f, back TTC_b, left TTC_{fl}, front right TTC_{fr}, back left TTC_{bl}, and back right TTC_{br}. On the other hand, the later are the angle of steering wheel a_{st}, accelerator pedal position a_{acc}, brake pedal position a_{brake}, indicator i, and current lane

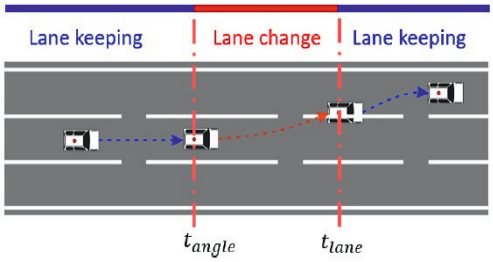

Figure 2: Complete lane change [11]

l [11]. These specific variables are selected as they represent the relationship between vehicles precisely.

The current lane l is defined using the ego vehicle's center point. A lane change is defined when the value of l changes, such that an increase in l indicates LCL, while a decrease indicates LCR. If the value of l remains constant, then LK is denoted [11]. Based on Fig. 2, the interval between the time a lane change occurs (t_{lane}) (when the vehicle crosses over the white lines) and the time of the last significant change in the steering wheel angle (t_{angle}) is defined as the duration of a lane change ($t_{duration}$) [11].

$$t_{duration} = t_{lane} - t_{angle}. \tag{5}$$

4.3 Application of Hyperparameter Optimization Methods

As previously mentioned, the number of hidden layer neurons, the activation function of first layer, learning rates, and the number of epochs are the hyperparameters considered for optimization using both methods to develop the ANN model. The selection ranges of the hyperparameter values for the techniques are detailed in Table 2. Also, the number of iterations used for the Bayesian algorithm is 30 (as it requires fewer number of iterations), while the population and generation sizes for GA are 25 and 100, respectively. Only the training data set is considered for the optimization, thus producing different optimal values for each set. The ANN model structure is based on the optimal number of hidden neurons and the activation function. In contrast, the learning rate and number of epochs only aids the selection of other hyperparameters (in terms of speed). For each data set, the three ANN models are based on the same optimized hyperparameter.

Selected hyperparameters	Ranges
Hidden layer neurons	1-30
Activation functions	Hyperbolic tangent sigmoid (tansig), Log-sigmoid (logsig)
Learning rates	0.001-1
Epochs	100-10000

Table 2: Hyperparameter optimized using both methods

4.4 Training and Test

After the hyperparameter optimization of ANN, the training and test procedures of the proposed model are done in the following manner:

1. Training: Input variables and actual lane changing behaviors are given to the developed model for the recognition of lane changing behaviors. Here, the NSGA-II is used to develop optimized weight and bias values of the three ANN models (with optimized hyperparameters) for generating estimations. The estimations are then used as transition conditions for the development of final estimations of the overall model. The ACC, DR, and FAR are calculated based on the actual and estimated behaviors, which are then used to generate the objective functions. The process is repeated until iteration limits are reached (generation and population sizes: 150 and 90).

2. Test: Based on the optimized parameters, the trained model of a specific driver is tested using the driver's test data to evaluate the model's recognition ability in terms of ACC, DR, and FAR.

5 Results

In this section, the results based on the Bayesian optimization and GA are presented. The optimal hyperparameter values for each individual ANN model for the different data sets are given. Consequently, the lane changing behavior recognition performances of the ANN-based state machine models (based on Bayesian optimization and GA) are given. Finally, comparisons between both models as well as with the model without hyperparameter optimization [6] are presented.

5.1 Hyperparameter Optimization Results

Based on the results of the Bayesian optimization and GA, the hyperparameter values obtained for each data sets are given in Table 3 and Table 4. As mentioned, the data sets used are the training data set of each driver.

Data sets	Hidden layer neurons	Activation functions (first layer)	Learning rates	Epochs
1	18	tansig	0.0211	2812
2	18	tansig	0.0039	2961
3	14	logsig	0.0020	9971
4	13	tansig	0.24218	3167
5	20	logsig	0.0014	2028
6	18	tansig	0.0148	4797
7	20	logisg	0.0272	852

Table 3: Optimal hyperparameter values (Bayesian optimization)

The number of hidden neurons are higher than or equal to 18 based on the Bayesian optimization for most of the data sets, while GA selects values between 11 to 19. The tansig is generally selected as the optimal activation function in both methods, with a few exceptions. High (higher than 1) learning rates averts convergence as the objective function displays divergent behavior (faster training) or it converges faster to a sub optimal solution. Conversely, low

Data sets	Hidden layer neurons	Activation functions (first layer)	Learning rates	Epochs
1	18	logsig	0.2609	7192
2	18	tansig	0.3298	2718
3	19	tansig	0.8791	1725
4	15	tansig	0.1390	8136
5	17	logsig	0.0531	4402
6	11	tansig	0.6680	6776
7	18	tansig	0.2562	5260

Table 4: Optimal hyperparameter values (Genetic Algorithm)

learning rates (lower 0.001) cause slow convergence (slow training) [30]. The obtained results show that the learning rates are generally optimal for all data sets in both methods. Nevertheless, the learning rates are much larger based on GA than Bayesian optimization. The optimal number of epochs required are higher than 800 for both methods. In addition, the computational times to run the Bayesian optimization and GA do not differ by much, whereby the former requires less than 400 seconds, while the latter requires more than 470 seconds (on a standard office PC (2.6 GHz)) . A possible reason the Bayesian optimization is slightly faster is because the Bayesian optimization only updates the posterior causing it to be less computationally expensive.

5.2 Lane Changing Recognition Evaluations

In Table 5, the ACC, DR, and FAR of the recognition models based on both methods and the original model (without hyperparameter optimization) [6] are presented and compared. The original model consists of a hidden layer of 10 neurons, uses the tansig activation function (first layer), 100-200 epochs, and a learning rate of 0.01. The metrics based on three different maneuvers are used to evaluate the models. Here, the average performance based on the test data of seven drivers are presented. For each metric, the model that achieved the highest value and lowest value are highlighted in green and red, respectively.

States	**Metrics**	**Developed models**		
		Bayesian [%]	**GA** [%]	**Original** [%]
Overall	ACC	71.27	68.38	84.55
Right	ACC	80.18	82.30	90.08
	DR	89.48	85.71	78.63
	FAR	20.49	17.75	8.88
Keep	ACC	71.59	68.62	82.88
	DR	68.89	65.76	87.11
	FAR	16.74	15.47	36.99
Left	ACC	90.77	85.84	92.38
	DR	74.78	81.79	42.58
	FAR	8.44	13.82	2.88

Table 5: Average metric values of different models based on seven test data sets

The average lane changing recognition based on both optimization methods shows moderate (60 %-69 %) to high (higher than 80 %) ACC, DR, and low FAR for all maneuvers. The model based on Bayesian optimization generates rather high ACC and DR except for DR_{keep}, with

a value of 68.89 %. High ACC and DR are generated based on GA as well, with the exception of $ACC_{overall}$, ACC_{keep}, and DR_{keep}. The results show that the model with optimized hyperparameters based on Bayesian optimization outperforms the model based on the GA in most metrics, with the exception of ACC_{right}, FAR_{right}, FAR_{keep}, and DR_{left}. A possible reason for this is because the Bayesian optimization takes past evaluations of hyperparameters into consideration, which enables the method to focus on a specific search space to develop the next set of hyperparameter values accurately. Nevertheless, the metric values obtained based on both techniques do not differ significantly. The original model produces high ACC, DR, and low FAR in all the maneuvers as well [6]. Furthermore, the original model outperforms the model based on Bayesian optimization and GA in most metrics, with the exception of DR_{right}, FAR_{keep}, and DR_{left} (for both). This could be due to the non-optimal search region for both optimization techniques. The original model has the most green highlighted values (highest values), while the GA-based model has the most red highlighted values (lowest values).

6 Conclusion

In this contribution, the aim is to compare different hyperparameter techniques on the performance of an ANN-based state machine model [6] for lane changing estimations. The model is described using three states (representing the different maneuvers) that transition between each other or remain in the same state to develop the lane changing estimation. Estimations of three ANN models (representing the different maneuvers) are used as transition conditions in the state machine model to generate the final lane changing estimation. Two methods are considered for the hyperparameter optimization: Bayesian optimization and GA. The hyperparameters optimized are related to the ANN, such as the number of hidden layer neurons, activation function (first layer), learning rate, and epoch number. The obtained hyperparameter values are applied to develop the ANN models in the state machine-based model. The ANN-based state machine is trained with NSGA-II to optimize the model parameters and generate optimal lane changing estimations. Data sets from seven drivers are used to evaluate recognition performance. The results show that the developed model has a better performance with optimized hyperparameters generated by Bayesian optimization than GA for most metrics. Nevertheless, the previously developed model without hyperparameter optimization outperforms the models with hyperparameter optimization. In future, using other optimization methods could improve the model's recognition performance.

7 Acknowledgments

This research is partly supported by the German Academic Exchange Service (DAAD) scholarship received by first author for her Ph.D. study at the Chair of Dynamics and Control, University of Duisburg-Essen, Germany.

References

[1] Paul Adamczyk. The anthology of the finite state machine design patterns. In *The 10th Conference on Pattern Languages of Programs*, 2003.

[2] Ieva Ancveire and Inese Polaka. Application of genetic algorithms for decision-making in project management: A literature review. *Information Technology and Management Science*, 22:22–31, 12 2019.

[3] AV Simulation. Scaner. https://www.avsimulation.com/scaner/.

[4] Nejra Beganovic and Dirk Söffker. Remaining lifetime modeling using state-of-health estimation. *Mechanical Systems and Signal Processing*, 92:107–123, 2017.

[5] Yann Carbonne and Christelle Jacob. Genetic algorithm as machine learning for profiles recognition. In *2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, volume 1, pages 157–166, 2015.

[6] Ruth David, Sandra Rothe, and Dirk Soffker. Lane changing behavior recognition based on artificial neural network-based state machine approach. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3444–3449, 2021.

[7] Ruth David, Sandra Rothe, and Dirk Söffker. State machine approach for lane changing driving behavior recognition. *Automation*, 1(1):68–79, 2020.

[8] Ruth David and Dirk Söffker. Effect of environmental and eye-tracking information: An artificial neural network-based state machine approach for human driver intention recognition. In *2022 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, pages 16–22, 2022.

[9] Giovanni . De Micheli, Robert K. Brayton, and Alberto Sangiovanni-Vincentelli. Optimal state assignment for finite state machines. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 4(3):269–285, 1985.

[10] Kalyanmoy. Deb, Amrit. Pratap, Sameer. Agarwal, and Thirunavukarasu . Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[11] Qi Deng and Dirk Söffker. Improved driving behaviors prediction based on fuzzy logic-hidden markov model (fl-hmm). In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 2003–2008, 2018.

[12] Gonzalo I. Diaz, Achille Fokoue, Giacomo Nannicini, and Horst Samulowitz. An effective algorithm for hyperparameter optimization of neural networks. *IBM J. Res. Dev.*, 61:9:1–9:11, 2017.

[13] Anup Doshi and Mohan Manubhai Trivedi. On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):453–462, 2009.

[14] Russell C. Eberhart, Roy C. Dobbins, Patrick K. Simpson, and Roy W. Dobbins. *Computational Intelligence PC Tools*. AP Professional, 1996.

[15] Farbod Farhangi. Investigating the role of data preprocessing, hyperparameters tuning, and type of machine learning algorithm in the improvement of drowsy eeg signal modeling. *Intelligent Systems with Applications*, 15:200100, 2022.

[16] Arthur Gill. *Introduction to the Theory of Finite-state Machines*. Electronic science series. McGraw-Hill, 1962.

[17] Deva Hema and Kulandasamy Ashok Kumar. Hyperparameter optimization of lstm based driver's aggressive behavior prediction model. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 752–757, 2021.

[18] John H. Holland. Genetic algorithms and the optimal allocation of trials. *SIAM J. Comput.*, 2(2):88–105, jun 1973.

[19] Xavier Hue. Genetic algorithms for optimization: Background and applications. *Edinburgh Parallel Computing Centre, Univ. Edinburgh, Edinburgh, Scotland, Ver*, 1(0), 1997.

[20] Rubén Izquierdo, Ignacio Parra, Jesús Muñoz-Bulnes, David Fernández-Llorca, and Miguel A. Sotelo. Vehicle trajectory and lane change prediction using ann and svm classifiers. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[21] Tinu Theckel Joy, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Hyperparameter tuning for big data using bayesian optimisation. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2574–2579, 2016.

[22] Chien-Ho Ko and Shu-Fan Wang. Ga-based decision support systems for precast production planning. *Automation in Construction*, 19(7):907–916, 2010.

[23] Friederike Kögler and Dirk Söffker. State-based open-loop control of plant growth by means of water stress training. *Agricultural Water Management*, 230:105963, 2020.

[24] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[25] Nolwenn Monot, Xavier Moreau, André Benine-Neto, Audrey Rizzo, and François Aioun. Comparison of rule-based and machine learning methods for lane change detection. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 198–203, 2018.

[26] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2:37–63, 2011.

[27] Ranjeet Singh Tomar, Shekhar Verma, and Geetam Singh Tomar. Svm based trajectory predictions of lane changing vehicles. In *2011 International Conference on Computational Intelligence and Communication Networks*, pages 716–721, 2011.

[28] Wenshuo Wang, Ding Zhao, Wei Han, and Junqiang Xi. A learning-based approach for lane departure warning systems with a personalized driver model. *IEEE Transactions on Vehicular Technology*, 67(10):9145–9157, 2018.

[29] Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on bayesian optimizationb. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

[30] Yanzhao Wu, Ling Liu, Juhyun Bae, Ka-Ho Chow, Arun Iyengar, Calton Pu, Wenqi Wei, Lei Yu, and Qi Zhang. Demystifying learning rate policies for high accuracy training of deep neural networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1971–1980, 2019.

[31] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.