**American University of Beirut**

**ECE Department**

**EECE 796 Special Project Proposal**

**Summer term 2019**

|  |  |
| --- | --- |
| **Title:** | Machine Learning Meets Software Engineering |
| **Student:** | Ahmad Mustapha |
| **ID:** | 201924287 |
| **Supervisor:** | Prof. Wassim Masri |

Abstract**:**

Machine Learning has been successfully applied to different domains. And with the rise of Deep Learning, the applicability and efficiency of Machine Learning have increased. However, these technologies have so far a little impact on the domain of Software Engineering. We argue that things are changing and the relevance of machine learning in Software Engineering is increasing. In this project we aim to study the interaction between Machine Learning including Deep Learning from one side and Software Engineering including Software Testing from the other side. To do this we will distill a collected set of 1516 research papers from five top Software Engineering Conferences published between 2017 and 2019 inclusive and study the papers that applies Machine Learning in Software Engineering and vice versa.

Introduction**:**

The application of Artificial Intelligence on Software Engineer have been conducted since the mid 80’s [1]–[3]. However, and as ML is a subfield of AI, efforts to summarize the interaction between ML and SE on the Application levels is scarce. Notable efforts is done by [4], [5], however those are based on very old research that go back to 80’s and 90’s and it is well known that the domain of machine learning have considerably changed. New publications related to the topic are very broad [6], [7]. The most comprehensive effort have been done by [8], their work directly related to our expected work where they studied around 100 papers that applies Deep Learning on Software Engineering. However, they only targeted Deep Learning and their focus was on models used rather than the types of applications. From this short review one can see that this niche is not yet fully investigated.

Methodology:

In order to get insights about the interaction between machine learning and software engineering we followed the following methodology. We first downloaded all publications from 6 major Software engineering conferences throughout the years 2017-2019. The conferences we considered are:

* IEEE International Conference on Software Testing, Verification and Validation (ICST)
* International Conference on Software Engineering (ICSE)
* The ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)
* IEEE/ACM International Conference on Automated Software Engineering (ASE)
* The ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)
* International Symposium on Software Reliability Engineering (ISSRE)

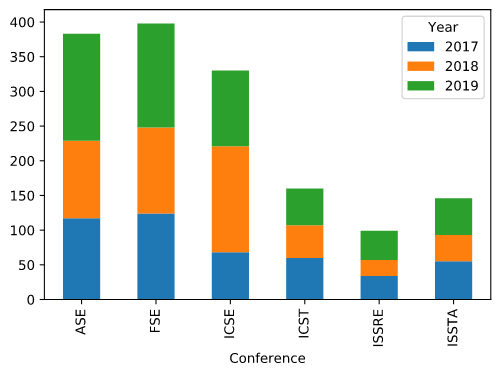
 dsddsThe total number of collected papers was 1516 paper, distributed on the conferences as visible in Figure 1:

Figure The number of papers scraped

In order to filter out the papers that are related to machine learning we proposed the following NLP based approach. We manually defined a bag of words that encapsulates machine learning concepts. The *ML\_tokens* are ["train", "training", "learning", "classify", "classifier", "cluster", "clustering", "regression", "machine", "deep", "neural", "network", "networks"]. We counted the number of ML\_tokens appeared in the **abstract** and **title** of each paper. Note that we only counted whether each token appeared or not discarding the number of times it appeared. We used the sum of the appeared tokens as a metric to score papers relatedness to machine learning. Table 1 shows the distribution of machine learning score.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| score | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| count | 988 | 264 | 139 | 51 | 39 | 19 | 8 | 6 | 1 | 1 |

Table Count Distribution of Machine Learning Score

According to our heuristic 34% of the papers contained at least 1 machine learning term. However, this might not be insightful. As some tokens might appear in a different context to machine learning. For example, “deep” token can appear as “deep understanding” and “network” can appear in the context of computer networks. Using bi-grams might increase the precision of the filtering process, though the naïve approach we used is enough to take a sample to study. The sample we studied for this manuscript is the top 100 papers with respect to machine learning score. The score for those papers ranged from 3 to 9. While studying the papers we found that they actually related to machine learning and the number of false positive is negligible. Figure 2 shows the count distribution and the relative percentage of the studied papers (machine learning papers) with respect to conferences.

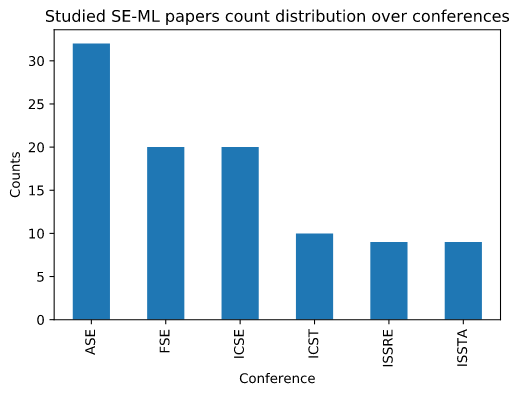
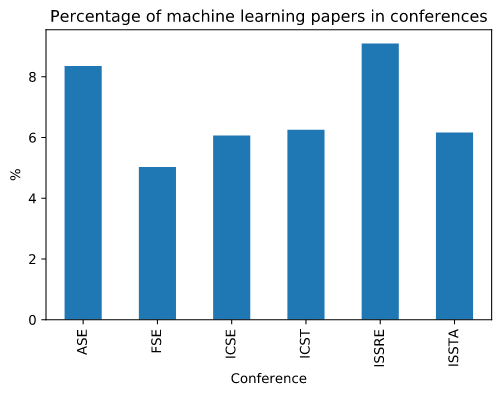


Figure Count Distribution and percentage of studied papers w.r.t conferences

On Average machine learning papers represent 6.8% of published papers in each conference throughout 3 years. Interestingly ISSRE achieved the highest percent (9%) of machine learning papers relative to the number of published papers (approx. 100) in case of ISSRE.

In the following sections we will discuss different insights we abstracted from the studied papers on different dimensions. Note that for time restrictions and the sake of reporting we only studied 40 papers out of the selected 100. We plan to complete our insights based on all the 100 papers.

Machine Learning Insights**:**

Machine Learning have a very recurrent pipeline. The pipeline can be abstracted as follows:

1. Dataset Collection and Preparation
2. Feature Selection
3. Model Selection
4. Model Training
5. Model Evaluation
6. Model Deployment

In this section we will study the unique characteristics of applying Machine Learning in the scope of Software Engineering. Model training, evaluation, and deployment are fine grained details that depends on the selected model and thus will not be used to abstract insights.

Datasets:

Dataset collection and the quality of the used data is very essential and critical for any machine learning application. In fact, the development of machine learning domain (including deep learning) dependent on a special practice performed by the community which is *dataset benchmarks*. The practice is defined by creating different well-developed datasets for every sub-domain and make them easily accessible for the community to work on and use as benchmarks. Most notable is the ImageNet dataset which contained about 1 million images distributed over 1000 class and was fueling traditional computer vision and recently deep learning research. Does the Software Engineering community follow a similar practice?

Insights:

Application:

Models:

Datasets:

Code Representation:

**Reference:**

[1] J. Mostow, “Foreword What is AI? And What Does It Have to Do with Software Engineering?,” *IEEE Trans. Softw. Eng.*, vol. SE-11, no. 11, pp. 1253–1256, Nov. 1985, doi: 10.1109/TSE.1985.231876.

[2] D. Barstow, “Artificial intelligence and software engineering,” in *Proceedings of the 9th international conference on Software Engineering*, Monterey, California, USA, Mar. 1987, pp. 200–211, Accessed: May 01, 2020. [Online].

[3] K. Srinivasan and D. Fisher, “Machine Learning Approaches to Estimating Software Development Effort,” *IEEE Trans. Softw. Eng.*, vol. 21, no. 2, pp. 126–137, Feb. 1995, doi: 10.1109/32.345828.

[4] D. Partridge, “Artificial intelligence and software engineering: a survey of possibilities,” *Inf. Softw. Technol.*, vol. 30, no. 3, pp. 146–152, Apr. 1988, doi: 10.1016/0950-5849(88)90061-4.

[5] D. Zhang and J. J. P. Tsai, “Machine learning and software engineering,” presented at the Software Quality Journal - SQJ, Feb. 2002, vol. 11, pp. 22–29, doi: 10.1109/TAI.2002.1180784.

[6] P. Louridas and C. Ebert, “Machine Learning,” *IEEE Softw.*, vol. 33, no. 5, pp. 110–115, Sep. 2016, doi: 10.1109/MS.2016.114.

[7] K. Meinke and A. Bennaceur, “Machine Learning for Software Engineering Models, Methods, and Applications,” Dec. 2017, doi: 10.1145/3183440.3183461.

[8] X. Li, H. Jiang, Z. Ren, G. Li, and J. Zhang, “Deep Learning in Software Engineering,” *ArXiv180504825 Cs*, May 2018, Accessed: May 01, 2020. [Online]. Available: http://arxiv.org/abs/1805.04825.