Make AI Testing Meaningful

From Understanding to Mastering of AI Testing

Dr. Ulrich Bodenhausen, Vector Consulting Services GmbH, Stuttgart

Abstract
Application of AI is an important mission for the automotive industry. However, testing of AI is not yet a widespread and well understood discipline. Testing an AI driven system means testing 1.) that the code is correct, 2.) what has been learned from the training cases, and most important 3.) what has not been learned. Machine Learning research relating to this topic is investigating the generalization ability of learning systems, meaning the generalizing across the training patterns and not just memorizing them. However, even state-of-the-art research can currently not provide a concise explanation of generalization ability nor a practical measure. To bridge the gap in the short term, we propose a Deep AI Test scheme to combine important principles: Independence of test data, frequent full testing, selective extension of the data base and test-time data augmentation. The principles should be spread in the automotive industry, become part of standards (especially for self-learning vehicles) and be supported by capability assessment models.

Artificial Intelligence (AI)
Artificial Intelligence (AI) offers great potential for automotive development because it offers the opportunity to develop new or enhanced functionalities, e.g. for

- Automated driving
- Innovative user interfaces
- Predictive maintenance.

Besides advancements in the functional improvement of AI based systems, the testing of AI based systems needs an extension of existing best practices to reflect the specific characteristics of learning systems, especially for self-learning vehicles.

Why is this important? In AI based systems, algorithmic operations are based on

- The software implementation of the AI algorithm
- The adjustment of internal parameters by learning algorithms based on training cases.
• Fine-tuning of hyperparameters, such as processing layers, processing parameters such as number of units, activation functions, learning rates, stopping criteria.

Figure 1 shows a comparison of classic software development with AI software development. Both need an implementation of software. The implementation of the AI algorithm needs to be programmed or put together from pre-programmed modules. With classic software development, the result of the implementation is directly used after compilation of the code. With AI, the result of a machine learning process is used.

Fig. 1: Comparison of classic software development with AI software development.

From Understanding of AI to Testing of AI

As a consequence of the explanations above, testing an AI driven system means testing

• The code is correct
• What has been learned from the training cases
• (Most important) What has not been learned

Even testing with an independent set of data (meaning a set of test data that was not used in the training process) will give only an incomplete answer, because the data will always be an incomplete representation of the reality.

Machine Learning research relating to this topic is investigating the generalization ability of learning systems, meaning the generalizing across the training patterns and not just memorizing them. Even state-of-the-art research can currently not provide a concise explanation of generalization ability [1] nor a practical measure [2] that would be applicable in the automotive industry. Therefore, the automotive industry cannot just add a "generalization measure" to the test results. In consequence, the evidence for compliance with the system requirements is incomplete.
How would the well-known measure “Test Coverage” be translated into AI-Testing? Just measuring the percentage of test cases from the set of available data will only give an incomplete answer, because it may not be known how well the available data represents the diversity and variability of the real-world application of the system.

**Deep AI Testing**

To bridge the gap in the short term, we propose a Deep AI Test scheme to combine important principles and methods:

- **Independence of Test Data:** Testing based on data that has not been used in the training process to understand what has been learned from the training cases and what not.
- **Frequent Full Testing:** Re-running tests on the full TEST data set to ensure that previously developed and tested software still performs after changes.
- **Selective Extension of TEST Data Base:** Requirements for data collection, that are needed to improve function improvement as well as testing throughout the project.
- **Test-Time Data Augmentation**

These principles and methods are discussed in the following paragraphs.

**Independence of Test Data:**

With systems based on machine learning it is important to assure that the data used to test the system is independent from the data used for training. Why is this?

As everybody knows from school: There are two ways of “learning”:

- **Memorization:** The examples from the classroom can be repeated correctly. This could be done by pure memorization of the training examples.
- **Generalization:** New tasks can be solved correctly. This can NOT be done by just memorization of the training examples. Some underlying principle must be extracted.

Relating this to technical AI based systems, this means:

- The examples from the training phase must be tested: Is the system at least able to perform correctly for TRAINING examples? This is usually done in the training phase.
- In addition, independent TEST examples must be used to test the system, if it performs correctly beyond the TRAINING examples, e. g. something has been learned beyond the memorization of the TRAINING data. These TEST examples must be different from the examples used during training.
Figure 3 shows the process of dividing the available data into TRAINING and TEST data sets.

Fig. 3: Divide Data in Two Independent Sets of Data. In practice, some part of the TRAINING data is often set aside and is used to determine hyperparameters, such as number of processing layers or processing parameters such as number of units, activation functions, learning rates, stopping criteria.

In practice, the following requirements for TEST data apply:

- TEST data shall be large enough to yield statistically meaningful results.
- TEST data shall be representative of the data set as a whole. In other words, don’t pick a TEST data set with different characteristics than the TRAINING data set.
- In practice, the following number should give some guidance:
  - Share of TRAINING data should be >= 50%.
  - For very large datasets, 80%:20% to 90% : 10% should fulfill the requirements.
    - Example: The well-known data set for image recognition CIFAR-10 dataset consists of 60,000 images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images (83% : 17%)
    - For small datasets, the above requirements might translate to something like 60% : 40% to 70% : 30%.

In practice, the number of independent TEST examples will be limited. There will be cases that the development team does not know yet, although they could occur in the real-world application of the system. Therefore, it is strongly recommended to extend the data set throughout the development phase, and even through the operations phase of the system, and split the data to TRAINING and TEST data sets.

It is also required that precautions are taken to assure that the TEST data remains independent during the development phase.
Frequent Full Testing

In the context of AI-based development, the development team is frequently confronted with the situation, that the AI does not learn well enough due to many root causes (e. g. overfitting, underfitting, noise, bias, preprocessing, …). The solution to this is to work in increments and to repeat the following development steps

1. Set up of the AI learning module
2. Confirm correct implementation of the code
3. Training of the module with the TRAINING data
4. Testing of the module after training with TEST data
5. Analysis of the results
6. Definition of improvements, if needed

frequently and do many incremental improvements to the system. Figure 2 illustrates the improvements steps. The incremental approach to AI development assures that the AI based module is tested frequently, and that all changes are tested accordingly after changes.

Due to the nature of Machine Learning, it is usually not possible to define a subset of the TEST data set for a certain change that has been made. In consequence, testing AI means always using as much TEST data as available. The full TEST data set must be tested after changes.

Fig. 2: Improvement steps for AI-based development.

Selective Extension of the Data Base

In the process of data collection, the question is: Which cases are needed to improve the training of the system and which cases will improve the evidence for compliance with the system requirements. Figure 4 illustrates the issue by examples of driving situations in a public road.
Fig. 4: Examples of driving situations on public roads. The left example shows rather seldom road users. However, cows can become possible road users. Should the data collection consider an equal share of cases including cows? This may not only depend on the recognition as one justified type of road user and its location, but also on the ability of the system to create a collision-free and lawful driving plan. The specific speed and “driving plan” of a cow may be significantly different from other road participants. The right picture shows a driving condition that may not lead to new data points which will improve the performance of the system. (Source of left picture: pixabay.com)

How can the extension of the data base be guided? Independent if the development team uses pre-trained AI modules (off the shelf or provided by development partners) or if the training of the AI modules is done by the development team, there are methods to evaluate and select samples. First of all, after the first improvement cycles of AI development (see Figure 2), the performance deficiencies on the available data are known. In case of a classifier, the confusion matrix can be analysed. See Figure 5 for explanation.

One method to guide data sampling is to collect data from classes that are more frequently confused than others and use this data to enhance the performance in the training phase and improve the evidence for compliance with the system requirements.
Fig. 5: Example of a confusion matrix of a classifier distinguishing between 26 classes of patterns. The numbers in green (the diagonal of the matrix) show correct classifications. The numbers in red (False Negatives) and orange (False Positives) indicate errors. The example shows that in real world problems, errors are often not evenly distributed. In this example, the patterns “B” and “V” are confused more often than other pairs, with more False Positives than False Negatives.

Selection of Samples in the Machine Learning Literature

In Machine Learning research there are elaborate methods to select samples that have been developed for the training of AI modules, e. g. the aim of these methods is the improvement of the training phase. However, they can also be applied to the selection of TEST data. Two examples are selected to explain the concepts:

- The Triplet Loss approach minimizes the distance between an anchor and a positive example and maximizes the distance between anchor and a negative example [3].
- Minimal Margin Score Selection (MMS) is an approach to choose the most useful examples for training [4].
Test-Time Data Augmentation

Data augmentation is a method to ease the effect of lack of sufficient amount of TRAINING data or uneven class balance of the TRAINING data set. For example, for classical mage recognition some typical data augmentations are rotating, flipping, cropping or scaling of the images. Data augmentation can be a powerful approach by the usage of additional training patterns that have been generated from the original training patterns by rotating, flipping, cropping or scaling. The learning modules just “sees” more training examples.

Although this is not the original intention of expanding the TRAINING data set, the approach can also be applied in the test phase. How does that work? One original TEST pattern is modified by rotations, flips, crops and scales, and all of them are fed through the classifier. The final vote of the classifier is then generated by a defined policy. For example, one could average the predictions of all modified patterns and take the average as the classification result.

Compared to the training phase, the effect on prediction accuracy in the test phase is not so clear, and by far not so easy to understand. Recent research [5] indicates that it produces a net improvement in accuracy, but it can also change correct classifications into incorrect classifications. The effect of data augmentation in the test phase depends on the nature and amount of training data, the learning model architecture, and the augmentation policy.

Summary of Deep AI Testing Principles

The presented principles for Deep AI testing can be used together. They vary greatly in complexity and benefits. Table 1 provides a summary.

Table 1: Recommendation of test principles and methods. “+++” means it is mandatory, because without it, testing is just measuring how well the TRAINING data was memorized. Like in ISO 26262, “++” indicates that the method is highly recommended and “+” indicates that the method is recommended.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Recommendation for Deep AI Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independence of TEST data from TRAINING data</td>
<td>+++</td>
</tr>
<tr>
<td>Frequent Full Testing</td>
<td>++</td>
</tr>
<tr>
<td>Selective Extension of the Data Base</td>
<td>++</td>
</tr>
<tr>
<td>Test-Time Data Augmentation</td>
<td>+</td>
</tr>
</tbody>
</table>
Meaningful Testing in Context of Self-Learning Vehicles

In Self-Learning Vehicles, the modification of functionality is the result of a learning process. If the learning phase is highly automated, then the modification can also be applied in the operations phase, and not only in the development phase. Machine Learning can take in principle take place in the vehicle (given the needed resources and automation), or in a back end. Figure 6 illustrates the principle set up.

![Diagram](https://example.com/diagram.png)

**Fig. 5:** Illustration of the general set up of machine learning for Self-Learning Vehicles.

Machine Learning can take in principle take place in the vehicle (given the needed resources and automation), or in a back end.

The explanations and recommendations given in this paper for testing hold for both, the development phase as well as the operations phase. If the vehicle is learning in the operations phase, all recommendations concerning testing shall be applied in the operations phase, too. Figure 6 summarizes the principles that shall be used the operations phase with the same rigor as in the development phase.

When it comes to the rigor that should be applied in the obeyance of test principles, then we need to look at the proven method to enforce process maturity in the automotive industry: Capability Assessment models.
Deep AI Testing:
- Independence of Test Data
- Frequent Full Testing
- Selective Extension of Data Base
- Test-Time Data Augmentation

Fig. 6: If the functionality in the vehicle is updated in the operations phase, then the principles Independence of Test Data, Frequent Full Testing, Selective Extension of Data and Test-Time Data Augmentation shall be applied with the same rigour as in the development phase.

Fig. 7: Additional Machine Learning Process Areas within the ASPICE framework. Please note the V-Model is only used as reference to the established ASPICE process areas.
Capability Assessment Model for AI (CAI)
In an earlier paper we introduced the Capability Assessment Model for AI (CAI) [6]. The CAI comprises process areas, goals and best practices from AI experts worldwide. See Figure 7 for an illustration.

The result works surprisingly good for the analysis of development projects applying Machine Learning techniques not only in the automotive industry. Assessments are another application of this process model. We think we will see more assessments including AI-driven development within the next years, because we expect that the technical gaps particularly in the area of safety will be closed.

Conclusions
Application of AI is an important mission for the automotive industry. However, testing of AI is not yet a widespread and well understood discipline. To progress with it, testing of AI based systems needs to reflect the characteristics of learning algorithms. In this paper we propose the Deep AI Test scheme to make AI testing effective and meaningful. The following principles should be applied in AI testing: Independence of test data, frequent full testing, selective extension of the data base and test-time data augmentation. These principles should be spread in the automotive industry and become part of standards, especially for self-learning vehicles.


