

Article

Autoencoder-Based Semantic Novelty Detection: Towards Dependable AI-Based Systems

Andreas Rausch * , Azarmidokht Motamedi Sedeh and Meng Zhang

Institute for Software and Systems Engineering, Technische Universität Clausthal, 38678 Clausthal-Zellerfeld, Germany; azarmidokht.motamedi.sedeh@tu-clausthal.de (A.M.S.); meng.zhang@tu-clausthal.de (M.Z.)

* Correspondence: andreas.rausch@tu-clausthal.de; Tel.: +49-5323-72-7177

Abstract: Many autonomous systems, such as driverless taxis, perform safety-critical functions. Autonomous systems employ artificial intelligence (AI) techniques, specifically for environmental perception. Engineers cannot completely test or formally verify AI-based autonomous systems. The accuracy of AI-based systems depends on the quality of training data. Thus, novelty detection, that is, identifying data that differ in some respect from the data used for training, becomes a safety measure for system development and operation. In this study, we propose a new architecture for autoencoder-based semantic novelty detection with two innovations: architectural guidelines for a semantic autoencoder topology and a semantic error calculation as novelty criteria. We demonstrate that such a semantic novelty detection outperforms autoencoder-based novelty detection approaches known from the literature by minimizing false negatives.

Keywords: safety engineering; autonomous system; perception; artificial intelligence; autoencoder; novelty detection



Citation: Rausch, A.; Sedeh, A.M.; Zhang, M. Autoencoder-Based Semantic Novelty Detection: Towards Dependable AI-Based Systems. *Appl. Sci.* **2021**, *11*, 9881. <https://doi.org/10.3390/app11219881>

Academic Editors: Rui Araújo and João M. F. Rodrigues

Received: 20 September 2021

Accepted: 20 October 2021

Published: 22 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

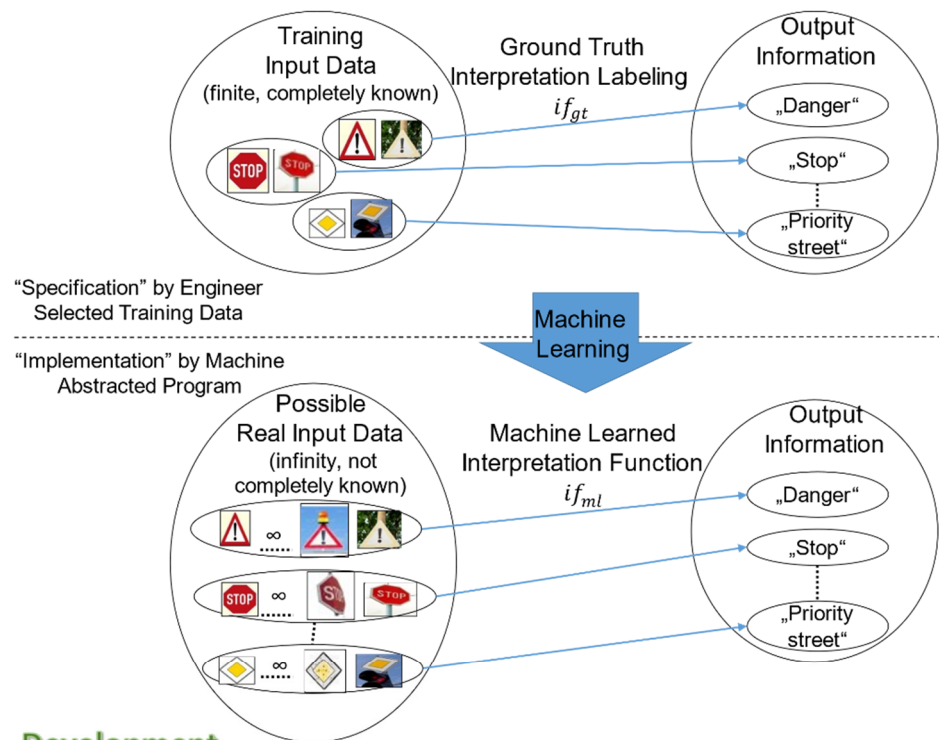
1. Introduction

Recently, autonomous systems have achieved success in many application domains, including autonomous vehicles (AVs), smart home systems, and autonomous financial agents. Autonomous systems are becoming more useful and beneficial for us. As a side effect, we, the users, increasingly rely on their services, even in safety-critical applications such as driverless taxis [1,2].

Many recent advancements in the performance of autonomous systems have been made possible by the application of machine learning (ML) techniques [3]. Nowadays, autonomous systems are hybrid AI-based systems, integrating classical engineered subsystems combined with subsystems using artificial intelligence (AI) techniques. For instance, vehicle controllers are classical engineered subsystems, whereas the perception subsystems of AVs are nowadays mainly AI based; both are integrated in an AV and together perform safety-critical functions.

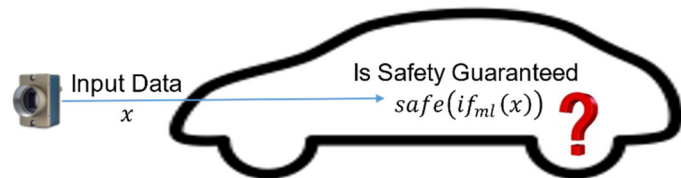
For the development of perception subsystems, engineers use labeled training data and ML technologies to train an interpretation function. For illustration, a simplified perception task the classification of traffic sign images to the corresponding traffic sign classes, i.e., a semantic interpretation of images, as shown in Figure 1. Training data map a finite set of input data (images of traffic signs) to correct output information (traffic sign classes). A machine-learned interpretation function is trained using this training data. As ML abstracts from given training data examples and produces a machine-learned function that can process an infinite number of different images, the resulting machine-learned interpretation function maps any type of image taken by the camera in a real environment to one of the traffic sign classes. During system operation from a safety perspective, an essential question is whether the produced output information, $if_{ml}(x)$, of the machine-

learned interpretation function, if_{ml} , processing the current input data, x , is sufficiently reliable to be safe.



Development

Operation



$$safe(if_{ml}(x)) \Leftrightarrow semantic_similar(x, y) \geq threshold_{semantic_similarity}, \text{ with } (y \in TrainingInputData) \wedge (if_{gt}(y) = if_{ml}(x))$$



Figure 1. Operation time check of dependability requirements.

There are differences between the engineering of classical systems and AI-based systems [4]. The classical engineering process for safety-critical applications begins with a (semi-) formal requirements specification, which must be complete and correct. While this idealized process is rarely realized to its full extent, the requirements specification is later used as the main input for testing and verification. For the development of an AI-based system, instead of requirements specification, a large data collection is used. The collected data may have missing information or contain a small percentage of incorrect data samples.

Nevertheless, these AI-based systems have been widely applied to fulfill safety-critical tasks. Product liability regulations impose high standards on manufacturers regarding the safe operation of such systems [5]. Established engineering methods are no longer adequate to guarantee the dependability requirements (safety, security, and privacy) in a cost-efficient way due to many limitations, for instance, they are not able to handle the specific aspects of AI-based systems [1]. Therefore, engineers cannot completely test and verify autonomous systems during development to fully guarantee their dependabil-

ity requirements in advance. Thus, Aniculaesei et al. [6] introduced the dependability cage concept. Dependability cages are derived from existing development artifacts and are used to test the system's dependability requirements both during development and during operation.

One essential part of the dependability cage concept (cf. Figure 3) is the quantitative monitor. In the scenario of Figure 1, the safety task of the quantitative monitor intuitively indicates whether the system is currently processing actual real input data x , which are from a reliability perspective similar enough ($semantic_similar(x, y) \geq threshold_{semantic_similarity}$) to the (ground truth) training input data (y) used for ML, so that the produced actual output information of the machine-learned interpretation function $if_{ml}(x)$ can be assumed to be correct and safe. If this is not the case, the AI-based system is possibly in an unsafe state.

Providing a measure for the semantic similarity of input data that serve as an argument for the output information reliability is a challenge ($semantic_similar(x, y)$). Novelty detection to automatically identify new relevant test data differing from the available training data becomes an interesting approach to realize such a semantic similarity measure. Very promising novelty detection approaches use AI-technologies, such as an autoencoder or generative adversarial network approaches [7].

The basic principle for autoencoder-based novelty detection was introduced by Japkowicz et al. [8]. An autoencoder was trained to minimize the error between an input image and a reconstructed input image. Known images were used to train the autoencoder. After training, the autoencoder was fed with new images. If the difference between the original image and the reconstructed image was higher than a given threshold value, the new image was suspected to be novel. Richter et al. [9] trained an autoencoder with known images of handwritten digits. Their autoencoder-based novelty detection approach could identify images of letters with nonstandard fonts as novel. Amini et al. [10] demonstrated the same with images from known daytime versus novel nighttime driving situations. These "naive" autoencoder-based novelty detection approaches can meaningfully generalize a family of input images with a common structure and classify input images with different structures as novel data.

However, a significant structural difference in input data does not necessarily correlate with a different output information class. To guarantee the dependability of an AI-based system, we must ensure that there are sufficient comprehensive training input data for each relevant output information, regardless of how large the structural difference in input data is. For instance, as illustrated in Figure 2, completely different traffic situations (output information) frequently have similar images (input data). On the right-hand side, the AV has to stop and let the pedestrians pass the zebra crossing. On the left-hand side, the AV can pass the zebra crossing.

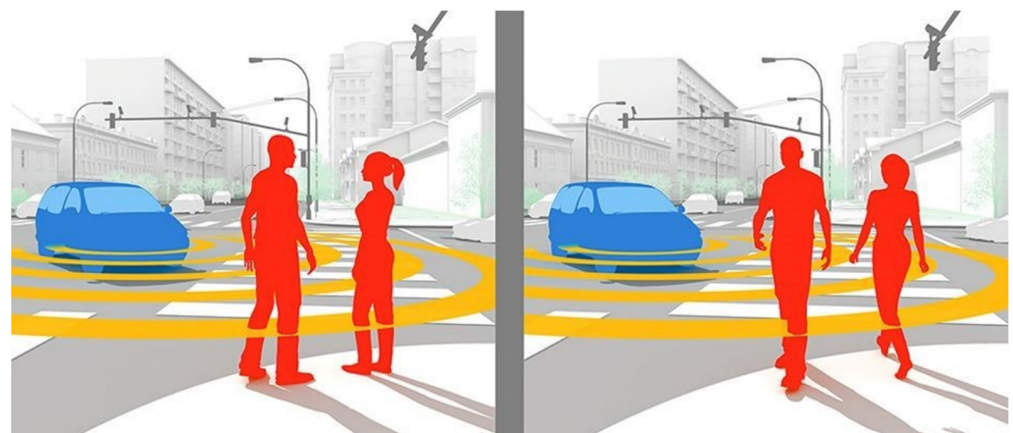


Figure 2. Similar images of pedestrians represent different situations [11].

AI-based systems are generally known to lack robustness, for example, in the case that the training data, rarely or completely, did not cover critical situations. The challenging task is, if a new image represents a data example for an unknown driving situation (relevant new output information), it must be identified as novel, regardless of how similar it is to an existing one.

To mitigate this problem, we propose a new architecture for autoencoder-based semantic novelty detection with two innovations: (a) architectural guidelines for a semantic autoencoder topology that provokes the autoencoder to reconstruct novel input data with a distinct sufficient reconstruction error and (b) an architectural blueprint for a semantic error calculation as novelty criteria that calculates the difference between the original and the reconstructed input based on the resulting error of the corresponding output information. We demonstrate that such a sophisticated autoencoder-based semantic novelty detection approach is much more powerful than existing “naive” autoencoder-based novelty detection approaches, such as the approach by Richter et al. [9].

To compare our results with the “naive” autoencoder-based approach, we apply the same setting as Richter et al. using the Modified National Institute of Standards and Technology database (MNIST-dataset) [12] for our experiments in this study. Since the presented work is part of an ongoing study, our results will be later validated in real autonomous driving use cases. More extensive experiments will be published in subsequent publications. However, the results we present in this paper based on the MNIST-dataset [12] are promising. Once they are validated in real traffic situations, such a semantic novelty can be used to (1) continuously improve the quality of training data for development, (2) to provide a safety argument for the approval of an autonomous system, and (3) to establish an on-board diagnosis system for system operation. In addition, it can be used to identify corner cases and facilitate the de-biasing of training data.

The remainder of this paper is structured as follows: In Section 2, we introduce a quick overview of the dependability cage monitoring architecture for autonomous systems; in Section 3, we provide an overview of related work, particularly AI-based novelty detection; in Section 4, the main research hypotheses are introduced; to evaluate the proposed concepts and research hypotheses, we describe, in Section 5, the evaluation framework, which we have used; in Sections 6–8, the architecture of the autoencoder-based semantic novelty detection is incrementally developed and evaluated concerning our three research hypotheses; in Section 9, we provide a brief summary of conclusions and future work.

2. Dependability Cage Approach: A Bird’s Eye View

To tackle the challenges of engineering-dependent autonomous systems, the dependability cage concept has been proposed [6,13–16]. Dependability cages are derived by engineers from existing development artifacts. The core concept of the approach is a continuous monitoring framework, as shown in Figure 3.

For developing AVs, a common high-level functional architecture has been established, which includes three parts: (1) environmental and self-perception, (2) situation comprehension and action decision, and (3) trajectory planning and vehicle control [17–19].

The monitor framework focuses on two issues: (a) shows the system correct behavior in terms of dependability requirements (*qualitative monitor*) and (b) operates the system in a situation and environment that has been trained or tested during development (*quantitative monitor*) (cf. Figure 3).

The monitors both require consistent and abstract data access to the system under consideration, with the help of the *input* and *output abstraction* components. They depict the interface between the autonomous system and the two monitors. The input and output abstraction components both use defined interfaces to access the autonomous system’s data and transform it into abstract representations. Abstract representation types and values are derived from the requirements specification and dependability criteria of the autonomous system.

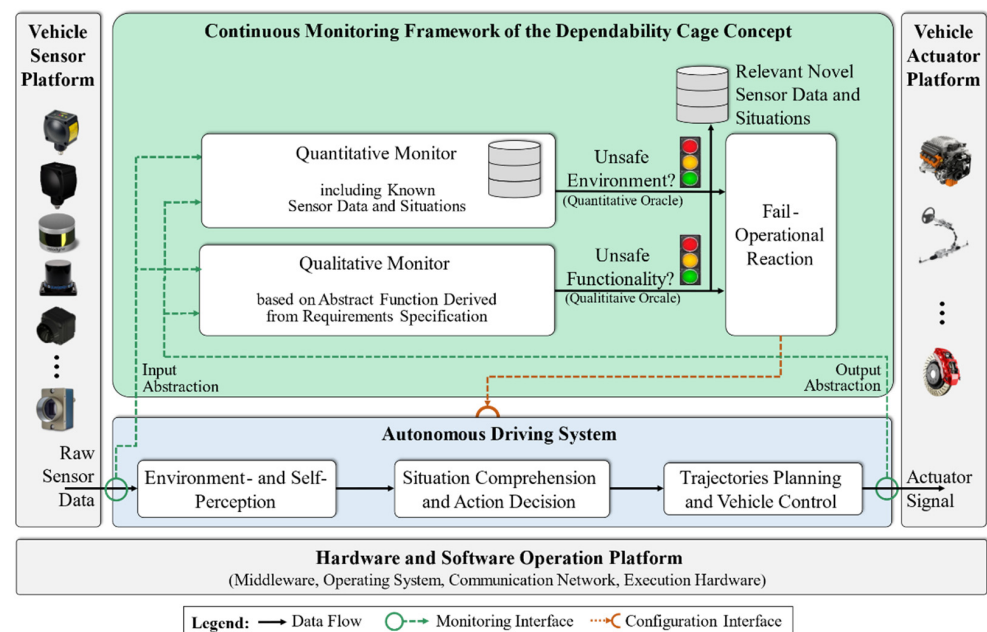


Figure 3. Continuous monitoring architecture as a part of dependability cage.

The qualitative monitor evaluates the correctness and safety of the system's behavior under the assumption that the system operates in a situation and environment that conform to the requirements specification, and it consists of an abstract behavior function and a conformity oracle. The abstract behavior function calculates a set of correct and safe actions in real time for the system in the current abstract situation. The conformity oracle compares the output abstraction with the set of correct and safe abstract actions from the abstract behavior function.

The quantitative monitor observes the encountered abstract situations. For each situation, the monitor evaluates in real time if the encountered abstract situation is already known from development. A knowledge base provides information about tested situations on an abstract level. A canonical representation of abstract situations is used in this study. These canonic abstract situations are considered to be unique situations.

If one of the monitors detects incorrect and unsafe system behavior, safety measures must be initiated to guarantee dependability requirements. The *fail-operational reaction* component must transfer the corrupted system into a safe state with acceptable risk. For instance, one fail-operational reaction can be a graceful degradation of the autonomous system, as described in [20].

Moreover, in the case that safety measures are initiated, system data will be automatically recorded. The recorded data can be transferred back into system development to identify and to resolve previously unknown faulty system behavior as well as to extend the scope of system tests. The qualitative monitor has already been successfully applied and evaluated using an industrial prototype of a highway pilot and data collected from road tests on German highways [13,15].

To realize a quantitative monitor, we need an efficient and effective solution to identify unknown sensor input data. As previously mentioned, we claim and will show in the following sections that our proposed semantic novelty detection approach can be used to successfully realize a quantitative monitor for a perception system.

3. State of the Art in Novelty Detection

Novelty detection as a classification task to determine whether newly observed data differ from the available training and test data has been widely used in different fields. Different approaches of novelty detection have been investigated [7,21,22]. Novelty detection is closely related to anomaly detection, which typically refers to the detection of

samples during inference that do not conform to an expected normal behavior [23]. If enough diverse training data is available any anomaly is a novelty, but not any novelty is necessarily an anomaly. Hence, novelty detection is the more general and safer approach for us. Note that automotive novelty detection is also known as corner case detection.

One well-known classification of novelty detection approaches was proposed in [7], consisting of five taxonomies, i.e., probabilistic, distance-based, domain-based, reconstruction-based, and information theoretic approaches. In our work, we focus on the taxonomy of reconstruction-based approaches, which are often used on safety-critical applications and cover approaches based on diverse neural networks (NNs) such as MLP, SVM, and auto-associator, as well as oscillatory and habituation-based networks [24].

Auto-associator approaches, particularly referring to autoencoder approaches, have gained our attention in the research because of their strong ability to implicitly learn the underlying characteristics of an input dataset without any prior knowledge or assumptions [24]. Autoencoder approaches automatically learn data abstraction on different implicit representation levels by attempting to regenerate the system output as the same as the originally given input, such as camera images, which have been investigated in different studies [25–27].

For example, Byungho et al. identified three critical properties of autoencoder-based novelty detection: (a) production of the same output vector with infinite input vectors, (b) output-constrained hyperplane when using bounded activation functions, and (c) hyperplane located in the vicinity of training pattern by the minimizing error function [28]. Diaz et al. found that kernel regression mapping is better suited than least squares mapping for autoencoder networks of novelty detection [25]. Japlowicz et al. [8] focused on the idea of setting the threshold as some function of the largest reconstruction error for any single training data point, and based on this, Richter et al. [9] selected a high percentile of the cumulative distribution of losses on training datasets as their threshold to make it more robust in detecting novel instances with different input data structures. Their experimental setting—handwritten digits versus (novel) letter images—serves as starting point for our work. To improve the criterion of novelty detection, Manevitz et al. investigated autoencoders on different levels of error and determined an optimal real multiple of the standard deviation of the average reconstruction error as a better threshold. Their evaluation was based on an application for document classification [29].

As already introduced in the previous sections, the aim of this study is to realize a quantitative monitor as part of the dependability cage by using a new improved autoencoder-based approach. The autoencoder-based approach by Richter et al. [9] is used as a reference in our study, for which details are introduced in Sections 5–8. By introducing a new semantic novelty detection approach using a new autoencoder architecture, we will improve the results of Richter et al.

4. Research Hypothesis of This Work

To realize the previously mentioned task of quantitative monitoring, we propose an autoencoder-based approach of semantic novelty detection during real-time system operation to classify a current situation into (a) tested and known or (b) novel and unknown. In the case of (b), the corresponding raw data will be recorded for further development of the monitored autonomous system. Moreover, the classification may also lead to a (a) safe or (b) unsafe state decision in the system and a corresponding fail-operational reaction. It can be concluded that from the viewpoint of safety it is important to detect all novel situations. Otherwise, the system decides to operate in an unlearned environment, which could lead to a safety-critical situation.

In Section 3, an autoencoder-based novelty detection approach by Richter et al. [9] was briefly introduced, in which, as compared to the images of the MNIST handwritten digits dataset [12], the images of letters with nonstandard fonts from the notMNIST-dataset [30] were successfully classified as novel. To compare our results with their approach, we reproduced the results of Richter et al., as described in Section 5. We focused particularly

on their example of detecting letter images as novel. Our reproduced results showed that Richter's approach works well in identifying novel data that have different input data structures as compared with the MNIST-dataset, which is used to train the autoencoder.

In this paper, we argue that their success depends on major structural differences in the input data, i.e., images of digits and letters. Therefore, we modified the experiment of Richter et al. such that we randomly excluded some classes of digits from the training set of the autoencoder to determine whether the autoencoder could still detect these unknown digits as novel. Therefore, in Section 5, we present a slightly modified dataset, containing seven classes of digits for training, and the other three classes of digits as well as nine classes of letters remain as novel data.

We expect that Richter's approach will fail once the novel data have a minor structural, but a major semantic difference. Consequently, images will be detected as known instead of novel, producing a high rate of false negative errors. In the subsequent section, Richter's approach is taken as a reference and referred to as the "naive" autoencoder-based novelty detection, for which failure regarding false negative errors is further proven in Section 6. Additionally, to compare the autoencoder-based approach, Richter's, as well as our improved approach, with other AI-based approaches for novelty detection, a normal classification neural network for novelty detection is introduced in Section 5, which is named as a simple classification approach in this study and serves as a lower bound benchmark solution in our evaluation.

To minimize the false negative errors, we improve autoencoder-based novelty detection in two steps. The first step is based on a semantic topology, which is called semi-semantic novelty detection. In this approach, the semantics of the monitored system's output, which is a classification system, is considered. We force the autoencoder to reflect the semantics in its topology by applying three rules (cf. Section 7). In the second step, to realize a so-called fully semantic novelty detection, we reuse the topology of semi-semantic novelty detection, but the novelty detection criterion is replaced by a semantic loss instead of the original mean squared error (MSE). With this, our research hypothesis (HYPO) for upcoming sections are as follows:

Hypothesis 1 (H1). *"Naive" autoencoder-based novelty detection does not work well for similar structural input data with a different semantic interpretation.*

Hypothesis 2 (H2). *Autoencoder-based semi-semantic novelty detection works better than the "naive" approach.*

Hypothesis 3 (H3). *Autoencoder-based fully semantic novelty detection can find novel data completely.*

The first hypothesis is evaluated in Section 6 by using the simple classification network mentioned above as a lower bound benchmark solution. The last two hypotheses are evaluated, respectively, within benchmarks with the "naive" approach to evaluate the performance of both improved approaches for minimizing false negative errors in the proposed novelty detection approach (cf. Sections 7 and 8). Our results show that both approaches have significantly fewer false negatives than the "naive" approach, and the fully semantic approach can find all novel data in our experimental setup.

The following section introduces our concrete evaluation framework, including repetition of the original experiment of Richter et al. [9], a slight change to the applied dataset of Richter et al. with respect to our HYPO1, and the establishment of the mentioned lower bound benchmark solution based on the simple classification neural network with the usage of the changed dataset.

5. Evaluation Framework for This Work

To evaluate our proposed semantic novelty detection approach, we developed an evaluation framework based on Keras [31] in the TensorFlow [32] framework. The mentioned novelty detection approaches ("naive", semi-semantic, and fully semantic) were imple-

mented as ROS2 [33] nodes and serve as part of the quantitative monitor (cf. Figure 3). For a comparable and comprehensive evaluation, firstly, we reproduced the results of Richter et al. [9] by training a feedforward autoencoder with configurations in Table 1.

Table 1. Training setup of the autoencoder in the “naive” novelty detection.

Setup of Training Autoencoder in the “Naive” Novelty Detection Approach		
Autoencoder Topology	784-50-50-50-784	
Hyperparameters	Batch size	1024
	Epochs	3000
	Learning rate	0.01
	Activation function	sigmoid
Training set	Randomly selected 30,000 digit images from MNIST-dataset	
Test set	Randomly selected 12,500 digit images from MNIST-dataset	

The autoencoder aims to minimize the mean squared pixel-wise deviations between raw and reconstructed images during training. The percentile of the empirical cumulative distribution of reconstruction errors for the training set (cf. Table 1) was decided as the threshold to identify known and unknown data. In the case of an image, if the reconstruction error falls below this threshold, the input image is classified as known, otherwise as unknown (novel).

One hundred randomly selected digit images were taken as the dataset of known classes (known digits in Figure 4), and an additional one hundred letter images from the notMNIST dataset (letters in Figure 4) were taken as novel data. Our result of novelty detection (cf. Figure 4) is similar to the work of Richter et al. [9] and shows that the autoencoder detects all digit images as known (black graph line) and all letter images as unknown (red graph line). Thus, we could reproduce the results of [9].

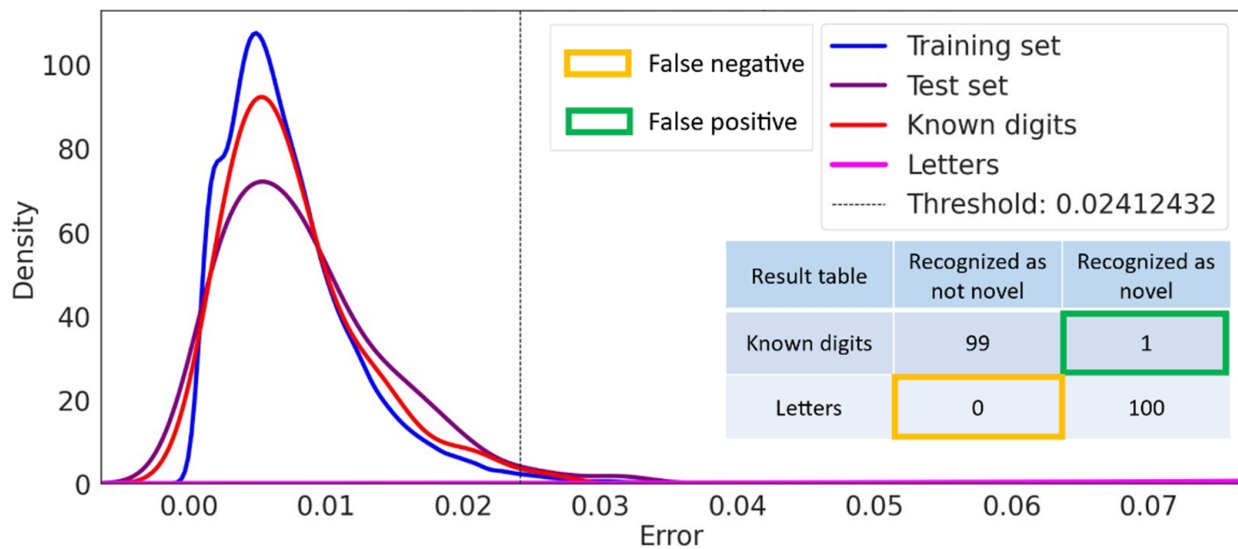


Figure 4. Reproduced results of the “naive” novelty detection approach of [9].

To prove our HYPO 1–3 above regarding the work of Richter et al. in [9], namely that the “naive” autoencoder approach fails once the novel input data have a minor structural but a major semantic difference and that our new semantic autoencoder approach is even still convincing in those cases, we modified the experimental setting of the dataset. We randomly excluded three classes of digits (“2”, “3”, and “5”) from the training and test set. The other seven classes of digits remained in the known training data. Thus, novel input

data consist, now, out of three classes of digits and nine classes of letters. The following experiments have all used the same dataset shown in Table 2.

Table 2. Used dataset for the evaluation benchmarks.

Training and Test of Autoencoder	
Training set	30,000 randomly selected images of known digits ($KN \in \{0, 1, 4, 6, 7, 8, 9\}$)
Test set	12,500 randomly selected images of known digits ($KN \in \{0, 1, 4, 6, 7, 8, 9\}$)
Performance Evaluation of Novelty Detection	
Known digits	100 × randomly selected images of known digits ($KN \in \{0, 1, 4, 6, 7, 8, 9\}$)
Unknown digits	100 × randomly selected images of unknown digits ($UN \in \{2, 3, 5\}$)
Letters	100 × randomly selected images of (unknown) letters ($L \in \{A, B, C, D, F, G, H, I, J\}$)

Based on the resulting probability, a simple novelty detection can be performed. We used the 99th percentile as the threshold, which means that if the probability for one of the classes is higher than 0.99, the input image is categorized into this class. Otherwise, if none of the probabilities is higher than the threshold, the input image cannot be classified into any known class, and thus is identified as novel.

The simple classification network already mentioned in Section 4 was built sequentially with three convolutional and corresponding three max pooling layers, as described in Table 3. Afterward, this convolutional network was followed by two fully connected ReLU layers as well as a ReLU output layer with 7 nodes according to the number of known classes. We used the Adam optimizer [34] and attempted to minimize the categorical cross-entropy loss during training. To maintain evaluation authenticity, we used the same number of images for training and testing results (cf. Table 2).

Table 3. Topology of simple classification network.

Topology of Simple Classification Network				
Layer type	Kernel Size	Number of Filters	Padding	Activation
1st Convolutional layer	3×3	28	Same	ReLU
Max pooling	2×2	Dropout: 0.2, with Batch Normalization		
2nd Convolutional layer	3×3	64	Same	ReLU
Max pooling	2×2	Dropout: 0.2, with Batch Normalization		
3rd Convolutional layer	3×3	64	Same	ReLU
Max pooling	2×2	Dropout: 0.2, with Batch Normalization		
		Number of Neurons		Activation
Dense layer		256		ReLU
Dense layer		128		ReLU
Output layer		7		ReLU

The performance of the novelty detection based on the simple classification network is shown in Figure 5. As introduced before, it will serve as a lower bound benchmark for the following experiments. It is noted that for this simple classification approach, we could not show the density graph of the reconstruction error because the classification approach is not based on the principle of image reconstruction. As shown in Figure 5, almost all known digits were classified as not novel. Most of the unknown digits were classified as novel, but most of the letters were identified as not novel.

Result table	Recognized as not novel	Recognized as novel
Known digits	97	3
Unknown digits	39	61
Letters	37	63

Figure 5. Performance of simple classification network for novelty detection.

In the following three sections, we use this evaluation framework and the lower bound benchmark solution based on the simple classification network to prove HYPO 1–3 and show the performance of the autoencoder approaches within following experiments: (1) “naive” novelty detection vs. classification network (cf. Section 6), (2) semi-semantic novelty detection vs. “naive” novelty detection (cf. Section 7), and (3) semi-semantic novelty detection vs. fully semantic novelty detection (cf. Section 8).

6. “Naive” Autoencoder-Based Novelty Detection

6.1. Architecture

To evaluate the performance of the “naive” autoencoder-based novelty detection approach, we used a quantitative monitor (cf. Figure 3), whereas the input abstraction of the quantitative monitor is realized by the autoencoder component. The quantitative monitor itself in Figure 3 is represented by a discriminator component, as shown in Figure 6.

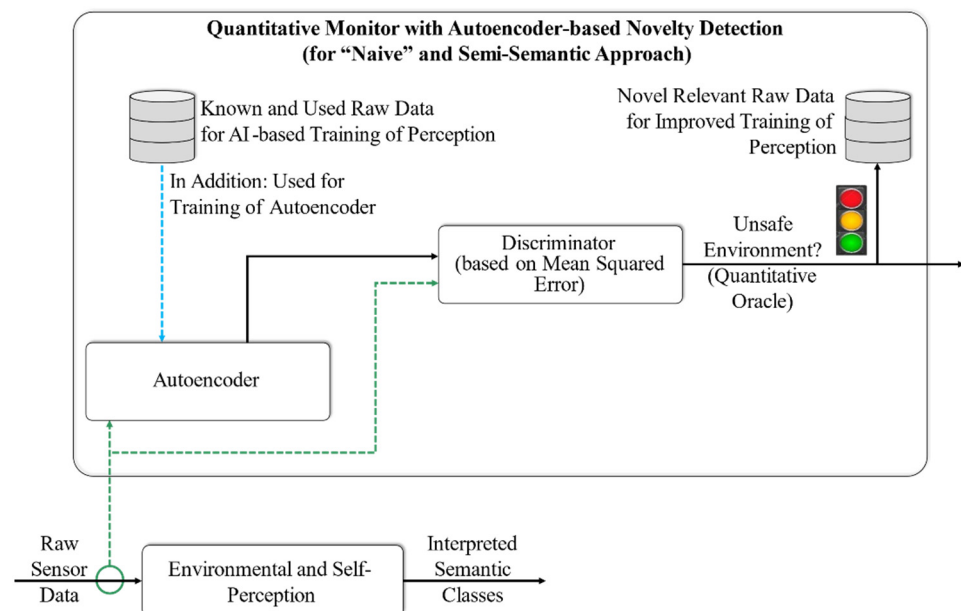


Figure 6. Quantitative monitor with autoencoder-based novelty detection.

The autoencoder component here refers to the “naive” one (following the Richter’s “naive” approach [9]). It is trained with the same data used for the training of the normal digit classification network, which serves as a toy sample for the perception part of an autonomous system. Thus, the autoencoder has abstract knowledge of known data from training and testing.

The configuration of the “naive” autoencoder is described in Table 1, which is generated from our reproduced results of [9]. The discriminator component compares the output

of the autoencoder, which is the reconstructed input by the autoencoder, with the inputs of raw sensor data to compute a reconstruction error in terms of MSE (mean squared error).

The autoencoder is trained to successfully reconstruct known (from training and test) input data, which was in [9] assumed to have a significantly lower MSE than novel input data. On the basis of the computed threshold mentioned in Section 5 (99th percentile of the empirical cumulative distribution of reconstruction errors), the discriminator determines whether the input sensor data are known or novel. In the case of novelty, novel relevant data will be recorded to (re-)train, and thus improve the perception part of an autonomous system. After training the perception part, the new relevant data will be taken as known raw data to (re-)train the autoencoder.

6.2. Evaluation

Since Richter et al. [9] did not consider the unknown digits (cf. Figure 4), the performance of their approach cannot be compared with that of the simple classification approach (cf. Figure 5). Therefore, we had to use our refined evaluation framework with Richter's "naive" autoencoder-based novelty detection.

The results of our evaluation, shown in Figure 7, confirmed HYPO 1. Using our evaluation framework with the "naive" autoencoder (cf. Table 2) illustrates that the "naive" autoencoder-based approach cannot identify the structural similarity but semantic difference between the unknown digits as novel (green graph line in Figure 7). The results in Figure 7 show that 84% of the unknown digits are not identified as novel. The remaining results were comparable to the original evaluation by Richter et al. and were also comparable to our reconstruction of the evaluation, as shown in Figure 4.

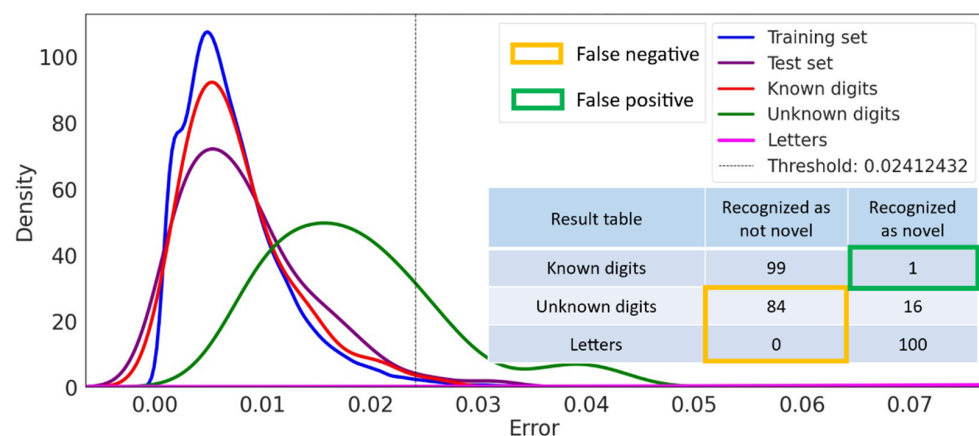


Figure 7. Performance of "naive" autoencoder-based novelty detection.

Overall, the "naive" autoencoder-based novelty detection only performs well when there is a strong structural difference in input images (known digits vs. letters). When there are minor structural differences in input images (known digits vs. unknown digits), the "naive" autoencoder-based novelty detection showed a poor performance.

In contrast, the performance of the simple classification approach for novelty detection performed well when there was a minor structural difference (unknown digits) and poorly in strong structural difference (letters) (cf. Figure 5). Hence, a promising novelty detection approach is required to discover novel data when there are structural and semantic differences in input data. This helps the quantitative monitor reduce the number of false negatives (unknown classes of digits resp. letters that are not recognized as novel).

7. Autoencoder-Based Semi-Semantic Novelty Detection

7.1. Architecture

To minimize these false negative errors, we proposed our autoencoder-based semi-semantic approach and used it in the quantitative monitor. To enable a reliable novelty

detection under the consideration of correlation between the degree of structural difference in the input data and different semantic interpretations, the topology of the autoencoder is derived from the parameters of the semantics of the monitored system’s output in this approach. Since only the topology of the autoencoder (component autoencoder) was changed to a so-called **semantic topology**, the architecture stays the same (cf. Figure 6).

In our work, the semantics of the monitored system’s output is a classification system comprised of different classes. To reflect the knowledge about known classes, we force the autoencoder to code the known classes in the middle layer, the latent space. Thus, the number of neurons in the latent space is between the number of classes and the number of bits one would need for a binary coding of the known classes. Therefore, the following rule determines the range for the number of neurons in the middle layer (the latent space, cf. Figure 8), as the **1st rule** for a semantic topology-based autoencoder as follows:

$$\lceil \log_2 n \rceil \leq f \leq n, \tag{1}$$

f : Number of neurons in the middle layer.

n : Number of known classes.

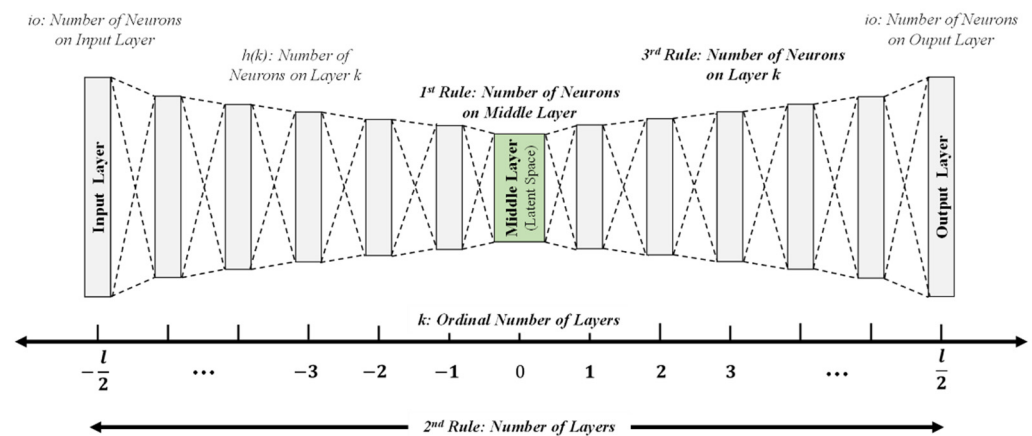


Figure 8. Rules for semantic topology of the autoencoder.

By applying this rule, we enforce the autoencoder to find an abstraction of the known classes in its topology. Therefore, when an input image does not belong to one of the output classes (thus a novel image) the autoencoder reconstructs this input image with a distinct reconstruction error.

In addition to the number of neurons in the middle layer, we postulate two other rules. These two additional rules ensure that the topology of the encoder part of the autoencoder has an exponentially decreasing number of neurons per layer and the decoder part of the autoencoder has an exponentially increasing number of neurons per layer. The intention of these two rules is to force the autoencoder to focus on the abstraction of the known classes in the latent space. Hence, the 1st rule forces the latent space to realize the coding of the known classes, the 2nd and 3rd rules aim to force the autoencoder to quickly abstract the input data to the known classes in the latent space.

The natural number $\lceil \log_2 n \rceil$ is the base number of layers for the encoder as well as for the decoder part ($2 * \lceil \log_2 n \rceil$). As the optimal number of layers is around this base number and has to be an uneven number, we add a ∓ 1 . Finally, as we also consider input and output layers for these rules, we add $+2$ at each side of inequality, for the **2nd rule** for a semantic topology-based autoencoder as follows:

$$((2 * \lceil \log_2 n \rceil) - 1) + 2 \leq l \leq ((2 * \lceil \log_2 n \rceil) + 1) + 2, \tag{2}$$

l : Number of layers, including input and output layer. Note, only an uneven number of layers is a valid choice.

For the last rule, we need an ordinal number for each layer. We defined the middle layer (latent space) to be the layer with the ordinal number 0. The ordinal number k of a layer is the distance to the middle layer. This means that the next layer on the left-hand side of the middle layer has an ordinal number -1 , the next layer on the right-hand side has an ordinal number 1, and so on. We considered the input and output layer to be the layers with the lowest (input layer) and the largest (output layer) ordinal numbers, for the 3rd rule for a semantic topology-based autoencoder as follows:

$$h(k) \pm \begin{cases} io, \text{ with } k = \pm \lfloor \frac{l}{2} \rfloor \text{ (input or output layer)} \\ h(k) \text{ derived from 1st rule, with } k = 0 \text{ (middle layer)} \\ 2^{\lfloor \log_2 io \rfloor - \sum_{i=0}^{\lfloor \frac{l}{2} \rfloor - |k| - 1} i} \geq h(k) \geq 2^{\lfloor \log_2 io \rfloor - \sum_{i=0}^{\lfloor \frac{l}{2} \rfloor - |k| - 1} 2^i} \\ , \text{ otherwise} \end{cases} \quad (3)$$

- k : Ordinal number of a layer.
- $h(k)$: Resulting number of neurons in layer k .
- io : Number of neurons in the input or output layer.

By applying these two additional rules, we ensured that the autoencoder shows an exponential packaging of the relevant features for the semantics of the monitored system’s output. We ensured that the autoencoder quickly focused on the relevant semantic features. The proposed quantitative monitor with the autoencoder-based semi-semantic novelty detection is, then, compared to the “naive” autoencoder-based approach, which is discussed in greater detail in the following section.

7.2. Evaluation

As previously mentioned, three digits in the MNIST-dataset were omitted in the evaluation framework. Thus, the number of classes to be identified had been reduced to seven. Following the 1st rule, the number of neurons in the latent space should stay between three and seven. As a first attempt, we only used seven neurons in the latent space in the original autoencoder. The resulting topologies of the autoencoder were 784-50-7-50-784 neurons, hence, the three hidden sigmoid layers had 50-7-50 neurons and the input and output layer had 784 neurons, due to the image size. The rest of the evaluation setting remained constant.

Figure 9 demonstrates the evaluation result of the first rule of the semi-semantic autoencoder, which has a better performance than the “naive” autoencoder (cf. Figure 7) but does not perform better than the simple classification approach (cf. Figure 5) regarding the identification of unknown digits.

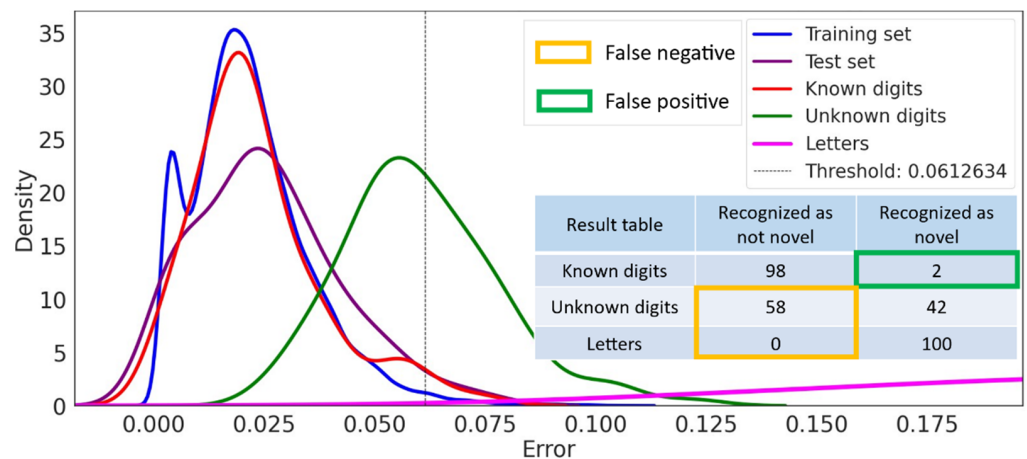


Figure 9. First improvement of novelty detection with a semantic topology.

The 2nd and 3rd rules provide a design space of a semi-semantic topology-based autoencoder for novelty detection. Experimentally, we discovered the best topology for the given novelty detection task within the design space: an autoencoder with five hidden layers, respectively, with the neuron numbers of 512, 128, 7, 128, and 512 for these five layers, for which the evaluation result is better than the “naive” autoencoder-based novelty detection as well as the simple classification network, as shown in Figure 10. Overall, by using our three rules, we derived an autoencoder-based semi-semantic novelty detection, which performs better than the “naive” autoencoder approach, and thus confirms HYPO 2 (cf. Section 4).

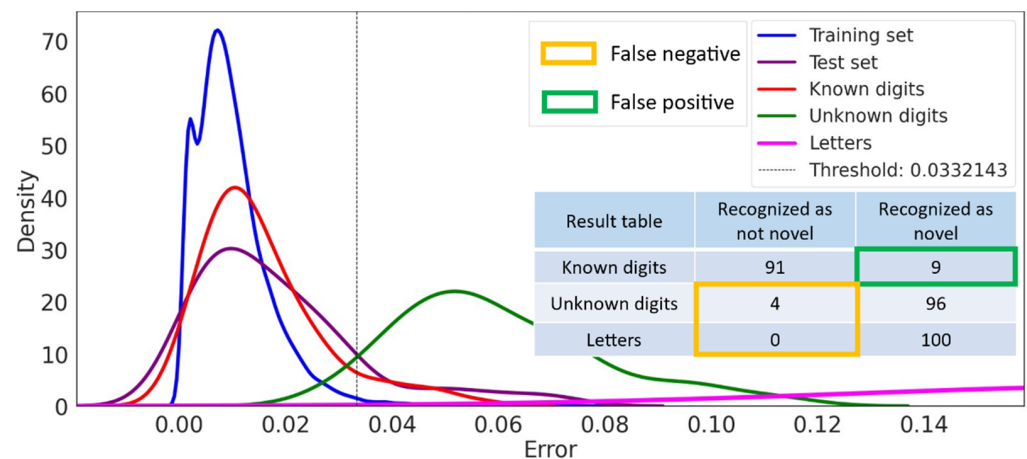


Figure 10. Applying the rules: novelty detection with a semantic topology.

8. Autoencoder-Based Fully Semantic Novelty Detection

8.1. Architecture

According to the evaluation results in Section 7, the semi-semantic approach with semantic topology of the autoencoder has shown great performance for minimizing the false negative errors in novelty detection. During the evaluation, we found that the current classification criterion of the threshold based on MSE could not segment known and novel input data, which were similarly structured but had different semantic meanings, for example, in the case of known digits versus unknown digits (cf. Figure 10). This could also be another reason for the false negative errors in addition to the topology of the autoencoder.

Since the current classification criterion for novelty detection is the threshold solely based on the reconstruction MSE of raw input data, it means that in the case of camera image data, the threshold is determined based on pixel-wise deviations between the raw and reconstructed image without any semantic meaning. In this case, the false negative errors due to similarly structured but semantically different novel data are obviously remaining.

With this hindsight, we propose our next improvement: autoencoder-based fully semantic novelty detection based on a semantic error instead of the threshold based on MSE. The architecture of the quantitative monitor using this approach is shown in Figure 11.

As compared with the semi-semantic novelty detection, the component autoencoder with the semantic topology (cf. Section 7) remains the same. The output of the autoencoder (reconstructed raw sensor data) is forwarded to the *environmental and self-perception* components with the identical interpretation mechanism acting as the perception part of the monitored autonomous system, which delivers its own interpreted semantic classes as output (cf. Figure 11), such as the object list in the case of the autonomous driving system on the vehicle. These interpreted semantic classes are used as inputs of the component *discriminator* in the quantitative monitor. The discriminator detects novel input by com-

paring both identified semantic classes, delivered by the components environmental and self-perception, respectively, in the autonomous system and the quantitative monitor.

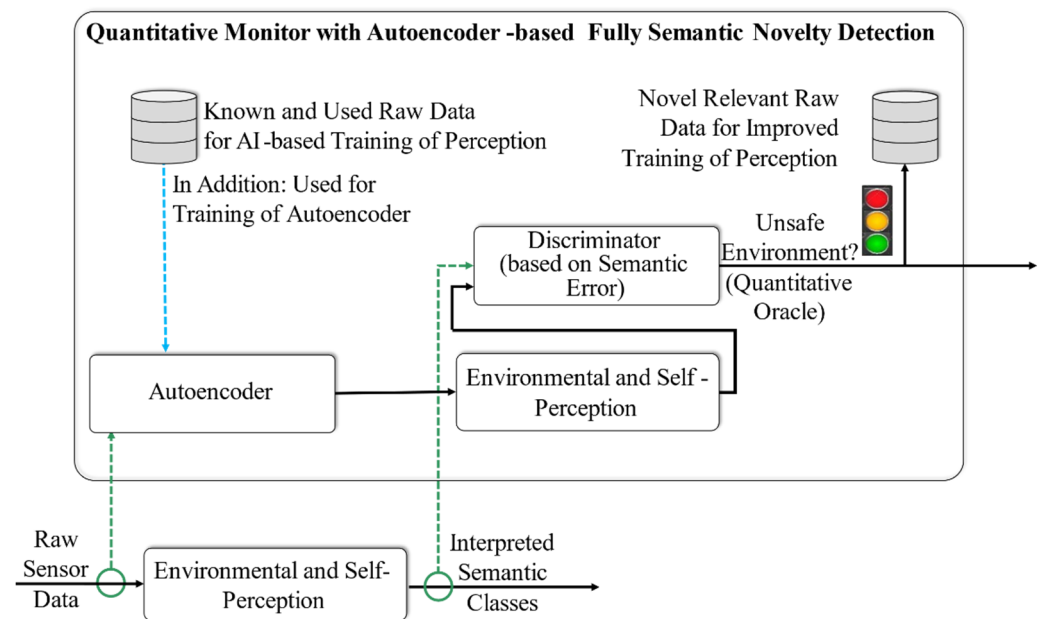


Figure 11. Quantitative monitor with fully semantic novelty detection.

Thus, it can be understood that the quantitative monitor with the fully semantic approach has novelty detection on two different levels. First, the identical component environmental and self-perception classifies the reconstructed raw data provided by the autoencoder. Second, both sets of interpreted semantic classes are compared within the component discriminator on a higher semantic level due to the consideration of integrated semantic errors. Thus, the false negative errors due to similarly structured but semantically different novel data can be minimized.

8.2. Evaluation

According to the architecture, a semi-semantic autoencoder and a classification network, deployed in the component of environmental and self-perception, are required in our evaluation experiment. To find the best result, we used the classification network built for the benchmark mentioned in Section 5 as a sample component for environmental and self-perception. Additionally, the semi-semantic autoencoder with five hidden layers (512-128-7-128-512) described in Section 7 is also used here together with the classification network to build an experimental environment for the evaluation of the fully semantic approach for novelty detection.

The classification network classifies the input data based on a minimal permitted probability of 99%, which is used in this study as the threshold for the classification task (cf. Section 5). The autoencoder reconstructs the raw input data and sends the reconstructed data to the classification network. With the same procedure, the classification network calculates the class of the autoencoder's output (reconstructed input) as well as the class of the original raw input data. If the identified classes of both candidates, including the raw input data and the reconstructed input data, are the same, the input data are identified as known. If there is a difference between these two candidates, the input is novel.

As the results of this approach depend on the classification process, we could not build a density diagram. However, the results shown in Figure 12 illustrate a perfect novelty detection, which satisfies our expectation of minimizing false negatives, and thus confirms HYPO 3 (cf. Section 4).

Result table	Recognized as not novel	Recognized as novel
Known digits	93	7
Unknown digits	0	100
Letters	0	100

Figure 12. Performance of applying the fully semantic novelty detection.

9. Summary and Outlook

ML is generally known to lack robustness, for example, in the case that the training data did rarely or completely not cover critical situations. Nevertheless, many AI-based systems fulfill safety-critical tasks such as in the case of AVs. Engineers cannot completely test and verify autonomous systems in advance during development due to many reasons. Thus, novelty detection for automatically identifying new relevant data differing from the available training data becomes an interesting approach to improve the robustness of AI-based systems, and thereby improve their safety.

In this study, we successfully reproduced the results of a “naive” autoencoder-based novelty detection approach published in [9]. On the basis of a benchmark with a simple classification neural network, we demonstrated that such a “naive” approach is incapable of minimizing false negatives, when novel and known input data are similarly structured but semantically different.

To reduce the number of false negatives, we proposed a semi-semantic approach, which used a semantic topology to ensure that the autoencoder reflected the semantics of the monitored system’s output using several derived rules. These architectural rules were postulated and have been validated with test samples under our observation. To validate the rules, we also considered networks out of the boundaries of these rules. However, they could not beat the networks within the boundaries of the rules. Hence, in our experiments, the semi-semantic approach reduced the percentage of false negatives in unknown digits from 84% to 4%. However, the three rules have only been extensively evaluated, validated, and adjusted in the context of the MNIST dataset. Therefore, further ongoing research is required.

Moreover, we discovered that the classification criterion for novelty detection, which is a threshold solely based on MSE, limits the semi-semantic approach’s ability to reduce the number of false negatives. To overcome this limitation, we proposed a fully semantic approach, which relied on a semantic loss instead of on the original MSE. The semantic loss is realized by a new architectural structure. Instead of comparing with the input data, the loss is calculated based on the output of the AI-based system, in our case referring to the digit classification function, fed with the original input data as well as with the reconstructed input data. Thus, the fully semantic approach can detect novelty on the input data level and also on a higher semantic level. Consequently, the critical cases of novelty detection regarding similarly structured but semantically different novel data were eliminated. According to the comparison of the fully semantic and the semi-semantic approaches in our experiments, the percentage of false negative errors for the classification of unknown digits was reduced from 4% to 0%.

The results of our study demonstrate that semantic novelty detection can prevent false negative errors, especially for structurally similar but semantically different input data. However, false positive errors increase. From a safety perspective, false positive errors are less critical than false negative errors in novelty detection.

In this study, datasets such as handwritten digit images and letters are used in the experiments, which are different from data used in the real-world domain for AVs. Another limitation is that our test samples and experiments do not allow generalizing the presented results (architectural rules and structure). In the future, real traffic images are required to

be used to evaluate the proposed semantic novelty detection. The postulated architectural rules must be validated with more experiments. Moreover, other network types should be evaluated. This will lead us to a semantic novelty detection approach for a quantitative monitor to efficiently improve the quality of training data, and thus the safety of the monitored AI-based systems in the long run, for example, in the case of AVs. Thereby, it could serve as a safety argument for the approval of AI-based systems.

10. Patent

The presented approach for autoencoder-based semantic novelty detection in this paper has been registered as patent: EP 21 181 014.8 and DE 10 2020 122 735.3, “KI-basierte automatische Erkennung von neuen und relevanten Datensätzen”, on 31 August 2020.

Author Contributions: Conceptualization, A.R., A.M.S. and M.Z.; Methodology, A.R.; Software, A.M.S.; Validation, A.M.S.; Formal Analysis, A.R.; Investigation, A.R., A.M.S. and M.Z.; Resources, A.R. and A.M.S.; Data Curation, A.M.S.; Writing—Original Draft Preparation, A.R., A.M.S. and M.Z.; Writing—Review & Editing, A.R. and M.Z.; Visualization, A.R., A.M.S. and M.Z.; Supervision, A.R.; Project Administration, A.R.; Funding Acquisition, A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The MNIST-dataset is available online: <https://chromium.googlesource.com/external/github.com/tensorflow/tensorflow/+r0.7/tensorflow/g3doc/tutorials/mnist/download/index.md>, accessed: 20 September 2021. The notMNIST-dataset is available online: yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html, accessed: 20 September 2021.

Acknowledgments: We thank Yunsu Cho, who came to our research team as a new employed research assistant. Following a prefinal version of this paper, she was able to reproduce our whole experiments, and thereby independently validated our research results. Moreover, we thank our internal reviewers Rüdiger Ehlers and Adina Aniculaesei. They provided many helpful comments to improve this paper. And finally, we thank all anonymous referees for their useful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Williams, A.P.; Scharre, P.D. *Autonomous Systems: Issues for Defence Policymakers*; NATO, Capability Engineering and Innovation Division, Headquarters Supreme Allied Commander Transformation: Norfolk, VA, USA, 2015. Available online: https://www.act.nato.int/images/stories/media/capdev/capdev_02.pdf (accessed on 16 August 2020).
2. Youtie, J.; Porter, A.; Shapira, P.; Woo, S.; Huang, Y. *Autonomous Systems: A Bibliometric and Patent Analysis*; Studien zum deutschen Innovationssystem 14-2018; Exptertenkommission Forschung und Innovation (EFI)—Commission of Experts for Research and Innovation: Berlin, Germany, 2018.
3. Müller, V.C.; Bostrom, N. Future Progress in Artificial Intelligence: A Survey of Expert Opinion. In *Fundamental Issues of Artificial Intelligence*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2016. [CrossRef]
4. Rushby, J. *Technical Report: Quality Measures and Assurance for AI Software*; NASA, Computer Science Laboratory, SRI International: Menlo Park, CA, USA, 1988.
5. Harel, D.; Marron, A.; Sifakis, J. Autonomics: In search of a foundation for nextgeneration autonomous systems. Inaugural article. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 17491–17498. [CrossRef] [PubMed]
6. Aniculaesei, A.; Grieser, J.; Rausch, A.; Rehfeldt, K. Towards a holistic software systems engineering approach for dependable autonomous systems. Proceeding of the 1st International Workshop on Software Engineering for AI in Autonomous Systems, Gothenburg, Sweden, 28 May 2018.
7. Pimentel, M.A.F.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. *Signal Process.* **2014**, *99*, 215–249. [CrossRef]
8. Japkowicz, N.; Myers, C.; Gluck, M. A novelty detection approach to classification. In Proceedings of the 14th IJCAI Conference, Montreal, QC, Canada, 20–25 August 1995; pp. 518–523.
9. Richter, C.; Roy, N. Safe visual navigation via deep learning and novelty detection. In Proceedings of the Robotics: Science and Systems Foundation, Massachusetts Institute of Technology, Online, Cambridge, MA, USA, 12–16 July 2017.

10. Amini, A.; Schwarting, W.; Rosman, G.; Araki, B.; Karaman, S.; Rus, D. Variational Autoencoder for End-to-End Control of Autonomous Driving with Novelty Detection and Training De-biasing. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
11. Byran Christie Design. Driverless Car (IEEE), Online Image. Available online: <https://spectrum.ieee.org/image/MjkyOTE3Nw.jpeg> (accessed on 4 September 2021).
12. LeCun, Y.; Cortes, C.; Burges, C.J.C. MNIST Dataset of Handwritten Digits. 1999. Available online: <https://chromium.googlesource.com/external/github.com/tensorflow/tensorflow/+r0.7/tensorflow/g3doc/tutorials/mnist/download/index.md> (accessed on 20 September 2021).
13. Mauritz, M.; Howar, F.; Rausch, A. Assuring the Safety of Advanced Driver Assistance Systems Through a Combination of Simulation and Runtime Monitoring. In Proceedings of the International Symposium on Leveraging Applications of Formal Methods, Rhodes, Greece, 17–19 October 2016.
14. Mauritz, M.; Rausch, A.; Schäfer, I. Dependable ADAS by Combining Design Time Testing and Runtime Monitoring. In Proceedings of the FORMS/FORAM: 10th Symposium on Formal Methods, Braunschweig, Deutschland, 30 September 2014.
15. Mauritz, M. Engineering of Safe Autonomous Vehicles through Seamless Integration of System Development and System Operation. Ph.D. Dissertation, Technische Universität Clausthal, Clausthal-Zellerfeld, Germany, 28 June 2019.
16. Mauritz, M.; Howar, F.; Rausch, A. From Simulation to Operation: Using Design Time Artifacts to Ensure the Safety of Advanced Driving Assistance Systems at Runtime. In Proceedings of the International Workshop on Modelling in Automotive Software Engineering 2015, Ottawa, ON, Canada, 22 September 2015.
17. Behere, S.; Törngren, M. A functional reference architecture for autonomous driving. In Proceedings of the IEEE 2015 1st International Workshop on Automotive Software Architecture (WASA), Montréal, QC, Canada, 4 May 2015.
18. Behere, S.; Törngren, M. A functional reference architecture for autonomous driving. *Inf. Softw. Technol.* **2016**, *73*, 136–150. [[CrossRef](#)]
19. Mauer, M.; Gerdes, J.C.; Lenz, B.; Winner, H. *Autonomes Fahren: Technische, Rechtliche und Gesellschaftliche Aspekte*, 2015th ed.; Springer Vieweg: Berlin/Heidelberg, Germany, 2015; ISBN 978-3-662-45853-2.
20. Aniculaesei, A.; Grieser, J.; Rausch, A.; Rehfeldt, K.; Warnecke, T. Graceful Degradation of Decision and Control Responsibility for Autonomous Systems based on Dependability Cages. In Proceedings of the 5th International Symposium on Future Active Safety Technology toward Zero, Blacksburg, VA, USA, 28–30 September 2019.
21. Hodge, V.J.; Austin, J. A survey of outlier detection methodologies. *Artif. Intell. Rev.* **2004**, *22*, 85–126. [[CrossRef](#)]
22. Miljković, D. Review of Novelty Detection Methods. In Proceedings of the IEEE 33rd International Convention MIPRO, Opatija, Croatia, 24–28 May 2010.
23. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv.* **2009**, *41*, 1–58. [[CrossRef](#)]
24. Markou, M.; Singh, S. Novelty detection: A review—part 2: Neural network based approaches. *Signal Proc.* **2003**, *83*, 2499–2521. [[CrossRef](#)]
25. Diaz, I.; Hollmen, J. Residual generation and visualization for understanding novel process conditions. In Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN'02, Cat. No.02CH37290), Honolulu, HI, USA, 7 May 2002; IEEE: Honolulu, HI, USA, 2002; pp. 2070–2075. [[CrossRef](#)]
26. Sohn, H.; Worden, K.; Farrar, C.R. Novelty detection under changing environmental conditions. In Proceedings of the SPIE's 8th Annual International Symposium on Smart Structures and Materials, Newport Beach, CA, USA, 4–8 March 2001.
27. Worden, K. Structural fault detection using a novelty measure. *J. Sound Vib.* **1997**, *201*, 85–101. [[CrossRef](#)]
28. Byungho, H.; Sungzoon, C. Characteristics of auto-associative MLP as a novelty detector. In Proceedings of the International Joint Conference on Neural Networks (IJCNN'99, Cat. No.99CH36339), Shenzhen, China, 18–22 July 2001; pp. 3086–3091. [[CrossRef](#)]
29. Manevitz, L.M.; Yousef, M. Document classification on neural networks using only positive examples. In Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, Athens, Greece, 11–15 July 2000.
30. Bulatov, Y. notMNIST Dataset, September 2011. Available online: Yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html (accessed on 4 September 2021).
31. Chollet, F. Keras. 2015. Available online: <https://keras.io/> (accessed on 29 September 2021).
32. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: Large-scale machine learning on heterogeneous systems, Google Brain. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2015. Available online: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>; <https://www.tensorflow.org/> (accessed on 29 September 2021).
33. Open Robotics. Robot Operating System 2. Available online: <https://index.ros.org/about/> (accessed on 4 September 2021).
34. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980v9. Available online: <https://arxiv.org/abs/1412.6980> (accessed on 29 September 2021).