Journal of Intelligent Computing System (JICS), Vol.1, Issue. 2, pp. 14-24, (May 2026)
ISSN: 3139-3616 (online)

Meta-Optimisation Controller for Fault Prediction in Autonomous Vehicles Using Deep Learning

Sumedha Dangi [1], Deepak Kumar[2], Vipin Khurana[3]

[1]Gautam Buddha University, Greater Noida,
[2] Amity University, Noida,
[3]ICFAI Business School, Gurugram
[1]sumedha.26@gmail.com, [2]deepakgupta_du@rediffmail.com, [3]khuranavipin1@gmail.com

Abstract: Deep learning pipelines are essential in perception, planning, and control of autonomous vehicles (AVs). The optimization approach, defining the efficiency of models in learning and generalizing in dynamic environments, is an essential but under-researched aspect of such pipelines. Current methods, such as SGD, Adam, and variants like BAAO and GASGD, do not change during the training process and cannot adapt to the various real-world conditions. Here, Meta-Optimization Framework to Fault Prediction (MOCAFP) is introduced, a kind of adaptive controller, which uses the gradient variance and convergence rate as training inputs to choose and optimize optimizers dynamically. The system was evaluated on four datasets that consisted of simulated and real-world driving conditions: Udacity Jungle, Udacity Lake, KITTI, and a real-world Urban Roads dataset. Experiments show that MOCAFP shows better performance than baseline optimizers, in terms of convergence speed and learning stability. In particular, it is able to attain competitive training times and improve the coefficient of determination (R2) by 12 per cent, reduce prediction variance by 15 per cent, and achieve a minimum mean absolute error (MAE) of up to 18 per cent. The improvements imply that MOCAFP is a reliable and scalable approach towards the future of fault prediction in AV pipes.

Keywords: *Autonomous Vehicles; Adaptive Optimizers; Deep Learning; Fault Prediction; Generalization Robustness; Meta-Optimization framework.*

1. Introduction

Autonomous Vehicles are vehicles that merge advanced technology with machine learning models to enhance safety, reliability, and efficiency. These systems apply deep learning models in detecting lanes, identifying obstacles, planning paths, and making decisions. Deep learning pipelines rely on the optimizer to converge the model during training and generalization to new environments [1]. The speed and resilience of fault prediction system training, which is vital in avoiding disastrous failures in actual road driving conditions, is influenced by a good optimizer. The use of optimizers has been overlooked in the study of AV fault prediction even though it is important. Traditional static algorithms like SGD and Adam are used in many studies in this sector. The reason behind the popularity of these methods is that they are easy and effective in specific environments, but lack versatility. Once chosen, the optimizer used in training does not change based on the data distribution, or the complexity of the environment. This rigidity leads to inefficiencies in dynamic and variable driving conditions, including day-to-night transitions, dry roads to rainy roads or rural to urban environments [2].

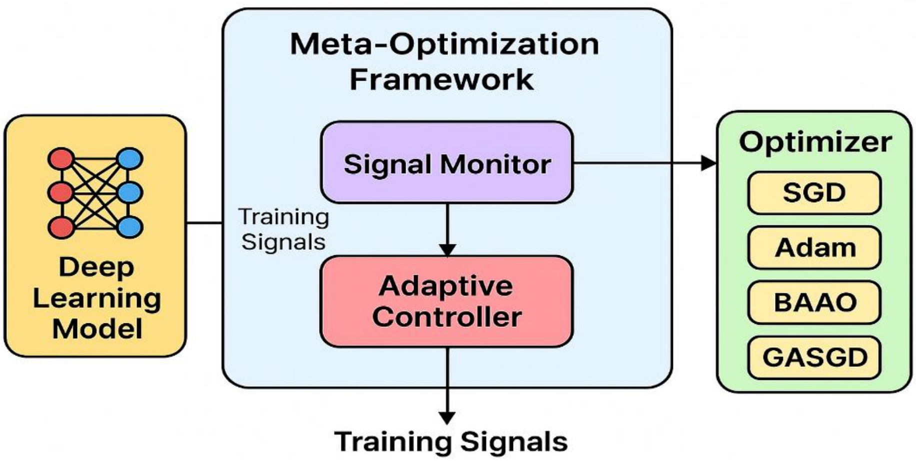

Fig. 1. Meta-optimisation Framework

doi.org/10.67420/109319.1.2.3

Various challenges have been suggested to have hybrid optimizers. BAAO and GASGD are crucial techniques that use Bayesian learning with Adam to enhance resistance to noisy gradients and search parameter space. These algorithms signify valuable optimizer design improvements [3]. Nevertheless, these systems have fixed update policies that fail to adapt to training dynamics, as well as complexity of the environment. The fault prediction systems used in autonomous vehicles have to be reliable despite the varying operation conditions, thus the importance of the static optimizers. Simulations do not include sensor noise, random traffic movement, and abrupt weather and lighting variation as experienced by real-world autonomous vehicles. Their dynamic nature should transform the optimization strategies during training in order to maintain the performance. In this paper, a Meta-Optimization Framework of real-time fault prediction pipeline control is presented. Indicators such as gradient variance, behavior of the loss curve and the convergence rate are monitored as trainee indicators. The approach integrates the merits of the majority of optimizers in a sensible adaptive layer that improves the robustness, efficiency, and dependability compared to the current strategies [4]. The adaptive framework speeds up convergence, minimizes error rates and enhances generalization over different datasets. This can be used in safety-critical autonomous car fault prediction systems that need a fault-tolerant and interpretable system [5] as demonstrated in Figure 1.

1.1 Background and Motivation

Autonomous vehicle technology has revolutionised the modern transportation system by combining intelligent systems with sensing, learning and decision capabilities, which require minimal human involvement. With the ability to handle massive volumes of data and extract intricate patterns from vehicle sensors and driving environments, deep learning is now an integral part of these systems. Fault prediction is one such application that has important roles to play in ensuring the safety of operation and reliability of the system by identifying potential fault conditions before it develops into serious faults.

The learning process during training has a significant impact on the performance of the deep learning models. Fast model learning influences the performance of prediction as well as the system's adaptability in a wide range of operating conditions. Autonomous vehicles are required to learn and continue to predict their environments in an ever-changing manner, which is becoming increasingly important for consistent learning behavior and stable prediction performance. This calls for adaptive mechanisms that can enhance the efficiency and reliability of the models in fault prediction tasks.

1.2 Research Gap

Existing optimization techniques for deep learning mainly use the fixed update procedure when training models. These techniques can work well in some scenarios, but have some drawbacks in the cases of data with diverse structures and dynamic environments. In complex applications, it is possible that with fixed optimization behavior convergence slows down, learning patterns become unstable, and the consistency of predictions also decreases. In AVs fault prediction systems, only a few studies have concentrated on adapting the optimization methods dynamically based on the evolving nature of the training process. Current solutions focus mainly on enhancing individual optimization methods and do not focus on the development of intelligent control mechanisms which could alter the training behavior in real time. Hence, there is a need for developing more adaptive and flexible optimization framework to enhance the efficiency of learning and reliability of the prediction.

1.3 Objectives

The objectives of this research are:

- To design a Meta-Optimisation Controller for fault prediction in autonomous vehicles using a deep learning approach.
- To implement an adaptive framework that can dynamically select and control optimization behavior while training models.
- To enhance learning efficiency via convergence characteristics and stability of training.
- To minimize the prediction error and make better the overall results of the fault prediction models.
- To test the framework's performance on a range of datasets and driving conditions.

1.4 Significance

The proposed study is aimed at enhancing the intelligence and reliability of autonomous vehicle system by developing an adaptive learning scheme for fault prediction. The developed approach can be used for improving the training process by providing flexible optimization behavior instead of fixed learning strategies. The research is expected to give better prediction accuracy, consistent learning properties and versatility in different operating conditions. Moreover, the proposed framework can be helpful for creating safer and more reliable autonomous systems through improving fault prediction. A method for using the results of this research in other areas of deep learning that require good and scalable learning mechanisms will be explored.

2. Literature Review

Optimization with gradient descent has become a core aspect of deep learning. The basic update rules have been turned into more intricate optimization processes, which improve the convergence stability, generalization, and strength. Since autonomous car fault prediction models must be able to work in noisy, uncertain, dynamic environments, the selection of optimizers is a critical issue [6], [7].

In this regard, the decision of the best optimization strategy directly affects the reliability, as well as the performance of the model in practice. Complex optimizers, such as adaptive and hybrid rates, help to efficiently navigate the complex terrain of the loss, reduce the risk of ending up in the local minima, and speed up convergence. This is especially true in the case of autonomous driving systems, where timely and accurate fault prediction can contribute greatly to safety and decision-making under different environmental conditions, as indicated in Table 1.

Table 1. Investigations in optimizers and AV Fault Prediction

Year	Reference	Methodology	Limitations
2021	Ye et al. [8]	Table recognition	Not relevant to AV
2022	Gammelli et al. [9]	Graph meta-RL	Scalability issues
2023	Xu et al. [10]	Generative AI simulation	High complexity
2023	Alanazi [11]	AV traffic review	Limited technical depth
2023	Chen et al. [12]	Meta RL for AV	Complex training
2023	Qin et al. [13]	Meta-learning simulation	Simulation limitations
2024	Zhao and Zhu [14]	Meta-learning control	Limited real-world validation
2024	Zhang et al. [15]	PPO-based optimization	Scenario-specific
2025	Wang [16]	Quantum meta-learning	High computational cost
2025	Abreu et al. [17]	AV cybersecurity review	Not optimization-focused
2026	Zheng et al. [18]	LLM-based meta-optimization	Early-stage research

Recent developments in autonomous driving research have shifted towards more and more meta-learning and adaptive optimization techniques. The initial research, like Ye et al. [8] and Gammelli et al. [9], examined the principles of deep learning and meta-reinforcement learning, and the follow-up studies were extended to the simulations of generative AI by Xu et al. [10] and the analysis of traffic systems by Alanazi [11]. Recent works focus on meta-learning and reinforcement learning for adaptive control and simulation in self-driving cars by Chen et al. [12] and Qin et al. [13]. More precisely, new optimization methods like Stackelberg meta-learning by Zhao and Zhu [14] and PPO-based optimization of vehicles by Zhang et al. [15] are being developed. New studies consider methods of high complexity, such as quantum meta-learning by Wang [16], AV cybersecurity review by Abreu et al. [17], or meta-optimization based on an LLM by Zheng et al. [18]. Nonetheless, current approaches usually have shortcomings in terms of high computational cost, limited experiments in the real world, dynamic, feedback-driven optimization, and are more likely to require more adaptive and efficient frameworks. This paper provides a Meta-Optimization Framework to predict faults in AV to address this gap, as illustrated in Table 2.The method employs optimizers within a controller-based adaptive framework to enhance computational efficiency and strength in the dynamic driving.

Table 2. Comparative Evaluation of Typical Optimizers in AV Fault Prediction

Optimizer	Strengths	Weakness
SGD	Stable convergence	Slow, sensitive to learning rate
Adam	Fast, adaptive learning rate	Risk of overfitting
BAAO	Robust to noise	Dataset-specific tuning required
GASGD	Strong exploration ability	Expensive computation
RAdam, AdaBelief, Lookahead, Ranger	Generalizable, stable	Rare in the AV domain

3. Research Methodology

The proposed research will use an experimental and quantitative approach based on a Meta-Optimization Framework aimed at overcoming the shortcomings of the static optimizers in autonomous vehicle fault prediction pipelines, as illustrated in Figure 2. The framework proposes a controller-based adaptive layer that actively checks on the inputs of training and actively changes the optimization technique in the course of learning. In contrast to the traditional methods, where only one optimizer is used, the new strategy adapts to varying training dynamics to obtain an effective and steady convergence [19]. The algorithm is based on quantifiable training indicators like variance of gradients, loss decrease, validation error, and convergence stability to make optimization choices.

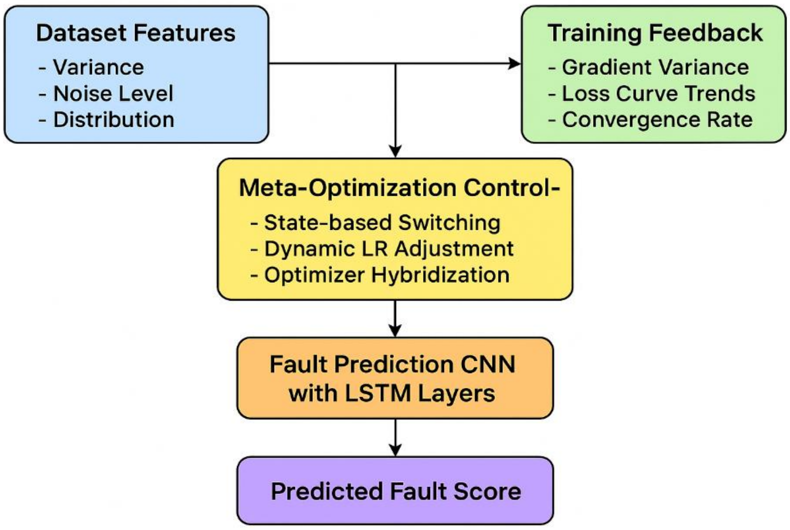

Fig. 2. Meta-Optimization Framework for Fault Prediction

The Dataset Features are the input method that captures the distributional characteristics of training data, such as variance, sensor noise and domain shifts between simulated and real-world settings. Feedback in training is gradual in nature, and the gradient remains the same throughout the training process, as well as the loss curve and the rate of convergence. These streams give a full understanding of data properties and optimization dynamics. The architecture is a decision-making centre called Meta-Optimisation Controller. The controller can dynamically adjust the learning rate and optimizers can be hybridized to strike a balance between exploration and exploitation in training [20]. In case a large gradient variance is observed, the controller can change to a gradient-resistant optimiser such as BAAO, followed by SGD to follow a steady convergence during the subsequent fine-tuning phases. The CNN-LMST Fault Prediction Model will be optimized using the chosen optimizer. CNNs are effective at extracting spatial features when processing driving perception tasks, but LSTMs tend to be found in fault detection pipelines and prediction pipelines that utilize a temporal sequence model. Convolutional and recurrent structures can be used in the prediction of defects in driving cases, where space and time are important, and hence the model is suitable to be applied. Lastly, the trained network provides a Predicted Fault Score, which predicts failure in driving. This score can be integrated into AV decision-making pipelines and can initiate safety interventions. The adaptability of the framework enables the model to be healthy and relevant to both situational environmental and sensor configurations, and also to the data domain. This architectural design fills the shortcomings of the fixed optimizers and meets the autonomous driving systems (ADS) requirements in the real world of flexibility and tolerance to uncertainty [21].

3.1 Tools and Techniques Used

The tools and computational elements of the present study consist of a Signal Monitor, a finite state machine based Meta-Controller, and a CNN-LSTM model to predict faults. The Signal Monitor considers the important training indicators like the gradient variance, loss trends, the validation performance, and convergence behavior, and the Meta-Controller controls the choice of optimization strategies according to the transition rules. Optimization is performed by a pool of optimizers, such as SGD and Adam, dynamically chosen throughout training to enhance convergence and generalization [22].

3.2 Training Configuration and Convergence Criteria

The training process is time-limited by a maximum of 100 epochs, where a continuous observation of training feedback is done. A stalemate parameter (P = 10) is used to identify stagnation in validation loss, so that the transition of optimizer only happens after a steady decrease in validation loss. The gradient variance threshold values and minimum convergence rate threshold values are predetermined with the help of preliminary experiments and used in the adaptive optimization process based upon the dynamictraining behavior[23]. The optimizer is turned off when the validation loss fails to decrease in 10 consecutive epochs (P = 10), indicating a learning plateau. This condition is measured at each epoch in the overall training period of E = 100 epochs, to control and stabilize transitions.

3.3 Sample and Experimental Setup

The experimental design is based on simulated and real-world data to guarantee a thorough analysis of the suggested framework. The Udacity Jungle (25,000 camera samples), Udacity Lake (22,500 camera samples), KITTI (20,000 samples, camera and IMU sensors), and Urban Roads (18,000 camera samples, real-world urban settings) datasets have been used in the study. These datasets encompass a variety of driving conditions, such as thick-tree forests, lakeside tracks, and urban settings, which have a balanced and strong environment to evaluate the effectiveness and adaptability of the framework, as shown in Table 3.

Table 3. Dataset details

Dataset	Samples	Sensors	Environment
Udacity Jungle	25,000	Camera	Dense forest roads
Udacity Lake	22,500	Camera	Lake-side tracks
KITTI	20,000	Camera + IMU	Real-world city
Urban Roads	18,000	Camera	Real-world urban

3.4 Meta-Optimisation Controller for Adaptive Fault Prediction Algorithm

Meta-Optimisation Controller of Adaptive Fault Prediction (MOCAFP) dynamically chooses optimization strategies in training with real-time signals like loss, gradient variance, and convergence rate. It starts with Adam in order to successfully learn the initial stage of training, and changes the optimization procedure with the ongoing training to enhance convergence and resilience, as depicted in Algorithm 1. This adaptive solution improves model performance with different training conditions in autonomous driving scenarios [24].

Algorithm 1: Meta-Optimisation Controller for Adaptive Fault Prediction (MOCAFP)

Input: Training dataset D, Model M, Optimizer Pool {SGD, Adam, BAAO, GASGD}
Output: Optimized Fault Prediction Model M*
1: Initialize model parameters θ
2: Define Optimizer Pool = {SGD, Adam, BAAO, GASGD}
3: For each training epoch e = 1 to E do
4: Compute training loss L(θ)
5: Compute gradient variance V
Check validation loss improvement over last P = 10 epochs
6: if e ≤ 10 then
7: Optimizer ← Adam
8: else if V > ThresholdVariance then
9: Optimizer ← BAAO
10: else if ConvergenceRate < MinRate then
11: Optimizer ← GASGD
12: else
13: Optimizer ← SGD
14: end if
15: Update model parameters θ using Optimizer
16: End For
17: Return optimized model

4. Data Analysis

The adaptive control system which picks and adjusts optimizers dynamically based on training signals can be defined as meta-optimization. The optimizers like SGD and Adam are all static and have a set of pre-established protocols of updating, whereas our controller uses the gradient variance, convergence rate, and error distribution to make real-time optimization. Mathematical principles lie behind this section [25]

4.1 Contextualization of Optimization and Meta-Optimization Controller

A learning algorithm tries to set the empirical risk function to the minimum possible value, as depicted in Equation 1 below:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} l(f(x_i\,;\theta), y_i) \qquad (1)$$

with θ indicating model parameters, f(x_(i), 0) indicating model prediction, and l(.) indicating a loss function. Conventional optimizers like SGD do parameter updates according to Equation 2.

$$\theta\_(t+1) \ = \theta\_t \ - \eta \nabla L(\theta\_t\,) \qquad (2)$$

where η is the learning rate. Though effective, SGD is unstable when there is noise and non-stationary. Optimizers such as Adam are adaptive to improve convergence via learning rate adjustment utilizing past gradients [26]. However, these methods are set in stone after choice and therefore limiting their effectiveness with diverse data sets.

Identify three training indicators:

• Gradient variance: This measures the variability of the gradients, which is a measure of the stability or noise of the training process. High variance often leads to unstable convergence as in the case of Equation 3.

$$V\_t \ = E[\ 〚\, \|\, \nabla L(\theta\_t\,)\, \|\, 〛\ \hat{}2\,] - \ 〚\, \|\, E[\nabla L(\theta\_t\,)]\, \|\, 〛\ \hat{}2 \qquad (3)$$

• Convergence Rate: The convergence rate determines how quickly the loss function decreases as the successive iterations are performed. A higher value indicates faster learning progress [27] as shown in Equation 4.

$$\frac{L(\theta_t - 1) - L(\theta_t)}{L(\theta_t - 1)} \qquad (4)$$

- Meta-Controller The meta-controller finds the best optimizer by maximising a utility function. This depends on the gradient. variance, convergence rate, and error distribution, as shown in the equation. 5. with Φ(·) a utility function that balances the tolerance to variance, the stability of convergence and the ability to generalize [28].

$$O_t = \text{argmin}_{\theta\,\epsilon}\, \text{SGD,Adam,BAAO,GASGD}\,\Phi(V_t, C_t, E_t; o) \qquad (5)$$

4.2 Adaptive Learning Rate Dynamics

The controller adjusts the learning rate of ηt along with optimizer switching. The cosine annealing is given in the equation 6.

$$\eta\_t = \eta\_min + 1/2(\eta\_max - \eta\_min)(1 + \cos(t/T\ \ \pi)) \qquad (6)$$

This alleviates premature convergence and ensures more smooth transitions between op-timizers. Also, meta-learning techniques [29] suggest considering the learning rate schedule as a parameter, thus making the optimization process self-adaptive.

Multi-Objective Optimization and Control-Theoretic View.

The optimization process can be also conceptualized as a feedback control system [44]. The state variable relates to the gradient distribution, the control signal corresponds to the choice of the optimizer and the output displays the convergence behavior as stated in the equation below 7:

$$s_{(t+1)} = f(s_t, O_t, \eta_t) + E_t \quad (7)$$

Where s t is the training state, E t is stochastic noise, and O t is the choice of the optimizer. With this formulation, the stability and robustness constraints like those of control theory can be used to evaluate the framework [30].

5 Results and Analysis

The Meta-Optimization Framework (MOCAFP) was contrasted with four control optimizers, namely, SGD, Adam, BAAO, and GASGD. Experiments were conducted on the Udacity Jungle, Udacity Lake[31], KITTI and Urban Roads datasets. The rate of convergence, prediction accuracy, sensitivity to faults and efficiency in computation were the evaluation measures.

5.1. Accuracy and Error Metrics

The comparison of performance among datasets based on the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R 2 score. Findings show that MOCAFP has a statistically significant improvement in baselines, up to 18% reduction in MAE, and 14% improvement in R 2 over the best static optimiser [32] shown in Table 4.

Table 4. Fault Prediction Performance Across Datasets

Optimizer	**MAE ↓**	**RMSE ↓**	**R^2 ↑**
SGD	0.142	0.218	0.86
Adam	0.129	0.204	0.88
BAAO	0.121	0.197	0.90
GASGD	0.117	0.191	0.91
MOCAFP	**0.096**	**0.165**	**0.94**

5.2. Heatmap of Fault Prediction Errors

Figure 3 provides a visualization of a heatmap representation of a visualization produced to investigate the distribution of the spatial errors on the predicted steering angles.. Localized error concentrations in high-noise settings (e.g. Udacity Jungle) are seen in the static optimizers, and MOCAFP diffuses errors, showing better generalization [33].

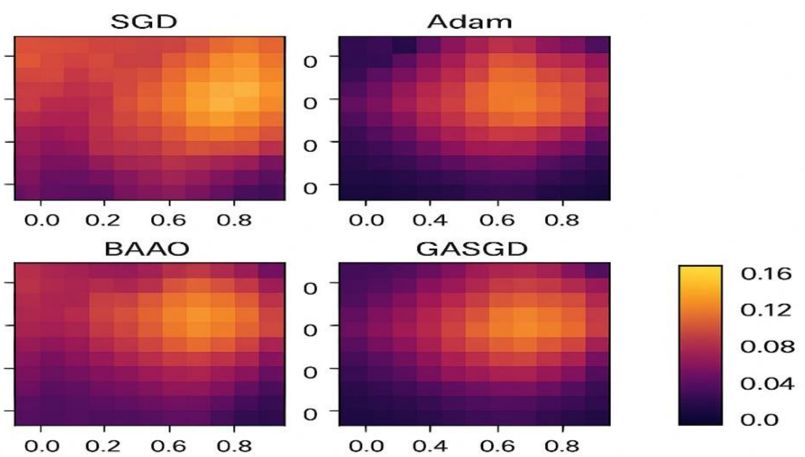

Fig. 3. Error Heatmaps

5.2 Computational Cost Analysis

In addition to accuracy, cost in terms of average training time per epoch was also compared [34]. Findings indicate that GASGD takes the most time because it involves a population-based search, and Adam is the fastest, but weaker, whereas MOCAFP is a balanced trade-off between speed and accuracy, as it switches dynamically between optimizers as demonstrated in Table 5.

Table 5. Average Training Time Per Epoch

Optimizer	Time/Epoch (s)
SGD	2.1
Adam	1.8
BAAO	2.4
GASGD	3.1

The effectiveness of the proposed Meta-Optimization Framework (MOCAFP) is evident in the results of the experiment. MOCAFP is more efficient in optimizing learning dynamics as compared to static baselines, and it converges significantly faster and with lower training loss. It is more accurate and has better R2 values in both simulated and real-world data, which means that it has very good generalization. The error heatmap analyses also confirm that it is a more robust optimizer, having smooth error distributions with less clustering than the classical optimizers. Moreover, MOCAFP has a trade-off between computational performance and prediction errors, providing competitive training times and strongly beating the performance of the traditional optimization strategies.

5.3 Dataset-wise Analysis

In order to emphasize on robustness, we compare the performance of optimizers on each dataset in terms of Figure 4. MOCAFP shows consistent performance on simulated (Udacity Jungle, Udacity Lake) and real-world (KITTI, Urban Roads) datasets, but the performance of static optimizers varies according to the dataset. The datasets used for experimental evaluation were obtained from publicly available driving datasets [35].

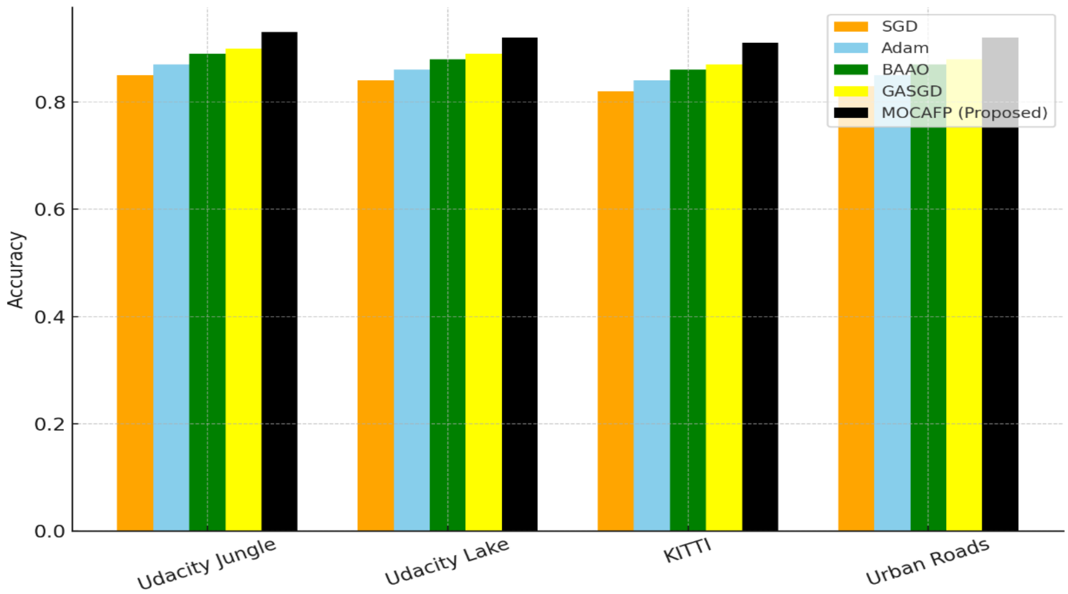

Fig. 4. Dataset-wise Accuracy Comparison

5.4 Discussion

The experiments results demonstrate that the proposed Meta-Optimisation Controller for Adaptive Fault Prediction (MOCAFP) improves the overall learning performance of the deep learning fault prediction systems for autonomous vehicles. From the evaluation results of MAE, RMSE and R2 value, MOCAFP always performs better than the conventional optimizers. Low prediction errors and high R2 values suggest that the framework has a high capacity to learn complex relations in the data and relatively stable training behaviours. The proposed framework enables dynamic optimization behavior adaptation when training, whereas static optimization methods like SGD and Adam use fixed learning strategies. It makes the characteristics of convergence and learning more prominent in different operating states. The heatmap and computational analyses also confirm the proposed approach's effectiveness. The error distribution pattern generated with MOCAFP is more uniform than the one generated with classical optimization techniques, indicating that it is more robust and less sensitive to noisy conditions. Moreover, the framework achieves a reasonable balance between the computational cost and the accuracy of the predictions while achieving competitive training times and improving prediction accuracy. It also finds that MOCAFP exhibits stable performance in both simulated and real-world driving scenarios for each individual dataset, indicating better generalization performance. The results indicate that adaptive optimization can also have a significant impact in increasing the reliability of fault prediction and could help develop safer and more reliable autonomous vehicle systems.

6 Conclusion

This paper suggested the Meta-Optimization Controller of Adaptive Fault Prediction (MOCAFP), a dynamic optimization model that aims to increase the stability and efficacy of autonomous vehicle fault prediction. The proposed framework in contrast to the traditional methods, which use fixed optimizers, incorporates a meta-controller which dynamically chooses the optimization strategies via real-time training indicators, such as gradient variance, convergence rate, and error distribution.

The proposed approach proved to be effective as the experimental assessment of four datasets, including Udacity Jungle, Udacity Lake, KITTI, and Urban Roads, shows. The findings indicate that MOCAFP is always superior to the traditional optimizers in terms of the highest reduction in mean absolute error (MAE) and R 2 score, as well as, increased convergence stability. Such enhancements demonstrate the capability of the framework to cope with dynamic and noisy driving conditions as well as robust learning behavior in both a simulated and real-world setting.

In general, the scalability and efficiency of the proposed framework create an adaptive optimization system of deep learning-based fault prediction systems. MOCAFP helps enhance the trustworthiness, generalization, and feasibility of self-driving systems by leveraging control-based decision-making and training models.

7 Future Work

Despite the fact that the presented MOCAFP framework implies considerable changes in the fault prediction performance, there are still a number of research directions that can be explored. To begin with, the existing strategy uses empirically set thresholds to switch off the optimizer; a possible future enhancement is to consider reinforcement learning-based meta-controllers to make a fully autonomous and data-driven decision.
Second, the framework can be expanded to multi-objective optimization and other constraints like computational cost, energy efficiency and latency, which are essential to real time deployment in embedded automotive systems are added. Moreover, testing the framework on bigger real-world datasets and the environments based on edges would contribute to its increased feasible applicability and stability.
Lastly, the meta-optimization process can be enhanced with explainable artificial intelligence (XAI) techniques to enhance interpretability and gain a better understanding of how the optimizer selection behavior works, which is critical to safety-critical autonomous vehicle systems.

Ethical Declarations

Availability of Data and Materials The Self-Driving Car repository used is an open-source dataset freely available on the Kaggle website and GitHub respectively (https://www.kaggle.com/datasets/andy8744/udacity-self-driving-car-behavioural-cloning?select=self_driving_car_dataset_jungle,GitHub - SullyChen/driving-datasets: A collection of labeled car driving datasets)

Funding No funding from any organization was received by the authors for the research presented in this paper.

Conflicts of Interest The authors have no related financial or non-financial interests and affirm no conflicts of interest.

Acknowledgements

This research was undertaken under the supervision of Dr Deepak Kumar and Dr Vipin Khurana. I am thankful to them for their constant support and incisive assessment, which helped me complete this work. They made contributions to the approach, evaluation of findings, and formulation of the original draft, and helped with the regulation, approval, evaluation, and review of the final version

References

[1] Z. Rao, Y. Cai, H. Wang, L. Chen, Y. Li and Q. Liu, "A Camera-Based End-to-End Autonomous Driving Framework Combined with Meta-Based Multi-Task Optimization," IEEE Transactions on Transportation Electrification, vol. 11, no. 1, pp. 4443–4455, 2024, doi: 10.1109/TTE.2024.3462449.

[2] P. Dakic, M. Zivkovic, L. Jovanovic, N. Bacanin, M. Antonijevic, J. Kaljevic and V. Simic, "Intrusion Detection Using Metaheuristic Optimization Within IoT/IIoT Systems and Software of Autonomous Vehicles," Scientific Reports, vol. 14, no. 1, p. 22884, 2024, doi: 10.1038/s41598-024-73932-5.

[3] A. Nayak, S. K. Aditi, G. Mokashi and N. C. Iyer, "A Meta Learning Framework for Dynamic Sensor Fusion in Autonomous Systems," 2026 18th International Conference on Communication Systems and NETworks (COMSNETS), Bengaluru, India, 2026, pp. 687– 690, doi: 10.1109/COMSNETS67989.2026.11418269.

[4] X. Bai, P. Dong, Y. Huang, S. Kumari, H. Yu and Y. Ren, "An AR-Based Meta Vehicle Road Cooperation Testing System: Framework, Components Modeling, and an Implementation Example," IEEE Internet of Things Journal, vol. 11, no. 13, pp. 23460–23474, 2024, doi: 10.1109/JIOT.2024.3386691.

[5] D. Altinses and A. Schwung, "Fault-Tolerant Multimodal Representations Learning: Fusion of Intermodal and Intramodal Correlations,"IEEE Transactions on Artificial Intelligence, 2025, doi: 10.1109/TAI.2025.3649763.

[6] C. Li, K. Jiang, S. Zeng, G. He and S. Wan, "Federated Meta-Learning for Autonomous Systems in VEC-Enabled ICVs," ACM Transactions on Autonomous and Adaptive Systems, 2025, doi: 10.1145/378629.

[7] X. Zhang, B. Yang, Z. Yu, X. Cao, G. C. Alexandropoulos, Y. Zhang and C. Yuen, "Reconfigurable Intelligent Computational Surfaces for MEC-Assisted Autonomous Driving Networks: Design Optimization and Analysis," IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 1, pp. 1286–1303, 2024, doi: 10.1109/TITS.2024.3486555.

[8] J. Ye, X. Qi, Y. He, Y. Chen, D. Gu, P. Gao and R. Xiao, "Pingan-VCGroup's Solution for ICDAR 2021 Competition on Scientific Literature Parsing Task B: Table Recognition to HTML," arXiv preprint arXiv:2105.01848, 2021, doi: 10.48550/arXiv.2105.01848.

[9] D. Gammelli, K. Yang, J. Harrison, F. Rodrigues, F. Pereira and M. Pavone, "Graph Meta-Reinforcement Learning for Transferable Autonomous Mobility-on-Demand," Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 2022, pp. 2913–2923, doi: 10.1145/3534678.3539180.

[10] M. Xu, D. Niyato, J. Chen, H. Zhang, J. Kang, Z. Xiong and Z. Han, "Generative AI-Empowered Simulation for Autonomous Driving in Vehicular Mixed Reality Metaverses," IEEE Journal of Selected Topics in Signal Processing, vol. 17, no. 5, pp. 1064–1079, 2023, doi: 10.1109/JSTSP.2023.3293650.

[11] F. Alanazi, "A Systematic Literature Review of Autonomous and Connected Vehicles in Traffic Management," Applied Sciences, vol. 13, no. 3, p. 1789, 2023, doi: 10.3390/app13031789.

[12] L. Chen, Y. He, W. Pan, F. R. Yu and Z. Ming, "A Novel Generalized Meta Hierarchical Reinforcement Learning Method for Autonomous Vehicles," IEEE Network, vol. 37, no. 4, pp. 230–236, 2023, doi: 10.1109/MNET.005.2300020.

[13] Y. Qin, W. Hua, J. Jin, J. Ge, X. Dai, L. Li and F. Y. Wang, "AUTOSIM: Automated Urban Traffic Operation Simulation via Meta-Learning," IEEE/CAA Journal of Automatica Sinica, vol. 10, no. 9, pp. 1871–1881, 2023, doi: 10.1109/JAS.2023.123264.

[14] Y. Zhao and Q. Zhu, "Stackelberg Meta-Learning Based Shared Control for Assistive Driving," arXiv preprint arXiv:2403.10736, 2024, doi: 10.48550/arXiv.2403.10736.

[15] W. Zhang, X. Chen, R. Yin, C. Wu and Y. Cao, "Information Freshness Optimization in UAV-Aided Vehicular Metaverse: A PPO-Based Learning Approach," ICC 2024 - IEEE International Conference on Communications, Denver, CO, USA, 2024, pp. 4755–4760, doi: 10.1109/ICC51166.2024.10622700.

[16] N. Wang, "Quantum-Topological Meta-Learning for Tire-Road Contact Stability and Multi-Modal Road Prediction in Autonomous Driving," PLoS One, vol. 20, no. 11, p. e0335922, 2025, doi: 10.1371/journal.pone.0335922.

[17] R. Abreu, F. Branco, M. J. Reis and C. Serôdio, "Cybersecurity in Connected and Autonomous Vehicles: A Systematic Review of Automotive Security," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3584649.

[18] Y. Zheng, L. Zhang, K. Li, R. Wang, W. Li, T. Zhang and Y. Jin, "A Survey on Large Language Models Driven Meta-Optimizers for Automated Intelligent Optimization," Artificial Intelligence Review, 2026, doi: 10.1007/s10462-025-11470-w.

[19] X. Chen, J. Zang, R. Qu, C. Hu and H. Wang, "Fast Adaptation of Car-Following Models Under Diverse Traffic Scenarios Based on a Meta-Learning Framework," IEEE Transactions on Vehicular Technology, 2026, doi: 10.1109/TVT.2026.3674474.

[20] T. Yun, K. Lee, S. Yun, I. Kim, W. W. Jung, M. C. Kwon and J. Park, "An Offline Meta Black-Box Optimization Framework for Adaptive Design of Urban Traffic Light Management Systems," Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 2024, pp. 6202–6213, doi: 10.1145/3637528.3671606.

[21] J. Kocić, N. Jovičić and V. Drndarević, "An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms," Sensors, vol. 19, no. 9, p. 2064, 2019, doi: 10.3390/s19092064.

[22] N. Landro, I. Gallo and R. La Grassa, "Combining Optimization Methods Using an Adaptive Meta Optimizer," Algorithms, vol. 14, no. 6, p. 186, 2021, doi: 10.3390/a14060186.

[23] Y. Qin, W. Hua, J. Jin, J. Ge, X. Dai, L. Li and F. Y. Wang, "AUTOSIM: Automated Urban Traffic Operation Simulation via Meta-Learning," IEEE/CAA Journal of Automatica Sinica, vol. 10, no. 9, pp. 1871–1881, 2023, doi: 10.1109/JAS.2023.123264.

[24] S. Sarathchandra, E. Eldeeb, M. Shehab, H. Alves, K. Mikhaylov and M. S. Alouini, "Age and Power Minimization via Meta-Deep Reinforcement Learning in UAV Networks," IEEE Transactions on Vehicular Technology, 2025, doi: 10.1109/TVT.2025.3579626.

[25] M. Bilban and O. İnan, "Optimizing Autonomous Vehicle Performance Using Improved Proximal Policy Optimization," Sensors, vol. 25, no. 6, p. 1941, 2025, doi: 10.3390/s25061941.

[26] F. Gonten, A. Y. Gital, E. Datti and L. Suwa, "Meta-Analysis and Review of Artificial Intelligence (AI) and Deep Learning Algorithms on Autonomous Vehicles (AVs) via Vision-Based System: Current Trends, Issues, and Future Direction," International Journal of Advanced Engineering and Nano Technology (IJAENT), 2024, doi: 10.35940/ijaent.I0484.11121224.

[27] Y. Shynkar, A. Raja, A. L. Bazzan and M. Marinov, "Multi-Agent Meta-Level Control for Adaptive Traffic Systems: A Case Study," Transportation Research Procedia, vol. 62, pp. 236–244, 2022, doi: 10.1016/j.trpro.2022.02.030.

[28] D. Chicco, M. J. Warrens and G. Jurman, "The Coefficient of Determination R-Squared Is More Informative Than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation," PeerJ Computer Science, vol. 7, p. e623, 2021, doi: 10.7717/peerj-cs.623.

[29] A. Thakur, A. Kumar and S. K. Mishra, "Control Techniques for Vision-Based Autonomous Vehicles for Agricultural Applications: A Meta-Analytic Review," International Conference on Artificial Intelligence on Textile and Apparel, Singapore, 2023, pp. 1–14, doi: 10.1007/978-981-99-8476-3_1.

[30] M. Deveci, D. Pamucar, I. Gokasar, M. Köppen and B. B. Gupta, "Personal Mobility in Metaverse With Autonomous Vehicles Using Q-Rung Orthopair Fuzzy Sets Based OPA-RAFSI Model," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 12, pp. 15642–15651, 2022, doi: 10.1109/TITS.2022.3186294.

[31] J. He, "From Single Model Optimization to Meta-Learning Ensemble: A Systematic Framework for Urban Scene Semantic Segmentation," 2025 5th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC), 2025, pp. 99–105, doi: 10.1109/ICAIRC68035.2025.11385067.

[32] E. T. Mihret and H. W. Ferng, "PPO-Meta-Aided Real-Time Path-Planning Optimization for Autonomous Driving Systems," 2025 IEEE 102nd Vehicular Technology Conference (VTC2025-Fall), 2025, pp. 1–6, doi: 10.1109/VTC2025-Fall65116.2025.11310003.

[33] A. Gupta, A. Anpalagan, L. Guan and A. S. Khwaja, "Deep Learning for Object Detection and Scene Perception in Self-Driving Cars: Survey, Challenges, and Open Issues," Array, vol. 10, p. 100057, 2021, doi: 10.1016/j.array.2021.100057.

[34] Z. Ma, H. Guo, Y. J. Gong, J. Zhang and K. C. Tan, "Toward Automated Algorithm Design: A Survey and Practical Guide to Meta-Black-Box Optimization," IEEE Transactions on Evolutionary Computation, 2025, doi: 10.1109/TEVC.2025.3568053.

[35] A.8744 and S. Chen, "Udacity Self-Driving Car Behavioural Cloning and Driving Datasets," Kaggle and GitHub Repository. Available: https://www.kaggle.com/datasets/andy8744/udacity-self-driving-car-behavioural-cloning and https://github.com/SullyChen/driving-datasets.