Towards the Characterization of Realistic Model Generators using Graph Neural Networks

José Antonio Hernández López and Jesús Sánchez Cuadrado

Department of Computer Science and Systems, Universidad de Murcia

{joseantonio.hernandez6,jesusc}@um.es

Abstract. The automatic generation of software models is an important element in many software and systems engineering scenarios such as software tool certification, validation of cyber-physical systems, or benchmarking graph databases. Several model generators are nowadays available, but the topic of whether they generate realistic models has been little studied. The state-of-the-art approach to check the realistic property in software models is to rely on simple comparisons using graph metrics and statistics. This generates a bottleneck due to the compression of all the information contained in the model into a small set of metrics. Furthermore, there is a lack of interpretation in these approaches since there are no hints of why the generated models are not realistic. Therefore, in this paper, we tackle the problem of assessing how realistic a generator is by mapping it to a classification problem in which a Graph Neural Network (GNN) will be trained to distinguish between the two sets of models (real and synthetic ones). Then, to assess how realistic a generator is we perform the Classifier Two-Sample Test (C2ST). Our approach allows for interpretation of the results by inspecting the attention layer of the GNN. We use our approach to assess four state-of-the-art model generators applied to three different domains. The results show that none of the generators can be considered realistic.

Keywords: Model generators · Realistic models and Graph neural networks · Two-Sample Test

1 Publication

– The paper won the ACM SIGSOFT Distinguished Paper Award.
– DOI: 10.1109/MODELS50736.2021.00015

2 Summary

2.1 Context

Model generators aim to create synthetic models automatically by taking one or more input parameters to configure the expected bounds or shape of the generated models. These types of tools can be applied to many areas of software and
system engineering such as the testing and benchmarking of graph databases [12, 1], to create complex test stubs in the object-oriented field [12, 4] or automated synthesis of prototypical test contexts in the assurance of smart cyber-physical systems [12, 5, 14].

Recent works [14, 12] have established four properties that a model generator should satisfy: 1) **consistency** (the generator creates consistent models which satisfy all well-formedness constraints), 2) **diversity** (the generated models include a sufficiently wide variety of shapes [11]), 3) **scalable** (with respect to the size of a generated model) and 4) **realistic** (models generated cannot be distinguished from the real ones). In particular, a generator is **structurally realistic** if the set of generated models cannot be distinguished from the real ones just by looking at the typed graph structure (ignoring the attribute values) [14, 13, 10].

Our work is situated in the context of model generators. More concretely, we focus on assessing the **structurally realistic** property of these artifacts.

### 2.2 Summary of state-of-the-art

Currently, a set of graph metrics and simple statistics (such as out-degree, dimensional degree, multiplex participation coefficient, etc) are used to assess whether a set of models can be considered similar to a dataset of realistic models [13, 14, 10]. This technique has three important shortcomings. Firstly, summarizing an entire graph model into a set of graph metrics causes an information loss. Secondly, a subset of graph statistics has to be chosen to perform the assessment, but not all metrics are equally effective to perform this task [13]. Thirdly, this approach is not interpretable, in the sense that it does not give us hints to determine why the generator is not realistic.

### 2.3 Proposal

In this paper, we address the task of determining whether a model generator is realistic using a different technique, which overcomes these aforementioned issues. Our approach follows this idea: given two set of models, one generated by a given generator and the other composed by real models. Under the supposition that the generator is realistic (i.e., the synthetic models are realistic), if we build a Graph Neural Network (GNN) [9] trained to distinguish between these two sets, it will not be able to achieve a good performance since the classification problem is impossible. In order to evaluate how realistic a generator is, we use the non-parametric test called Classifier Two-sample Test (C2ST) [3]. The test tells us whether the GNN actually distinguished between real and generated models. Following this approach, we have assessed how realistic four state-of-the-art generators are in three domains.

### 2.4 Contributions

Altogether, this paper presents a novel approach for assessing model generators, which is more robust than previous approaches and has the additional advantage
of being interpretable (i.e., by inspecting the weights of the attention layer of the GNN). Moreover, to the best of our knowledge, this is the first work that applies GNN to software models. Thus, our proposed GNN architecture can be adapted to face other model classification problems (e.g., meta-model classification [6, 7], UML classification [8], etc).

References