Demo: Enhancing Model Checking with LSTM-based Intelligent Filtering

Anonymous Submission

—

Abstract. This demo paper presents a prototype tool that integrates NuSMV with a Long Short-Term Memory (LSTM) network to enhance formal verification. The system automatically extracts valid paths and counterexamples from a given SMV model, trains an LSTM to predict path validity, and provides intelligent filtering and prioritization of verification traces. The demo showcases the pipeline on a realistic landing gear model, highlighting its usability, efficiency, and effectiveness for debugging and verification.

1 Introduction

Formal verification using model checking is essential for safety-critical systems such as avionics, railway, and automotive software. However, model checking suffers from the **state explosion problem**, which makes exploring all possible execution paths computationally expensive. Recent approaches incorporate AI techniques to assist verification by *prioritizing* the most relevant paths or identifying probable counterexamples.

This demo presents a tool that integrates **NuSMV** with a **Long Short-Term Memory (LSTM)** neural network to enhance verification workflows. By predicting the validity of execution paths, the system allows engineers and researchers to focus on critical traces, reduces computational load, and accelerates error detection. The demo highlights a practical pipeline from model input to LSTM-based prediction, applied to the **landing gear system**.

2 System Overview

The demo system consists of five main modules:

- **Input:** SMV model file and a CTL property to verify.
- **Extraction:** NuSMV generates valid paths and counterexamples automatically.
- **Encoding:** Paths are converted into sequences suitable for machine learning.
- **Learning:** LSTM classifier is trained on the extracted sequences to predict validity.
- **Output:** Filtered and prioritized results are generated, and a detailed PDF report is created.

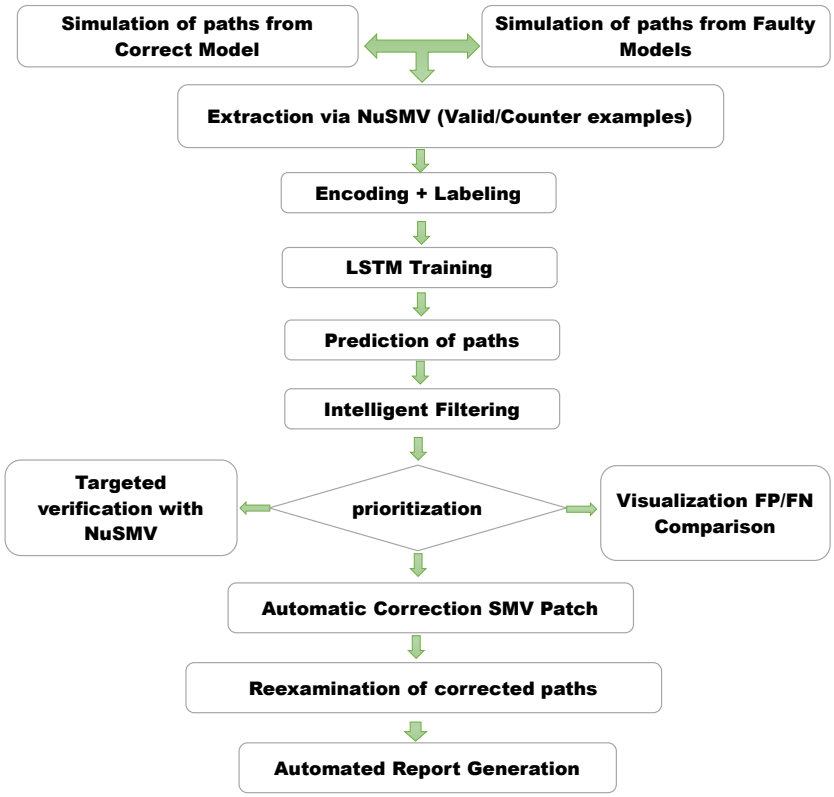

Fig. 1. Architecture of the demo system: from NuSMV extraction to LSTM-based intelligent filtering.

3 Key Features of the Demo

The demo includes several practical and interactive features:

1. Automatic generation of valid and invalid paths from SMV models.
2. Encoding of traces into sequences compatible with LSTM networks.
3. Training of the LSTM and prediction of path validity for unseen traces.
4. Intelligent filtering and prioritization of the most relevant verification paths.
5. Export of results in CSV and PDF formats for analysis and reporting.

4 Demo Walkthrough

We demonstrate the tool using the **Landing Gear system**, a realistic safety-critical example:

1. Load the SMV model file `landing_gear.smv` into the tool.

2. Run NuSMV to check the CTL property `AG (gear_down -> AF gear_locked)`.
3. Generate valid and invalid traces automatically.
4. Train the LSTM on these sequences and predict validity for new paths.
5. Visualize prediction results and download the PDF report containing metrics, filtered paths, and analysis.

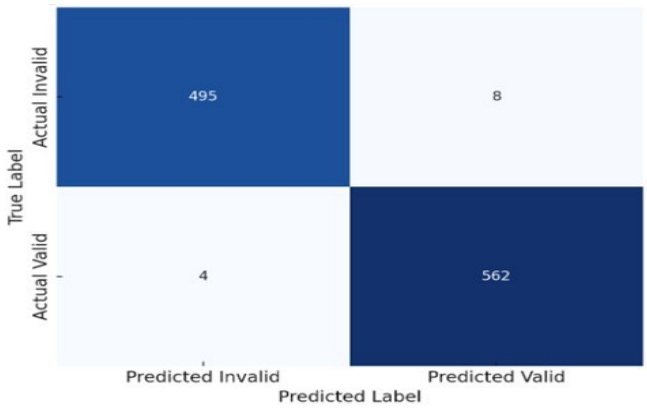

Fig. 2. Example output: confusion matrix and prediction probabilities.

Additional figures can include screenshots of the pipeline interface, path visualization, or example of filtered sequences.

5 Evaluation and Discussion

The LSTM classifier achieves promising results on the landing gear case study.

Table 1. Preliminary LSTM results for path classification

Metric	Accuracy	Precision	Recall	F1-score
LSTM Classifier	0.92	0.90	0.91	0.90

The system reduces the number of paths that need to be examined manually, allowing users to focus on critical traces. This approach demonstrates how AI techniques can enhance formal verification workflows in practice and supports debugging by highlighting suspicious paths early.

6 Conclusion and Demo Details

This demo presents a practical pipeline integrating NuSMV with LSTM-based intelligent filtering. The tool provides researchers and engineers a way to prioritize verification paths and reduce computational overhead while maintaining accuracy.

Demo access: The prototype and instructions are available at: `https://github.com/chaker-nasri/coopis-demo.git`

Participants can run the pipeline on example models, observe LSTM predictions, filter traces intelligently, and download detailed PDF reports.