Towards Functional Safety Compliance of Recurrent Neural Networks

Davide Bacciu[1], Antonio Carta[1], Daniele Di Sarli[1],
Claudio Gallicchio[1], Vincenzo Lomonaco[1], Salvatore Petroni[2]
{davide.bacciu, antonio.carta, daniele.disarli, claudio.gallicchio, vincenzo.lomonaco}@unipi.it[1],
salvatore.petroni@marelli.com[2]

[1]Department of Computer Science, University of Pisa, Pisa, Italy,
[2]Legal and Compliance Department, Marelli Europe S.p.A., Turin, Italy

Abstract. Deploying Autonomous Driving systems requires facing some novel challenges for the Automotive industry. One of the most critical aspects that can severely compromise their deployment is Functional Safety. The ISO 26262 standard provides guidelines to ensure Functional Safety of road vehicles. However, this standard is not suitable to develop Artificial Intelligence based systems such as systems based on Recurrent Neural Networks (RNNs). To address this issue, in this paper we propose a new methodology, composed of three steps. The first step is the robustness evaluation of the RNN against inputs perturbations. Then, a proper set of safety measures must be defined according to the model's robustness, where less robust models will require stronger mitigation. Finally, the functionality of the entire system must be extensively tested according to Safety Of The Intended Functionality (SOTIF) guidelines, providing quantitative results about the occurrence of unsafe scenarios, and by evaluating appropriate Safety Performance Indicators.

Keywords: Functional Safety, Dependability, Recurrent Neural Networks, Autonomous Driving, Safety Performance Indicators.

Nowadays artificial intelligence technologies are increasingly in demand for applications in several domains, ranging from gaming, finance, e-commerce, medicine, social media, education, home automation, entertainment, automotive, and others. The type of application in which the intelligent system is used determines additional properties that the latter must respect. If we consider intelligent systems that perform object recognition on autonomous vehicles, we shall take into account that the system shall comply with the *Functional Safety* for road vehicles [1] because without safety assurance the system can cause physical injuries or damage to the health of persons. For this reason, it is important to formally define the design process and the building blocks of safe intelligent systems, in order to develop such systems according to requirements imposed by the proper functional safety technological standard.

CAIP 2021, November 20-24, Bologna, Italy

DOI 10.4108/eai.20-11-2021.2314139

Today, the main building block of modern intelligent systems are Neural Networks [2], which are large adaptive models with potentially millions of parameters, trained to solve complex computational tasks, such as object recognition of road signs. Depending on the underlying task and data format, different network architectures may be used. In this paper, we focus on Recurrent Neural Networks (RNNs) [3], a class of learning models designed to model temporal dependencies. RNNs are a widely studied model, with state-of-the-art performance in several fields [4, 5] and they are fundamental whenever the input data has a temporal dimension and the learning task requires a system able to model the evolution over time of the state. For example, in the automotive domain, an RNN can be used to perform human state monitoring [6] by predicting the psychological state of the driver using physiological data. Human state monitoring can be used to detect a problematic driver's state (e.g. distracted, tired, drunk) [7]. If the decision-making model considers that the driver is not able to drive the vehicle, the system shall actuate safety measures to avoid disastrous consequences for the driver's health. A mistake in the prediction of the psychological state of the driver could allow the driver to control the vehicle despite their altered psychological state. This scenario can cause an accident with negative impacts on the safety of the driver, passengers, and people in close proximity to the vehicle. For this reason, the prediction of the psychological state of the driver is a safety-related task, which means that the system needs to be designed following Functional Safety guidelines.

A feature of RNNs and, in general, of all the Neural Networks, is that the core of these software elements is not easily interpretable by humans and can be considered as a black box. Even with a completely deterministic model, the RNNs computations are complex and difficult to understand under all the possible scenarios. As a consequence, the model may fail in unpredictable ways. Keeping this consideration in mind, we can say that the most popular Functional Safety standard for Road Vehicles, the ISO 26262 [1] cannot be applied to design and test Neural Networks because it refers to the development of traditional software, where the behaviour is simpler and explicitly defined by the programmer. A new standard that can overcomes this problem is the ISO 21448 (version prepared for DIS), also known as SOTIF [8], a standard born to address the challenges introduced by autonomous driving systems with automation levels from 1 to 5. SOTIF analyzes the possible behaviours of a function of the system (or the possible behaviours of a single element) that differ from the intended/desired behavior to verify if there are possible known scenarios that can be exploited to harm people. Furthermore, SOTIF tries to find out also possible unknown scenarios that can harm people. Once scenarios of potential risk for people's health have been discovered, SOTIF provides guidelines to mitigate the risk to an acceptable level. When all the possible safety risks have been mitigated to an acceptable value, the function can be released. One key requirement posed by SOTIF on each algorithm, component, and, in general, to the entire system is robustness. Robustness is usually understood as the ability of a system to react to adverse events, such as noise injection to the system inputs.

In this paper, we propose to verify the robustness of RNNs with respect to inputs perturbations, such as those generated by systematic errors in the sensors data acquisition, environmental conditions, or adversarial perturbations [9]. To ensure the safety, RNNs must be be robust to all these different noise sources. We propose a methodology that uses the robustness of the model, computed with state-of-the-art methods such as POPQORN [10], with respect to a range of accuracy values.

By themselves, this robustness analysis of the RNN does not provide sufficient information about the safety of the RNN. For this reason, our methodology provides also a method to evaluate how often we are potentially unsafe through the use of Safety Performance Indicators (SPIs) [11] that count the number of unsafe occurrences. Depending on the specific needs of the application, a set of appropiate SPIs can be defined, along with the target values to be reached. Finally, a number of test scenarios must be performed and evaluated for each SPI.

1 Adversarial Robustness in Recurrent Neural Networks

RNN Recurrent Neural Networks (RNNs) are typically used to learn tasks on sequential data. In their simplest form, the evolution of the state $\mathbf{h}(t) \in \mathbb{R}^{N_H}$ of an RNN is described by:

$$\mathbf{h}(t) = \tanh\left(\mathbf{W}_{in}\mathbf{x}(t) + \hat{\mathbf{W}}\mathbf{h}(t-1)\right), \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{N_X}$ is the input at time t, $\mathbf{W}_{in} \in \mathbb{R}^{N_H \times N_X}$ is the input-to-recurrent parameter matrix, and $\hat{\mathbf{W}} \in \mathbb{R}^{N_H \times N_H}$ is the recurrent parameter matrix. The bias term is omitted for simplicity. At any given time step t, an output $\mathbf{y}(t) \in \mathbb{R}^{N_Y}$ can be extracted from the state of the network as:

$$\mathbf{y}(t) = \mathbf{W}\mathbf{h}(t), \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{N_Y \times N_H}$ is another parameter matrix. The network is trained end-to-end by computing the gradient of the error with respect to the parameters. After the gradients have been computed, all the parameters in $\mathbf{W}_{in}$, $\hat{\mathbf{W}}$ and $\mathbf{W}$ are updated accordingly.

Adversarial Robustness Adversarial examples are carefully crafted inputs where a small amount of noise is added to an image to induce a wrong classification [9]. POPQORN [10] is a method for the quantification of robustness in RNNs. Given a set of inputs $\mathbf{X} = \{\mathbf{x_0}, \ldots, \mathbf{x_n}\}$ and a l_p ball with radius ε, POPQORN computes two linear functions that bound the RNN's output for each input in $\mathbf{X}$ and additive noise with radius ε. In this paper, POPQORN is a critical component of our evaluation methodology, and provides the robustness values that determines the quality of the model.

2 Robustness and Safety Guarantees for Dependable RNNs

The proposed methodology is aimed to evaluate the applicability of an RNN in an automotive context with Functional Safety implications. The evaluation of the safety of the RNN passes through three successive phases: the first phase evaluates the robustness of the RNN against adversarial perturbations applied to its inputs; the second phase is aimed towards identifying and understanding the proper safety measures necessary to perform plausibility checks and recovery actions when needed; finally, the third phase is aimed to validate the entire system by determining suitable indicators (SPIs), their target value to be reached and test length.

2.1 Determination of RNN adversarial robustness by inputs perturbation

The first phase relies on the evaluation of robustness using POPQORN[10], where we measure how much noise can be injected into the input samples before the RNN's accuracy decreases below a predefined threshold. More formally, given an input sequence X_0 we can add noise and move towards the input $X_0^{'}$ that is at a distance Δ from X_0 (Figure 1). Let us focus on a sequence classification task: for small values of Δ, the correct output of the network should be the same class of the original sequence, therefore hinting at a robustness of the RNN to small perturbations.

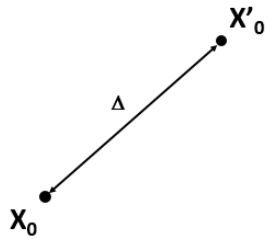

Fig. 1. Sample X_0 and perturbed sample $X_0^{'}$ at a distance Δ.

Now, let us provide a sample X_0 as input to a properly trained RNN and suppose that the RNN will provide as output a correct prediction Y_0. We can identify a region of space around the sample X_0 such that all input samples $X_0' = X_0 + \Delta, \Delta < d$ inside the region will be correctly classified as the class Y_0 (Figure 2), with a determined accuracy (e.g. accuracy greater then 95%).

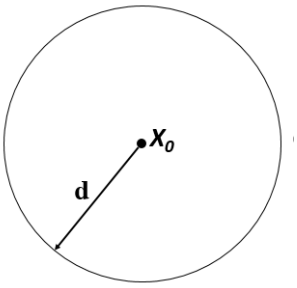

Fig. 2. Input space around X_0 providing a correct prediction with a detrmined accuracy.

The goal of this first analysis is to determine three input spaces around X_0, where all the samples inside each region are correctly classified with a probability above a specified minimum value (e.g. 0.95, 0.8, 0.5). As a result, we obtain a measure of the local robustness of the RNN around

the original input X_0, which are the three concentric hyperspheres corresponding to the different accuracy levels (Figure 3). To accomplish this goal, we define three different RNN output threshold values: a_1, a_2, a_3 with the following relations: $a_3 < a_2 < a_1$. After, we shall apply different perturbations to the input sample X_0 to generate three different spaces as follows (Figure 3):

- S_1, space of the perturbed inputs with distance $d \leq d_1$ from X_0;
- S_2, space of the perturbed inputs with distance $d_1 < d \leq d_2$ from X_0;
- S_3, space of the perturbed inputs with distance $d_2 < d \leq d_3$ from X_0.

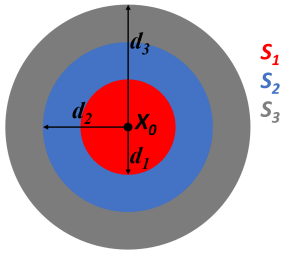

Fig. 3. Input samples with distance $d \leq d_1$, $d_1 < d \leq d_2$ and $d_2 < d \leq d_3$ from X_0 belonging respectively to spaces: red, blue and grey.

Indicating with s a generic sample, we want to determine the values d_1, d_2 and d_3 such that:

- providing as RNN input the samples $s \in S_1$, the corresponding outputs have an accuracy $a \geq a_1$;
- providing as RNN input the samples $s \in S_2$, the corresponding outputs have an accuracy $a_2 \leq a < a_1$;
- providing as RNN input the samples $s \in S_3$, the corresponding outputs has an accuracy $a_3 \leq a < a_2$.

The process to determine the values d_1, d_2 and d_3 must be repeated over an adequate number of samples in order to have a more robust evaluation of d_1, d_2 and d_3. To accomplish this, a number N of input samples must be considered, resulting in N different sets of values of d_1, d_2 and d_3, one set for each sample:

$$
\begin{array}{lll}
d_1^0 & d_2^0 & d_3^0 \quad \text{for } X_0 \\
d_1^1 & d_2^1 & d_3^1 \quad \text{for } X_1 \\
d_1^2 & d_2^2 & d_3^2 \quad \text{for } X_2 \\
 & & \ldots \\
d_1^N & d_2^N & d_3^N \quad \text{for } X_N.
\end{array}
$$

Table 1: Experiments on WESAD. The results show the accuracy computed on the test set and the robustness values statistics.

	Accuracy	Robustness			
		min	mean	max	std
RNN-64	0.83	0.0014	0.0890	0.2717	0.0717
RNN-128	0.86	0.0008	0.0022	0.0079	0.0014
RNN-256	0.89	0.0031	0.0353	0.1361	0.0334

We can compute average/max statistics from these values to determine the RNN robustness as indicated below:

$$d_1 = mean\{d_1^0, d_1^1, d_1^2, ..., d_1^N\}$$
$$d_2 = mean\{d_2^0, d_2^1, d_2^2, ..., d_2^N\}$$
$$d_3 = mean\{d_3^0, d_3^1, d_3^2, ..., d_3^N\}.$$

Notice that we do not give here any minimal values for d_1, d_2, d_3 since these will depend on the specific application and the chosen samples. The values should be used to compare between different models.

2.1.1 Experimental Assessment.

To give an example of the quantification of robustness using our method above, we provide an experimental evaluation of RNNs on WESAD[12], a stress recognition dataset with physiological and motion data from the users. In this paper, we only provide a very preliminary evaluation. We did not perform a large scale evaluation of recurrent models, and we use hyperparameters which were found optimal from our previous internal experiments. We leave a formal experimental evaluation as future work.

Table 1 shows the results for a vanilla RNN with 64, 128, 256 hidden units, respectively. Each model is trained to optimize the cross-entropy on the training data, using an Adam optimizer[13] for 200 epochs. After each epoch, the current model is evaluated on a validation set, and the best model is restored after the training loop is completed. The results in the table are computed on a separate test set. After training, we compute the prediction's robustness over a set of 50 samples, by finding the amount of noise needed to craft an adversarial example. We use POPQORN to compute such values and show the results in Table 1.

Notice that here we do not consider threshold values d_1, d_2, d_3, and instead we compute the amount of noise necessary to craft an adversarial example. Finally, notice that the robustness value can be hard to interpret by looking at its absolute value. It is better instead to compare different models against each other. For example, in our experiment the average robustness of the models shows that the most robust model, RNN-64, is also the less accurate.

2.2 Design of safety measures for plausibility checks

The second phase is aimed to evaluate the results achieved from the previous phase to establish which are the more appropriate safety measures that shall be applied. In this context safety measures can consist in a plausibility check to verify the information provided by the RNN. The plausibility check is provided by using one or more parallel redundant systems that can be different algorithms that exploit the same inputs of the main system based on the RNN or can be systems with algorithms and sensors of different technology (for example radar or lidar) and so on. After the RNN based system and the redundant systems have computed their input data, there is a third system, the comparison system, that is responsible to perform a comparison of the results achieved and shall decide if the output of the RNN based system can be plausible or not (Figure 4).

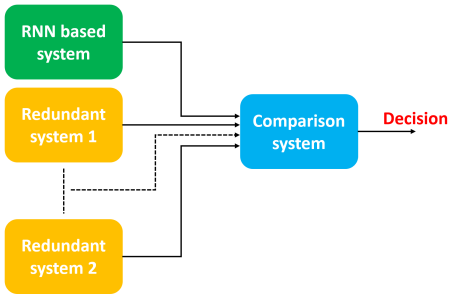

Fig. 4. The comparison system performs the plausibility check among the RNN output and the output of one or more redundant systems and makes a decision.

In the case that the output of the RNN based system is judged not plausible by the comparison system and there is no condition for making a decision, the comparison system shall bring the system (and the vehicle) to the proper safe state depending the current situation. However, to decide which is the more appropriate safety measure, the size of the spaces S_1, S_2 and S_3 determined during the evaluation of the adversarial robustness of the RNN shall be taken into account. For example, in the case that S_1 is much bigger than S_2 and S_3 can be considered negligible, we can consider the RNN quite reliable and so the safety measure can be constituted by a unsophisticated redundant system. A different case, for instance, involves a configuration where S_1 is bigger than S_2 but it is not predominant compared to this latter and S_3 can not be considered negligible; in this situation we shall design a more robust safety measure consisting of more redundant systems. The safety measure, comprising one or more systems, as well as the system based on the RNN and the comparison system, shall comply with specific time constraints provided by the application requirements. Safety critical applications shall operate in real time and so time constraints shall be considered during the design of the RNN based system, the reduntant system (or systems) and the comparison system.

2.3 Safety validation: determination of SPIs and test length

The third phase consists in validating the achieved overall system, which includes the RNN based system and the safety measures adopted (redundant system or systems).

The SOTIF provides strategies to verify and validate the system, determining whether the risk associated to the function is reasonable (and so acceptable) or not. The verification step consists in the testing of the funcion against the known hazardous scenarios, that are those situations in which the function does not behave as expected causing a potential harm for involved people. The goal of the tests is to demonstrate that the potentially hazardous scenarios have been properly managed and the associated risk previously discovered can now be considered reasonable and so acceptable.

After the verification, the functionality of the system shall be validated. The validation consists in the execution of tests to discover if there are unknown scenarios that can be potentially hazardous and so cause harm for involved people. To discover such unknown scenarios a series of tests are performed; such tests are aimed to observe the behaviour of the function in as many real life scenarios as possible: if the behaviour deviates from the desired one and a potentially hazardous situation causing an unreasonable risk for the safety of people is found, some (or additional) safety measures shall be planned and developed to reduce the risk at an acceptable value. To measure the performance of a functionality, the SOTIF suggests KPIs as metrics; the KPIs are aimed to evaluate the performance of the functionality, that is the quality of the functionality. But from a Functional Safety point of view, we are interested not exactly to the general quality of the functionality, that is how well the functionality performs, but we are interested to evaluate how often the functionality is potentially unsafe. For this reason it is better to use indicators that give a measure of the safety performance, the so called SPIs (Safety Performance Indicators) [11]. The SPIs give a measure about the dangerousness of the functionality (including the RNN) being tested, by telling us (for example) if there are dangerous misbehaviour, dangerous gaps in the considered ODD (Operational Design Domain), dangerous gaps in fault responses, dangerous defects in requirements, design, etc. In other words, an SPI gives a measure of the arrival rate of adverse events. SPIs shall be determined at different abstraction levels; so, we have SPIs for the overall functionality (or system), SPIs for the immediate sub-functionalities (or sub-systems) up to SPI for the atomic elements such as the RNN based algorithm, sensors, etc.

Once the SPIs have been defined, for each of them you shall define the target value, a threshold value that each SPI shall not exceed to consider the safety related risk associated to the functionality acceptable. This threshold value indicates the risk budget that you do not want to overcome when your tests ended. Before starting of the testing phase, a suitable test length shall be determined [8]. The test length expresses the quantity of hours or mileage you shall test the functionality and can be affected also by the criticalities of the selected test routes.

3 Conclusion

In this paper, we introduced a novel methodology to include the use of Recurrent Neural Networks within the context of Functional Safety compliance for autonomous driving applications. Our proposal is based on robustness and safety guarantees and is proposed in this paper as a work in

progress concept that lays the ground to future analysis and extensive experiments. Overall, the proposed methodology allows to monitor the robustness of the RNN, increase its safety by using redundant systems, and evaluate its safety through a set of SPIs. After the design of the RNN based system, validation with appropriate tests length is performed to quantify the safety performace through proper SPIs and relative target values in order to determine how often we are potentially unsafe. Tests length shall be carefully determined to achieve more reliable SPIs values.

This work can be used as a methodological basis to achieve a RNN based system that is compliant with Functional Safety. In a future work we will define a use case in TEACHING [7] by implementing a human state monitoring system (consisting of RNN based algorithm, sensors, etc.) that receives as inputs physiological data of the driver and predicts his/her psychological state. In this respect, we also find interesting the potential use of Reservoir Computing and Echo State Networks [14, 15], as a way to efficiently train RNNs while enforcing stability in the network's state computation by proper algebraic initialization of the model's parameters. Then, according to the system's robustness against input's perturbations, we will define suitable safety measures.

To test the achieved system (including its safety measures) we will define a proper set of SPIs and for each SPI we will define the proper target value that the system shall reach to be considered safe. At the end we will define the proper test length and we will test the RNN on real life scenarios to evaluate its safety performance.

References

[1] ISO 26262:2018, Road vehicles — Functional safety. International Organization for Standardization; 2018.

[2] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

[3] Kolen JF, Kremer SC. A field guide to dynamical recurrent networks. John Wiley & Sons; 2001.

[4] Lipton ZC, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:150600019. 2015.

[5] Cossu A, Carta A, Lomonaco V, Bacciu D. Continual Learning for Recurrent Neural Networks: an Empirical Evaluation. Elsevier Neural Networks. 2021.

[6] Bacciu D, Di Sarli D, Gallicchio C, Micheli A, Puccinelli N. Benchmarking Reservoir and Recurrent Neural Networks for Human State and Activity Recognition. In: Rojas I, Joya G, Catala A, editors. Advances in Computational Intelligence. Cham: Springer International Publishing; 2021. p. 168–179.

[7] Bacciu D, Akarmazyan S, Armengaud E, Bacco M, Bravos G, Calandra C, et al. TEACHING–Trustworthy autonomous cyber-physical applications through human-centred intelligence. arXiv preprint arXiv:210706543. 2021.

[8] ISO/PAS 21448:2019, Road vehicles — Safety of the intended functionality. International Organization for Standardization;.

[9] Kurakin A, Goodfellow I, Bengio S, et al.. Adversarial examples in the physical world; 2016.

[10] Ko CY, Lyu Z, Weng L, Daniel L, Wong N, Lin D. POPQORN: Quantifying robustness of recurrent neural networks. In: International Conference on Machine Learning. PMLR; 2019. p. 3468–3477.

[11] Koopman P. Safety Performance Indicators (SPIs) for Self-Driving Cars. 2020 Jun.

[12] Schmidt P, Reiss A, Duerichen R, Marberger C, Van Laerhoven K. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In: Proceedings of the 20th ACM international conference on multimodal interaction; 2018. p. 400–408.

[13] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014.

[14] Jaeger H. The "echo state" approach to analysing and training recurrent neural networks – with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report. 2001.

[15] Jaeger H, Haas H. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science. 2004;304(5667):78–80.