Software testing of machine learning systems

Yamen Albdeiwi
Lund University
Lund, Sweden
Email: mo4718al-s@student.lu.se

Max Fogwall
Lund University
Lund, Sweden
Email: ma4782fo-s@student.lu.se

Emil Friberg
Lund University
Lund, Sweden
Email: em5435fr-s@student.lu.se

Emil Eriksson
Lund University
Lund, Sweden
Email: em5184er-s@student.lu.se

***Abstract*—In this paper we conduct an exploratory study to investigate methods of using software testing principles to verify the behavior of machine learning systems.**

I. Points of Interest (To Be Removed)

- Relationship between developer, code and actual behaviour (see Figure 2 in [1]) MAYBE
- Hidden feedback loops, two systems (for example customer recommendations and customer reviews) can have an effect on each other [2]. MAYBE
- How is ML code written? Acccording to [2] anti-patterns can become a problem, such as glue code. This makes the system harder to maintain INCLUDED IN EXISTING POINT
- How do ML systems react when there are data problems due to faulty sensors and network problems? discussed in [3] INCLUDED IN EXISTING POINT
- How do you test machine-learned classifiers when only a single user is able to determine if the program is performing correctly? [4] considers the problem of testing this common kind of machine-generated program when the only oracle is an end user. MAYBE
- Grammar Based Directed Testing of Machine Learning Systems, which is the first approach, which provides a systematic test framework for machine-learning systems that accepts grammar-based inputs. [5] MAYBE

II. Decided Points of Interest (To Be Removed)

- From DeepXplore [1] (+ [2]): How can we use neuron coverage as a replacement for code coverage for ML? Is code coverage at all useful for ML? Does it actually reduce the number of faults?
- Why is it important to test ML at all? (This will be talked about in the introduction at least.)
- How does regular software and ML differ in the context of software testing? [2] (+ [3]) (This is a bit unclear, but we'll definitely take up the difference between regular software testing and ML testing, which Eriksson says this concerns.)
- Song et al., who are course responsible, wrote an exploratory study on the subject [6]. This discusses various challenges related to ML testing, as well as approaches including industry standard practices.

III. Introduction

TODO include

- why testing of ML systems is important
- brief coverage of the current state of ML testing
- chosen area of interest
- that we're conducting this from the perspective of the Software Testing course (ETSN20) at the Faculty of Engineering at Lund University (LTH)

A. Description

TODO include

- elaborate on area of interest
- questions we seek to answer
- why we chose this area

IV. Analysis

TODO include

- that it's an exploratory study using literary synthesis

A. Differences

In this section we will discuss the differences between ML and other types of programs, and how they are tested.

B. Problems

Here we discuss various problems in testing caused by the aforementioned differences.

C. Solutions

In this section we discuss potential solutions to the above problems.

V. Results

TODO include

- How many studies are discussed in this research?
- What approach appears to be the most reliable? why?

VI. Conclusion

VII. Contributions

A. Yamen Albdeiwi

- Helped with **Points Of Interest** as well as added some references.
- Some comments have been added about what will be discussed in the results section.

B. Max Fogwall

- Helped with the outline of the document and wrote brief comments on what sections should include.
- Wrote part of the **Abstract**.
- Helped with formatting and expansion of **References**.

C. Emil Friberg

-

D. Emil Eriksson

- Helped with **Points Of Interest** as well as added some references.

References

[1] J. Y. K. Pei, Y. Cao and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," ser. SOSP '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1–18. [Online]. Available: https://doi.org/10.1145/3132747.3132785

[2] D. S. et. al., "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf

[3] J. K. Nurminen, T. Halvari, J. Harviainen, J. Mylläri, A. Röyskö, J. Silvennoinen, and T. Mikkonen, "Software framework for data fault injection to test machine learning systems," in *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2019, pp. 294–299.

[4] A. Groce, T. Kulesza, C. Zhang, S. Shamasunder, M. Burnett, W.-K. Wong, S. Stumpf, S. Das, A. Shinsel, F. Bice, and K. McIntosh, "You are the only possible oracle: Effective test selection for end users of interactive machine learning systems," *IEEE Transactions on Software Engineering*, vol. 40, no. 3, pp. 307–323, 2014.

[5] S. Udeshi and S. Chattopadhyay, "Grammar based directed testing of machine learning systems," *IEEE Transactions on Software Engineering*, vol. 47, no. 11, pp. 2487–2503, 2021.

[6] Q. Song, E. Engstrom, and P. Runeson, "Concepts in testing of autonomous systems: Academic literature and industry practice," in *Proceedings - 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI, WAIN 2021*, ser. Proceedings - 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI, WAIN 2021. United States: Institute of Electrical and Electronics Engineers Inc., 2021, pp. 74–81, 1st IEEE/ACM Workshop on AI Engineering - Software Engineering for AI, WAIN 2021 ; Conference date: 30-05-2021 Through 31-05-2021.