Reproducibility Experiment: Identifying Suspicious URLs: An Application of Large-Scale Online Learning

Johannes Garstenauer - University of Passau - *garstenaue@fim.uni-passau.de*

ABSTRACT

This paper is a report on our efforts to replicate the 2009 paper “Identifying suspicious URLs: an application of large-scale Online Learning” [5]. It has achieved impressive in detecting malicious or fraudulent websites using Online Learning. Considering the dynamic and fast-paced nature of machine learning research and technology we reproduced these results in the modern and popular framework “Scikit-learn”. By reproducing the four cornerstone experiments of Ma et al., 2009 (original paper), we concluded, that Online Learning approaches can be successfully applied to the task of detecting malicious URLs. They achieve high accuracies by benefitting from Online Learnings distinct advantages. However, no statement could be made on the inherent superiority of Online Learning in all regards to other forms of machine learning and neither could we support a recommendation of a specific classifier to achieve optimal results. We hope that the contents of this paper may help engineers and researchers in making informed decisions about using Online Learning, especially in solving similar tasks to URL classification.

ACM Reference Format:
Johannes Garstenauer - University of Passau - *garstenaue@fim.uni-passau.de*. 2022. Reproducibility Experiment: Identifying Suspicious URLs: An Application of Large-Scale Online Learning. In *Proceedings of ACM Conference (Conference’17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

When regularly browsing the internet, it is not uncommon to be confronted with websites that have a malicious intent like scamming or phishing users. When coming across such an occurrence the user must decide about the trustworthiness of this website using its URL. The URL, standing for “Uniform Resource Locator” consists of various lexical features (those that can be seen in the URL string, like “https://” or “.com”) and is associated with many more host-based features (meta-information, like the location from where the website is hosted). The regular user though, will only ever conveniently be able to assess the danger of a website using the lexical features of an URL and not the host-based features, thereby already omitting a large pool of potentially viable information. Additionally malicious websites will try to disguise the malicious intent. In conclusion, assessing the danger associated with an URL is, in many cases, nothing more than a “guessing-game”. This presents a prime opportunity for machine learning to aid human users in this task, using all available features of an URL.

To this end the 2009 paper “Identifying suspicious URLs: an application of large-scale online learning” by Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker [5] has delivered promising results. A variant of machine learning, so called “Online Learning” was employed, due to its properties of being highly adaptable to changes in the underlying training data. We have decided to replicate Ma et al., 2009 for a variety of reasons. As stated before, malicious websites are still a major threat on the world wide web. Secondly, replication is a necessary and important task in any scientific endeavour (like machine learning research) and a major underpinning of the scientific method. The code as well as data was provided publicly and can be found online [3] and the scope of the experiments conducted was small enough to allow for reproduction, which made Ma et al., 2009 a good candidate for a reproducibility experiment.

Lastly Ma et al., 2009 is from 2009, which in and of itself poses a threat to the validity of the achieved results. It is no understatement to claim, that the technology and research landscape in Data Science has seen dramatic change within the last decade. Frameworks and technologies commonplace then experience little use now, having been replaced with more advanced and more user-friendly successors. Even computational power has become more available at lower costs. Therefore, we considered it useful to transfer the experiments into a modern framework.

Our paper’s contribution is to see whether the four main claims of the Ma et al., 2009 could be supported by reproducing its four experiments and evaluating the results.

1. "[. . .] successful application of online learning algorithms" [5]
2. "[..] online methods are far better suited to the practical nature of this problem” [5]
3. "[The] Confidence-Weighted algorithm achieves accuracies up to 99%” [5]
4. “continuous retraining [..] is critical for adapting the [online-learning] classifier to detect new, malicious URLs” [5]

In the end we were able to confirm that online learning can be successfully applied to this problem (1), however not that it is inherently superior in accuracy to other machine learning methods (2). For practical reasons we could neither confirm nor deny which algorithm is best used (3), whilst still showing that continuous retraining, over all features, is the most promising approach for online-learning algorithms (4). Therefor we could confirm some of the major claims of the Ma et al., 2009 while having to remain inconclusive about others.

We hope that this paper may help engineers and researchers alike in deciding whether an online learning approach is advisable in their specific objectives. Also, they may be aided in picking a

Conference’17, July 2017, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

proper training regimen and classifier by reading the contents of this paper.

2 BACKGROUND

2.1 Batch and Online Learning

There has been considerable mention of the term "Online Learning" in the context of Machine Learning. This section will delve more deeply into that topic and attempt to explain it. Since the inception of Machine Learning there have been two fundamental approaches: Batch and Online Learning. Batch Learning (what can be considered as the traditional and more common technique) relies on the following conditions [1]:

- The whole training data set can be accessed
- There are no time restrictions

In the absence of those two conditions Batch Learning is not feasible. In our case (URL classification) one might easily imagine a scenario, where this might be the case. Given a practical application which is supposed to detect malicious URLs in real-time. Since the nature of malicious URLs is subject to constant change, the first condition cannot be fulfilled (1). For example, a previously unknown entity might start launching several malicious websites. To react to this emerging threat, the application would have to readjust its model using these features, resulting in the conclusion that the whole data set could not be accessed from the very beginning. This retraining would also infer time restrictions (2) since that operation can be time consuming.

Online Learning on the other hand embraces the fact that learning environments do often change rapidly. This is achieved by the ability to simply update a model with new data, without having to retrain it entirely. This provides the ability to incrementally train and use a model, making it a good fit for the task of detecting malicious URLs.

2.2 From Matlab to Scikit-learn

The Online Learning algorithms and experiments in Ma et al., 2009 are exclusively coded in Matlab. Increasingly Matlab has been replaced with tools like NumPy, Matplotlib and Jupyter, however. Some of the reasons for falling out of favour in many cases, is its sometimes-unusual behaviour and syntax. Furthermore, Matlab is closed source, proprietary and quite expensive. High expenses result in paywalling code and scientific achievements. For those reasons and the wish to replicate the paper in a distinct environment Numpy, Matplotlib and Jupyter were chosen as the tools of choice for the replication. Especially, Scikit-learn deserves to be mentioned, which is generally regarded as "the most comprehensive and open-sourced machine learning package in Python"[2]. It provides a comprehensive coverage of machine learning methods and had most of the classifiers needed for replication pre-implemented and well documented. It stands out from the competition due to its ease of use, API consistency and extensive documentation [6]. Additionally, it provides good performance since it is based on compiled binary libraries that were originally written in Fortran, C++ or C [3]. As a result, we deemed Scikit-learn to be a good fit for our replication purposes.

2.3 Understanding the data

The findings in Ma et al., 2009 were derived from four experiments. Those experiments were run on a dataset of around 2 million URLs. The process of gathering those URLs was described as such: For 100 days around 6.000 − 7.500 of spam and phishing URLs were provided by a large Web mail provider. Benign URLs were gathered from a Yahoo random URL generator. [1] [5] Therefore around 20.000 URLs were gathered per day adding up to 2 million URLs in total in a 2-to-1 benign to malicious ratio. After that, the features of each URL were gathered. The lexical features include hostname, path tokens, primary domain and more. The host-based features include WHOIS info, the IP prefix, location etc. In the end these features were used by the machine learning classifiers to make their classification decision.

3 CONTRIBUTION

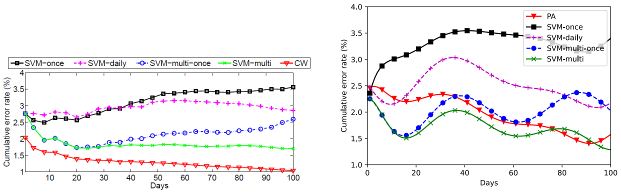
In total we conducted four experiments, to evaluate the claims made by the Ma et al., 2009. The experiments resemble those done in the reproduced paper as closely as possible. Where necessary, small adaptations were made. Those are more closely described in the Discussion section. One that deserves to be mentioned before going on to describe the replication experiments is, finding a suitable Python implementation of the Confidence Weighted algorithm. There are some candidates like the "Exact Soft Confidence-Weighted Learning" [4]. Unfortunately, neither this, nor any other candidate worked due to bugs arising when working with the provided data. In the end, the experiments had to be conducted by omitting the Confidence Weighted algorithm. It was replaced by the second most sophisticated Online Learning algorithm, the Passive-Aggressive classifier, as a means to ensure comparability. Nonetheless, this unresolved issue is a notable limitation to the comparability of reproduction and original.

3.1 Advantages of Online Learning

The first experiment is all about proving, that Online Learning is the superior approach in solving the task of detecting malicious URLs through machine learning. Therefore, the error rates of a Support Vector Machines (SVM) were compared to those of an Online Learning Algorithm. Depicted are the cumulative classification error rates when detecting malicious URL. Since the URLs were collected over a period of 100 days, each graph depicts the progression of error rates over that period. To achieve better comparability, the SVMs were trained using different training regimens. The description of those training regimens are as follows [5]:

- "The SVM-once curve represents training once on Day 0's data and using that model for testing on all other days."
- "SVM-daily retrains only on data collected the previous day - e.g., Day 6 results reflect training on the URLs collected on Day 5, and testing on Day 6 URLs."
- "[. . .] SVM-multi-once trains on data from Days 0 to 16, and from Day 17 on it uses that fixed model for testing on subsequent days."
- "[. . .] SVM-multi trains on the previous 14–17 days worth of data"

By looking at the graphs for the SVMs i Ma et al., 2009, it becomes clear, that the more data and the fresher that data is, the better the



(a) Error rates for CW and for batch algorithms under different training sets. (Ma et al., 2009.)

(b) Error rates for PA and for batch algorithms under different training set (reproduction)

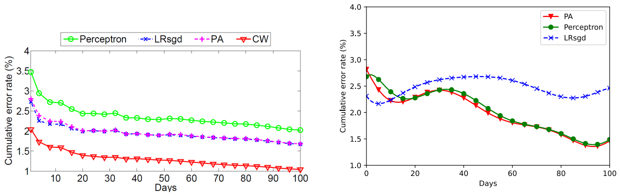
SVMs are in their prediction. Also, the figure can back up their claim, that the Online Learning algorithm (Confidence Weighted Classifier) is still outperforming the best SVM instance (SVM-multi).

We couldn't confidently reproduce those two claims. While we can support the first assertion, concerning the performance of the SVMs themselves, we could not do the same for the second claim, being that Online Learning algorithms show better results. When looking at the final result at day 100 (Fig.2), the SVM-multi instance has outperformed the Online Learning classifier. However, a reasonable case can made, that the same would not hold true if the Confidence Weighted classifier had been available, due to it being the most sophisticated of those used in Ma et al., 2009. As a result, this experiment remains partly inconclusive.

3.2 Comparison of Online Algorithms

After having compared Batch to Online Learning algorithms, the next experiment focusses on comparing Online Learning algorithms. It shows the comparison of the Perceptron, Logistic Regression with Stochastic Gradient Descent, Passive Aggressive and Confidence Weighted algorithms.

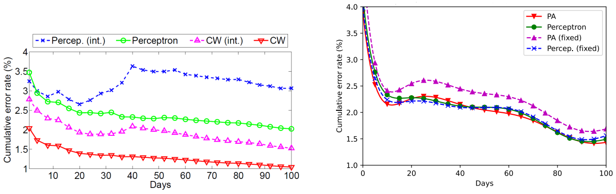
The Perceptron algorithm is the simplest, since it "treats mistakes equally (and ignores all correct classifications)" [5] when it comes to deciding in what way the current classification will impact those following. It achieves the highest error rates around 2-3%. Logistic Regression with Stochastic Gradient Descent (LRsgd) and Passive-Aggressive (PA) algorithm both achieve similar results, nearing a 1.6% error rate. They both can account for classification confidence [5]. The Confidence Weighted (CW) algorithm again achieves the best result, having an error rate as low as 1%.



(a) Error rates for online learning algorithms. (Ma et al., 2009)

(b) Error rates for online learning algorithms (Reproduction)

Overall, the reproduction shows different results. Whereas Logistic Regression with Stochastic Gradient Descent reaches an error rate of 1.6% in Ma et al., 2009, it can only approach 2.5% in the reproduction, while having an increasing error rate over the span of the experiment. The results for the Passive-Aggressive are almost identical to those of the reproduced paper. Interestingly, so is the Perceptron and therefore significantly outperforming its counterpart from Ma et al., 2009.



(a) Benefits of using continuous training over interval-based training (Ma et al., 2009)

(b) Benefits of using continuous training over interval-based training (Reproduction)

In conclusion, while some algorithms perform worse (LRsgd) and some better (Perceptron), only the Passive Aggressive remains comparable from Ma et al., 2009 to the reproduction. The fundamental takeaway from this experiment, was the superiority of the Confidence Weighted algorithm to other Online Learning algorithms in its accuracy, which we were unable to comment on.

3.3 Training Regimen

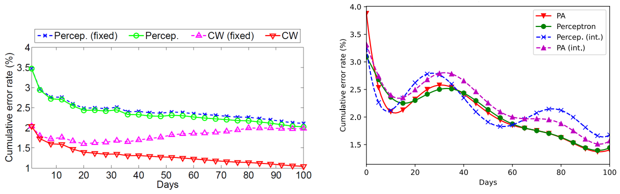
Having examined the Online Learning algorithms performances themselves, the third and fourth experiment focussed on examining what effect different training regimens have on them. In this case training regimen refers to "when the classifier is allowed to retrain itself after attempting to predict the label of an incoming URL" [5] and to "how many features the classifier uses during training" [5].

The third experiment shows a comparison of the Confidence Weighted to the Perceptron algorithm, each with another variant of themselves that differs in its training regimen. One has a continuous training regimen, meaning that the classifier is allowed to retrain itself after each incoming URL. Its variant however has an interval-based training regimen. This means, that it was only allowed to retrain itself after a full days worth of URLs. The figure makes it clear, how Online Learning algorithms benefit from continuous updates and therefore "fresh" data. The continuous instances outperform the interval-based instances significantly.

In the reproduction, the same holds true, only to a lesser degree. Just like in the second experiment the Passive Aggressive Classifier and the Perceptron perform similarly. Their interval-based counterparts do slightly, but noticeably worse. The difference is more strongly pronounced in the Perceptron Algorithm.

Another modification to the training regimen was to reduce the number of features that the classifier can use for its training. The aim of the last experiment was to examine the effects of that on Online Learning algorithms. Accordingly, a comparison of the Confidence Weighted to the Perceptron algorithm was performed, each with another variant of themselves that receives only 150 000 features (those that were encountered within the first day) from 3 million available. While the more simplistic Perceptron does not suffer greatly from this limitation, the Confidence Weighted algorithm does. As the data gets more stale, the error-rates increase.

In the reproduction, again, the Passive Aggressive and the Perceptron algorithm perform similar. Their fixed-feature variants,



(a) Benefits of using variable-feature sets over fixed-feature sets (Ma et al., 2009)

(b) Benefits of using variable-feature sets over fixed-feature sets. (Reproduction)

however, do not. Just like in Ma et al., 2009, the Perceptron instance is not significantly impacted. While the impact on the Passive Aggressive classifier is less pronounced, it is noticeable. In conclusion, we were able to support the claim in Ma et al., 2009, that more sophisticated algorithms do perform better when they are more regularly updated and when the features they encounter are as fresh as possible. Yet the effects of that were less pronounced in the reproduction.

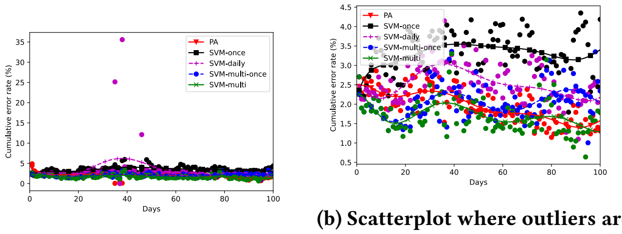
4 DISCUSSION AND LIMITATIONS

Implementing those experiments posed a variety of challenges, because of which some limitations naturally arose. The main issues here were adapting to irregularities in the provided data as well as evaluating some of the choices made by in Ma et al., 2009.

4.1 Limitations from the experimental setup

Firstly, for hyperparameter tuning we found different values to those in the reproduced paper. In a nutshell, Hyperparameter Tuning describes the effort of finding the optimal parameters for a classifier. Ma et al., 2009 proposes to tune the classifiers over one day of data and then picking the best parameters. We replicated this using the sklearn.model_selection.GridSearchCV [6] implementation in Scikit-learn. However, while Ma et al., 2009 proposed some values as optimal, the same results were not achieved using Scikit-learn. While it would have been possible to use the parameters provided by Ma et al., 2009, we thought it would be more useful to attempt to achieve the best results, just like in Ma et al., 2009, to provide better comparability.

4.2 Data inconsistencies



(a) Uncleaned scatterplot.

(b) Scatterplot where outliers are cleaned up.

Furthermore, inconsistency in the data posed a challenge to the replication. This section will attempt to transparently explain the handling of those issues. A telling example of this is: where usually each chunk of the data provided (signifying one day of collected data) has thousands of URLs, the 45th chunk only has 130. These inconsistencies are most likely due to URL feed outages during URL collection, mentioned by Ma et al., 2009. Additionally in the resulting scatterplots, we found significant and otherwise unexplainable outliers. Therefor the decision was made to classify those outliers which are significant and temporally clearly related to the days in and around the feed outages. Those outliers were then removed from the dataset to achieve a fair and balanced comparison in the reproduction and not to have conclusions based on clearly faulty data. This was achieved using functions to identify those outliers and remove them.

4.3 Discussion of experimental parameters

Lastly, varying the experimental setup may be a worthwhile pursuit for future work. This section will discuss one possible variation in the first experiment. As a reminder, the first experiment was about proving the superiority of Online Learning algorithms over Batch Learning. Part of the argument was the reduced use of computing resources necessary for Online Learning as compared to Batch Learning. In this first experiment, where that claim was made, one Batch Learning classifier was trained "on as much data as our evaluation machine with 4 GB RAM can handle" [5]. Since 4GB of RAM is sufficient for only around 14-17 days worth of URLs, it cannot outperform the Online Learning classifier, which at every moment has all data available for making a classification decision. Therefore, the conclusion is drawn, that for Batch Learning "accuracy is fundamentally limited by the amount of computing resources available" [5]. This is correct. However, it would not need to be an issue here if a machine with more RAM was used for evaluation. Since machines with RAM orders of multitude larger are widely spread nowadays, this could be a compelling candidate for future work.

5 CONCLUSION

In conclusion, the reproduction of the 2009 paper "Identifying suspicious URLs: an application of large-scale Online Learning" can confirm some of its major claims while having to remain inconclusive about others. Specifically, we could confirm that Online Learning can be used to detect malicious URLs with high accuracy. We remain inconclusive about whether it is superior to Batch Learning in this task. Especially providing more computing resources to Batch Learning classifiers like SVMs might prove to be interesting future work. Due to the unavailability of a suitable implementation for the most promising algorithm, the Confidence Weighted algorithm, it has yet to be examined which Online Learning algorithm will perform best on this URL classification task. It is clear, however, that implementation details and therefore framework choices will have a significant impact on results. The same applies to the question of training regimen for Online Learning algorithms. They benefit from continuous retraining, over all features. This benefit may however be significantly more or less pronounced in different implementations.

REFERENCES

[1] Óscar Fontenla-Romero et al. "Online machine learning". In: *Efficiency and Scalability Methods for Computational Intellect.* IGI Global, 2013, pp. 27–54.

[2] Jiangang Hao and Tin Kam Ho. "Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language". In: *Journal of Educational and Behavioral Statistics* 44.3 (2019), pp. 348–361. DOI: 10.3102/1076998619832248. eprint: https://doi.org/10.3102/1076998619832248. URL: https://doi.org/10.3102/1076998619832248.

[3] Jiangang Hao and Tin Kam Ho. "Machine learning made easy: a review of scikit-learn package in python programming language". In: *Journal of Educational and Behavioral Statistics* 44.3 (2019), pp. 348–361.

[4] Steven CH Hoi, Jialei Wang, and Peilin Zhao. "Exact soft confidence-weighted learning". In: *ICML*. 2012.

[5] Justin Ma et al. "Identifying suspicious URLs: an application of large-scale online learning". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 681–688.

[6] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.