Vol. 2, No. 2; Apr – Jun (2023)

Quing: International Journal of Innovative Research in Science and Engineering

Available at https://quingpublications.com/journals/ijirse

Artificial Intelligence and Machine Learning in Software Development

Dr. A. Ramalingam

Professor, Department of Master of Computer Applications, Sri Manakula Vinayagar Engineering College, Pondicherry, IND.

J. Aasik Milan★

Student, Department of Master of Computer Applications, Sri Manakula Vinayagar Engineering College, Pondicherry, IND.

M. Anbu Mani

Student, Department of Master of Computer Applications, Sri Manakula Vinayagar Engineering College, Pondicherry, IND.

ARTICLE INFO

Received: 07-05-2023
Received in revised form: 13-06-2023
Accepted: 15-06-2023
Available online: 30-06-2023

Keywords:

Artificial Intelligence;
Challenges;
Machine Learning;
Practices;
Software Engineering;
Software Development.

ABSTRACT

In recent times, there has been a growing trend in software development towards the incorporation of artificial intelligence (AI) and machine learning (ML). AI and ML technologies have gained significant traction across various industries, as businesses aim to enhance their offerings and expand their product portfolios. However, the development of AI/ML systems presents unique engineering challenges distinct from traditional software development. This study seeks to examine the application of software engineering (SE) practices in the development of AI/ML systems and explore the challenges and practices that align with the requirements of industry professionals.

DOI: https://doi.org/10.54368/qijirse.2.2.0007

1.0 INTRODUCTION

Recent advancements in cloud computing, big data management, algorithms, and hardware have opened up a range of possibilities for organizations, industries, and societies to leverage artificial intelligence (AI) (Nascimento *et al.,* 2020). This has led to the widespread adoption of AI, particularly machine learning (ML) systems, by businesses worldwide, as they see them as valuable propositions to enhance and expand their product offerings. The transition of AI/ML systems from laboratory environments to commercial settings is gaining momentum, aiming to harness the vast amount of available data. With the growing popularity of commercial AI/ML, a significant amount of

★ *Corresponding author's e-mail:* aasikmilanmca@gmail.com (J. Aasik Milan)

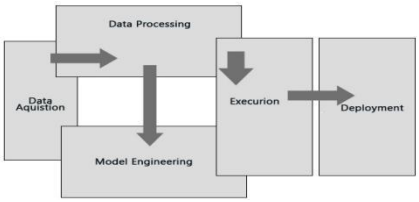
research has been conducted to understand the processes, practices, and challenges faced by professionals in various industries.

2.0 MACHINE LEARNING IN SOFTWARE DEVELOPMENT

Machine learning (ML) has the potential to revolutionize traditional software development practices. By enabling systems to operate autonomously, ML can be utilized by developers for a range of tasks such as code optimization, testing, and deployment. Automating certain phases of software development can free up developers' time for more productive activities. Furthermore, the integration of AI and ML can assist in generating code when provided with appropriate requirements and inputs (Braiek and Khomh, 2020).

Figure 1

Machine Learning Architecture



3.0 ROLE OF AI IN SOFTWARE DEVELOPMENT

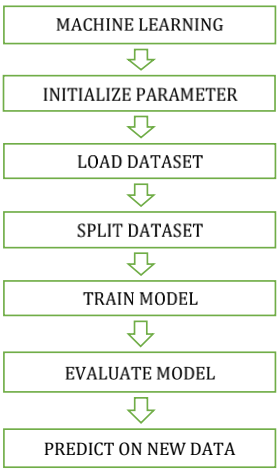
Prominent Software-as-a-Service (SaaS) companies such as Netflix, Spotify, Amazon Prime, YouTube, and others have extensively embraced the use of AI within their platforms to meet customer demands and deliver satisfactory services to their users (Chakravarty, 2010). This trend reflects a notable shift towards integrating AI into software development, indicating that AI indeed represents the future of software development.

4.0 AI/ML SOFTWARE SYSTEM

A simplified explanation of artificial intelligence (AI) is the field of study focused on replicating human cognitive abilities on a computer. In this context, AI refers to machines that imitate human-like "cognitive" capabilities such as learning and problem-solving. AI systems comprise modules that provide opportunities to facilitate various types of learning.

Figure 2

Machine Learning Pseudo-code

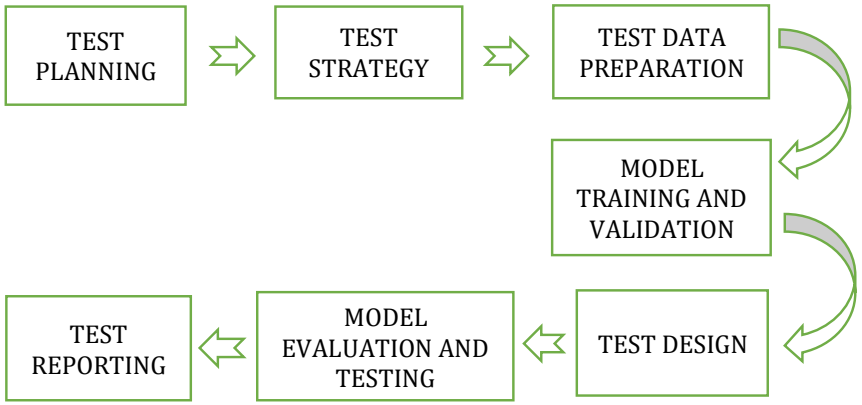


5.0 SOFTWARE TESTING

Testing involves the process of verifying the behaviour of AI software or ML models to ensure they perform as expected. The validation of module behaviour using ML, deep learning, recurrent neural networks, and other techniques is emphasized in numerous primary research studies. Traditional testing methods and approaches that focus on achieving comprehensive structural coverage for exploring various software states are not effective for DL systems. The current understanding of bug characteristics found in ML systems is limited, highlighting the necessity for systematic and efficient bug detection and resolution methods. Testing challenges in the context of AI/ML encompass bug identification, test case generation, data testing, debugging, and defining test criteria (Nascimento *et al.,* 2020; Bryson and Winfield, 2017; Fernandez *et al.,* 2011; Kim *et al.,* 2019). For the purpose of developing and verifying AI and ML models, high-quality and varied test data is essential. In order to evaluate the model's robustness and generalization skills, make sure your test data includes a wide variety of scenarios, including edge cases and outliers. Software for AI and machine learning needs specialized testing methods in addition to conventional functional and non-functional testing. This involves evaluating performance across various hardware configurations, testing against known adversarial attacks, and testing various input data distributions. Set up assessment measures that are appropriate for your ML and AI models.

Figure 3

Software Testing



6.0 AI SOFTWARE QUALITY

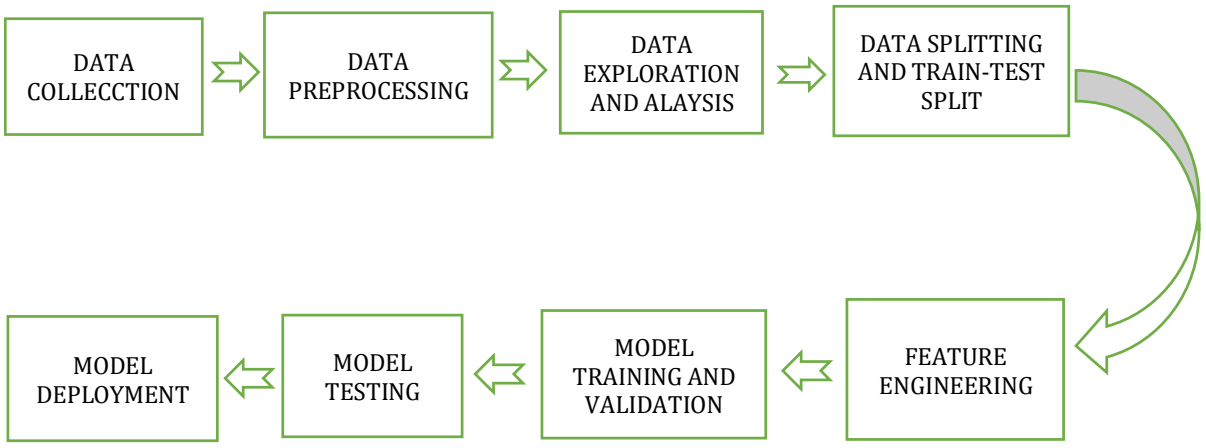
Ensuring the performance, accuracy, and reliability of AI and ML software, as well as the accompanying software components, is crucial. However, the absence of formal ethical requirements in AI projects and developers' limited knowledge or awareness of ethics implementation necessitate the use of practical approaches (Li and Li, 2011). The training of accurate and reliable AI models relies heavily on high-quality data. Careful selection, cleaning, and preprocessing of data are vital to mitigate noise, handle missing values, and address potential biases. Rigorous validation and evaluation processes are necessary to assess the effectiveness of AI models. Metrics such as accuracy, precision, recall, F1-score, mean average precision (MAP), or area under the receiver operating characteristic curve (AUC-ROC) are recommended for evaluating model effectiveness. The particular challenge and the required evaluation criteria influence the use of metrics. To create accurate and impartial AI and ML models, high-quality and varied training data is necessary. Make that the data is impartial, accurately labelled, and representative. Verify the data's correctness, consistency, and completeness. To create accurate and impartial AI and ML models, high-quality and varied training data is necessary. Make that the data is impartial, accurately labelled, and representative.

7.0 HANDLING OF DATA

The performance and efficacy of AI models are directly impacted by the quality, availability, and proper handling of data, making data management an essential component in the development of AI and ML software. Establish procedures for gathering and acquiring the essential data after determining the data requirements for your AI or ML project. This could entail data generation, data acquisition, data purchase using APIs, or data scraping. Ensure the accuracy and dependability of the data gathered. Preprocess data by handling missing values and performing operations including cleaning, filtering, deduplication, and normalization. Implement that ensure the safeguards in place to ensure the availability, integrity, and confidentiality of data (Cruzes and Dybå, 2011). To protect sensitive information, use the proper access controls, encryption strategies, and anonymization procedures. abide by data privacy laws. Maintain adequate security precautions by storing your data in databases or data lakes in an organized manner (Gilpin *et al.,* 2018). Particularly when working with sensitive or personally identifiable information, take data governance and compliance regulations into account. Your data handling infrastructure may need to be scaled as your AI or ML solution develops. Systems for distributed processing and storage. To guarantee the accuracy and usefulness of your data, periodically examine and update it. Through a variety of algorithms and optimisation strategies, the model discovers patterns and correlations within the data.

Figure 4

Handling of Data



8.0 MODEL DEVELOPMENT

Model development is a fundamental component of AI and machine learning software development. It entails creating, training, assessing, and deploying machine learning models to solve specific problems or complete specified tasks. The issue you're trying to address or the job you want the model to do should be made very clear. Determine the problem's goals, demands, and restrictions. This stage aids in selecting the best machine learning (ML) methods and model architecture for the task. Give your AI or ML model a clear description of the issue you're trying to solve or the job you want it to carry out. It may be fraud detection, sentiment analysis, picture categorization, recommendation systems, or any other particular goal (Kim *et al.,* 2019). To prepare the data for training, clean and preprocess it. This could entail dealing with missing numbers, scaling or normalizing numerical aspects, tokenizing language, or formatting data appropriately. Select the model type that best fits your issue and data. There are many different machine learning (ML) algorithms and models available, including ensemble approaches, support vector machines, decision trees, and neural networks.

9.0 PROJECT MANAGEMENT

In order to implement AI and ML projects in software development successfully, project management is essential. To design models, one needs to have practical experience with tools, libraries, and algorithms. That incorporate data collecting, preprocessing, labelling, storage, and security into your data management procedures. Establish data governance standards and make sure that privacy laws are followed. Create systems for team cooperation and data exchange. Encourage clear and efficient communication among team members, clients, and stakeholders. To make sure everyone is on the same page, schedule frequent meetings, status updates, and progress reports. To encourage communication and information exchange, use collaborative tools and platforms. The project's objectives, goals, and deliverable should be clearly stated. Recognize the issue description, solution's parameters, and success standards. Identify the stakeholders and the intended audience. Create a thorough project plan that involves specifying the tasks, deadlines, and dependencies for the project. Make a plan for the tasks of data preparation, labelling, and annotation. Allocate the funds and other resources required to support the AI/ML project, such as the computer environment and software (Tian *et al.,* 2018). Create effective validation and testing procedures to guarantee the accuracy and dependability of the AI/ML system. Utilizing the proper assessment metrics and approaches, carry out unit testing, integration testing, and model evaluation.

10.0 INFRASTRUCTURE

In order to handle the process of AI and ML, infrastructure is a process of hardware and software development. The difficulties include finding, setting up, configuring, and maintaining the infrastructure required for the creation and use of AI/ML systems. The infrastructure has such process that are hardware, software, storage and framework, distributed computing, deployment, inference, monitoring, and management of cloud services. "Despite having access to a wide range of tools, data scientists sometimes struggle to find the appropriate ones because general tools are ineffective for solving their particular challenges. They struggle to keep up with the latest technologies since they also have other obligations and only infrequently work in data science" it was sent by Kim *et al.,* (2017). AI/ML models need a lot of computer power to be trained. To hasten the training process, use infrastructure that allows dispersed training. Create the infrastructure needed for data preparation jobs including feature engineering, cleaning, and transformation. Infrastructure is required once models have been trained in order to serve predictions or conduct inferences on fresh data to support continuous integration and deployment of AI/ML models, adopt solid DevOps practise. Manage the versions of your model and code using version control tools. To assure repeatability and simplify the deployment pipeline, automate the build, test, and deployment procedures.

11.0 ENGINEERING REQUIREMENTS

Several important factors need to be taken into account when determining the technical requirements for artificial intelligence (AI) and machine learning (ML) in software development. These specifications aid in ensuring that AI/ML systems are created and implemented efficiently and that they satisfy the necessary performance, dependability, and scalability standards. Systems for AI and machine learning must be performance and scale-able, especially when working with huge datasets and intricate models. Designing effective algorithms and systems that can manage rising data quantities and computing demands is part of the engineering needs. Systems for AI and ML rely largely on reliable data. Mechanisms for efficient data collection, archiving, cleaning, and

preprocessing should be part of the engineering process. Determining data gathering methods, guaranteeing data accuracy, dealing with missing data, and implementing data management and versioning procedures are all part of this process.

In order to input relevant characteristics into the AI/ML models, relevant features must be extracted from the acquired data and transformed. The model's performance may be enhanced by engineering additional features, performing data normalization, handling categorical data, and identifying the suitable features. Based on the objectives of the issue, the data that is accessible, and performance indicators, the engineering team must evaluate and choose the most suitable models.

12.0 ARCHITECTURE DESIGN

A number of factors need to be taken into account while building the architecture for AI (Artificial Intelligence) and ML (Machine Learning) in software development to ensure an effective and efficient system. Create a reliable data pipeline to facilitate data collection, preparation, and transformation. This pipeline should have components for feature extraction, data cleansing, storage, and ingestion. Make sure your data collection, preparation, and transformation processes are reliable. Mechanisms for data intake, storage, cleansing, and feature extraction should be part of this pipeline. The gathering, preparation, and transformation of data may be accomplished by establishing a reliable data pipeline. The data intake, storage, cleaning, and feature extraction components of this pipeline should all be present. Components for consuming and storing data from diverse sources should be included in the design. Modules for data cleansing, normalization, encoding categorical variables, managing missing values, and producing pertinent features should be included in the architecture.

13.0 MODEL DEPLOYMENT

A crucial phase of AI and ML software development is model deployment, during which trained models are made accessible for usage in production settings. When moving trained models from a testing environment to an operational one, there is a paucity of bench marking knowledge on the migration and quantization procedures, including the effects on prediction accuracy and other factors. Create a framework that is scalable and effective for putting trained models into use. In order to enable applications to communicate with the models, this often entails developing APIs (Application Programming Interfaces) or micro services. For deploying AI/ML models, containerization methods like Docker are frequently employed. Containers isolate the model and its dependencies, guaranteeing consistency in a variety of settings. They offer portability, enabling the model to function without interruption on a variety of systems and infrastructures. AI/ML model deployment as micro-services gives scalability and flexibility. Each model may be contained within a distinct micro-service, allowing for autonomous scalability, deployment, and management.

14.0 INTEGRATION

Building new applications with AI/ML capability or integrating AI/ML capabilities into already-existing software systems is the practise of integrating AI (Artificial Intelligence) and ML (Machine Learning) into software development. For ML algorithms to learn well, the data must be organized and tidy. To preprocess and clean up raw data, managing missing values, outliers, and other data inconsistencies, AI and ML approaches can be used. Before training ML models, this integration aids in confirming the quality of the data. Automation of feature engineering chores is possible using

AI. The process of feature engineering is converting unstructured data into a form that ML systems can use. By automating the extraction, selection, and production of pertinent features, developers may speed up and streamline the development process.AI may help in the process of choosing the best ML model for a particular issue.

14.1 How are SE Practices Modified to Address Technical Issues Unique to ML?

By adding extra approaches and concerns, software engineering (SE) practices are modified to address the engineering problems unique to machine learning (ML). Iterative and exploratory approaches are frequently used in ML projects to create requirements. The dynamic nature of ML models and the requirement for data availability, quality, and relevance to particular use cases must be taken into account by SE practices. To properly capture and fulfil requirements, close cooperation between domain specialists and ML practitioners is essential. In order to investigate various models, methods, and data formats, ML initiatives require prototype and experimenting. Rapid prototyping is made possible by adapting SE practices, for as by using Jupyter notebooks or other interactive development environments. Since ML significantly relies on data, data management becomes essential. Strategies for data collection, storage, preprocessing, and labelling must be included in SE practises. It involves maintaining data quality, dealing with missing values, effectively managing enormous datasets, and taking data privacy and security concerns into account.ML models change over time, therefore it's critical to manage model updates and related information.

14.2 What SE Practices are Mentioned in the Initial Studies?

Initial ML-related papers mention several software engineering (SE) techniques. These studies emphasize the significance of modifying current SE practices and implementing fresh practices to handle the particular difficulties presented by ML initiatives. Applying version control tools, like Git, to ML code, data, and models to regulate changes, keep track of revisions, and promote teamwork. Putting a focus on the need of reproducibility in ML efforts by archiving and sharing code, data, and experiment setups to allow others to duplicate and verify findings. Using ML-specific testing methods, such as cross-validation, holdout sets, and evaluation metrics to evaluate model performance and validate findings. It is essential to accurately identify and analyse the needs for AI and ML systems. This entails comprehending the issue at hand, locating the relevant parties, and extracting both functional and non-functional needs unique to AI and ML components. In order to reduce the risk of integration problems and enable quicker iterations, frequent integration of ML code and automated deployment pipelines guarantee that changes are tested, integrated, and deployed effectively. In order to align expectations and facilitate collaboration among stakeholders, this requires creating cross-functional teams, promoting knowledge sharing, easing communication between data scientists and software developers, and goals.

14.3 What Kind of Empirical Techniques are Employed?

Data is essential to the training and validation of AI and ML systems. Empirical techniques entail gathering pertinent information from a variety of sources, such sensor readings, user interactions, or pre-existing datasets. To find patterns, correlations, and insights, this data is then pre-processed, cleansed, and analysed. Evaluation metrics for AI and ML systems are defined and quantified using empirical methodologies. These measures, such as accuracy, precision, recall, F1-score, area under the curve (AUC), or mean squared error (MSE), evaluate the effectiveness and

calibre of the created models. A research method is a methodology used to collect data, whereas a research strategy is a more general phrase that may incorporate the employment of one or more methodologies. Case studies entail in-depth analyses of actual AI and ML initiatives. They shed light on the difficulties, procedures, and effects of using AI and ML in software development. Typically, qualitative and quantitative data are gathered for case studies through observations, interviews, and document analysis.

15.0 DISCUSSION

The field of software development has seen a substantial transformation thanks to AI (Artificial Intelligence) and MI (Machine Intelligence). They have opened up new opportunities, increased effectiveness, and allowed for the creation of intelligent systems. Different software development jobs have been automated by AI and MI technologies, improving workflows and increasing effectiveness. Intelligent algorithms may automate processes like code generation, debugging, testing, and deployment, saving time and effort for humans. Software developers use machine learning (ML) methods to train models that can identify patterns, categorise data, and make predictions. Anomaly detection, recommendation systems, fraud detection, and picture or speech recognition are just a few applications for machine learning.

15.1 Professionals Working on the Creation of AI/ML Systems must Overcome Challenges?

The field of AI and ML in software development presents a number of obstacles for experts developing AI/ML systems. High-quality data are essential for the training and validation of AI/ML models. The data that professionals have access to has to be accurate, pertinent, and indicative of the issue domain. Data accessibility may be a problem as well, particularly in specialized or specialized sectors where there may be little data accessible.

16.0 CONCLUSION

The industry has undergone a transformation as a result of the inclusion of artificial intelligence (AI) and machine learning (ML) in software development, as well as changes in how we create and use software systems. Numerous advantages and developments have been brought about by AI and ML technologies, including increased effectiveness, improved user experiences, and the capacity to resolve challenging issues. In conclusion, by automating processes, delivering personalised experiences, and allowing data-driven decision-making, AI and ML have completely transformed the software development industry. Although there are many advantages to combining AI and ML, it is important to manage the related ethical issues and make sure that these technologies are developed and used responsibly. Advances in AI and ML will continue to influence the development of software, creating new opportunities and spurring innovation.

REFERENCES

Braiek, H. B., & Khomh, F. (2020). On Testing Machine Learning Programs. *Journal of Systems and Software, 164*, 110542. https://doi.org/10.1016/j.jss.2020.110542

Bryson, J., & Winfield, A. (2017). Standardizing Ethical Design for Artificial Intelligence and Autonomous Systems. *Computer, 50*(5), 116-119. https://doi.org/10.1109/MC.2017.154

Chakravarty, A. (2010). Stress testing an AI based web service: A case study. In proceedings of the *2010 Seventh International Conference on Information Technology: New Generations*, Las Vegas, NV, USA, 1004-1008. https://doi.org/10.1109/ITNG.2010.149

Cruzes, D. S., & Dybå, T. (2011). Recommended steps for thematic synthesis in software engineering. In proceedings of the *2011 International Symposium on Empirical Software Engineering and Measurement*, Banff, AB, Canada, 275-284. https://doi.org/10.1109/esem.2011.36

Damasceno, L., Werneck, V. M. B., Schots, M. (2018). Metric-based evaluation of multiagent systems models. In proceedings of the *Tenth International Conference on Software Engineering*, 67-74. https://doi.org/10.1145/3193954.3193960

Fernandez, A, Insfran, E, & Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology, 35*(8), 789-817. https://doi.org/10.1016/j.infsof.2011.02.007

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In proceedings of the *2018 IEEE Fifth International Conference on Data Science and Advanced Analytics (DSAA)*, Turin, Italy. https://doi.org/10.1109/DSAA.2018.00018

Karunamurthy, A., Kulunthan, K., Dhivya, P., Vickson, A. V. S., (2022). A Knowledge Discovery Based System Predicting Modelling for Heart Disease with Machine Learning. *Quing: International Journal of Innovative Research in Science and Engineering, 1*(1), 14-22. https://doi.org/10.54368/qijirse.1.1.0005

Karunamurthy, A., Yuvaraj, M., Shahithya, J., & Thenmozhi, V. (2023). Cloud Database: Empowering Scalable and Flexible Data Management. *Quing: International Journal of Innovative Research in Science and Engineering, 2*(1), 1-23. https://doi.org/10.54368/qijirse.2.1.0007

Kim, J., Feldt, R., & Yoo, S. (2019). Guiding Deep Learning System Testing Using Surprise Adequacy. In proceedings of the *2019 IEEE/ACM 41st International Conference on Software Engineering*, 1039–1049. https://doi.org/10.1109/ICSE.2019.00108

Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2017). Data Scientists in Software Teams: State of the Art and Challenges. In proceedings of the *IEE Transactions on Software Engineering*, 1-17. https://web.cs.ucla.edu/~miryung/Publications/tse2017-datascientists.pdf

Li, C., & Li, K. (2011). A Practical Framework for Agent-Based Hybrid Intelligent Systems. In proceedings of the *2011 Seventh International Conference on Computational Intelligence and Security*, 199–203. https://doi.org/10.1109/CIS.2011.52

Nascimento, E., Nguyen-Duc, A., Sundbø, I., & Conte, T. (2020). *Software engineering for artificial intelligence and machine learning software: A systematic literature review*. Available at https://doi.org/10.48550/arXiv.2011.03751

Tian, Y., Pei, K., Jana, S., & Ray, B. (2018). DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars. In proceedings of the *40th International Conference on Software Engineering*, 303-314. https://doi.org/10.1145/3180155.3180220