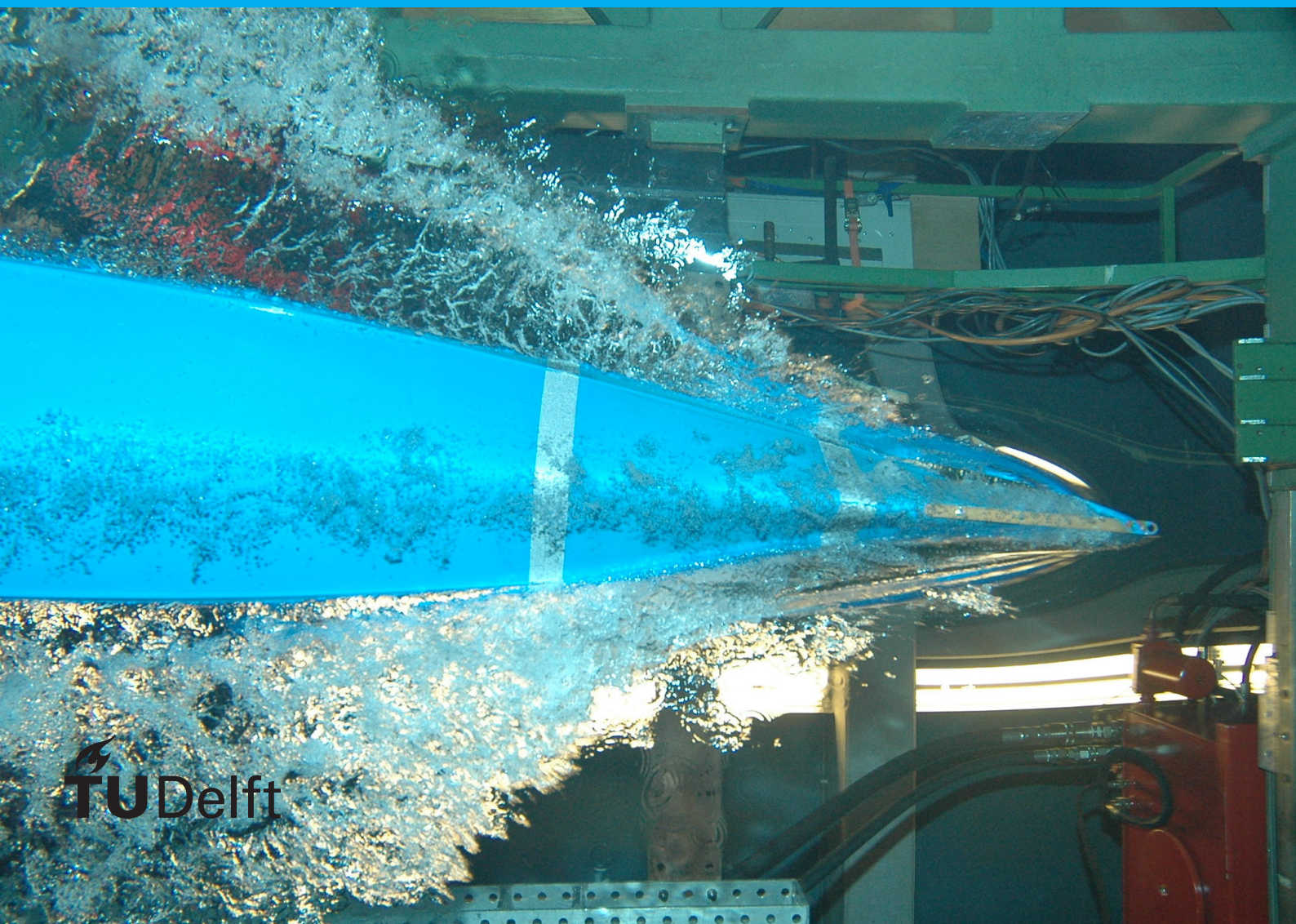
Debugging Machine Learning on Time-series Data

Meng Zheng

Cover Text
possibly
spanning multiple lines

ISBN 000-00-0000-000-0



Debugging Machine Learning on Time-series Data

by

Meng Zheng

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday January 1, 2013 at 10:00 AM.

Student number:	1234567	
Project duration:	March 1, 2012 – January 1, 2013	
Thesis committee:	Prof. dr. ir. J. Doe,	TU Delft, supervisor
	Dr. E. L. Brown,	TU Delft
	Ir. A. Aaronson,	Acme Corporation

This thesis is confidential and cannot be made public until TBD.

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

Preface...

Meng Zheng
Delft, January 2013

Contents

1 Introduction

Machine learning models are so called a "black-box", which means people can not easily observe the relationship between the output and input, or explain the reason for gaining such result. Recent years, much work has been done on interpratable machine-learning, such as Shapley values, counterfactual explanations, partial dependence plots or saliency maps [7]. However, methods based on such works are targeting on model trained with 2-dimensional images static data, little work has been done or implements on debugging models with time-series data [1] [5] [3][6]. This research targets on identifying what a model with time-series dataset really knows and what it don't know. To further evaluate such method, LightDigit which is proposed by Hao Liu etc [4], will be implemented.

LightDigit is an embedded based Machine Learning project, which focus on achieving touch-free interaction with public touchscreens and buttons. This project used a 3X3 photodiodes. The key idea is, recognizing the shadowed area by hand while using finger writing digits above the screen, captured the time series data. These time series data and labels will be feed to train the deep learning model. By iterating multiple epochs, the model shall be able to predict the corresponding digit.

Through Hao Liu's work, still the model can achieve the accuracy of 91 percent. To further improve this work, it could be essential to know what this model really knows, and what this model actually should show. After gaining these knowledge, it will be possible for further research done on improving the performance of this model. Agathe Balayn and etc provided a human-in-the-loop approach for knowing what a model knows [2]. This work proposed the SECA framework, which firstly use Local interpretability method to generate saliency map, then saliency map are provided for annotators to annotates, and finally process these data and gain the result,

However, Agathe's work focus on static data, like images, this project will migrate SECA framework on time-series data, and take LightDigit as a usecase. Find out what such time-series based model really knows and what such model should know.

1.1. Problem statement

1.2. Challenges

This template will automatically generate a cover page if you issue the `\makecover` command. There are two formats for the cover page: one with a page-filling ('bleeding') illustration, with the title(s) and author(s) in large ultrathin typeface, and the other where the illustration fills the lower half of the A4, whereas title(s), author(s) and additional text are set in the standard sans-serif font on a plain background with a color chosen by the user. The last option is selected by the optional key `split`: `\makecover[split]` yields a page with the illustration on the lower half. All illustrations are bleeding, in accordance with the TU Delft style.

Before generating the cover, you need to provide the information to put on it. This can be done with the following commands:

- `\title[Optional Color]{Title}`
  This command is used to provide the title of the document. The title title is also printed on the spine. If you use a title page (see below), this information will be used there as well. As the title, subtitle and author name are printed directly over the cover photo, it will often be necessary to adjust the print color

in order to have sufficient contrast between the text and the background. The optional color argument is used for this.

- `\title[Optional Color]{Subtitle}`
  This command is used to provide a subtitle for the document. If you use a title page (see below), this information will be used there as well. It possible to adjust the print color in order to have sufficient contrast between the text and the background – the optional color argument is used for this.

- `\author{J. Random Author}`
  This command specifies the author. The default color is `tudelft-white`, but this may be adjusted in the same way as the titles.

- `\affiliation{Technische Universiteit Delft}`
  The affiliation is the text printed vertically on the front cover. It can be the affiliation, such as the university or department name, or be used for the document type (*e.g.*, Master's thesis). The default color is again `tudelft-white`, adjustable through the `color` option.

- `\coverimage{cover.jpg}`
  With this command you can specify the filename of the cover image. The image is stretched to fill the full width of the front cover (including the spine if a back cover is present).

- `\covertext{Cover Text}`
  If a back cover is present, the cover text is printed on the back. Internally, this text box is created using the LaTeX `minipage` environment, so it supports line breaks.

- `\titleoffsetx{OffsetX},\titleoffsety{OffsetY}` If the cover page contains a page-filling picture (i.e., `split` is not specified with the `makecover` command, the best position of the title depends a lot on the picture chosen for it. The lower left corner of the minipage containing title, subtitle and author is specified by these two commands. The offsets are measured from the top left corner of the page.

- `\afiloffsetx{AfilX}, \afiloffsety{AfilY}` specifies the lower left corner of the text containing the affiliation, measured from the top left corner of the page.

In addition to `[split]`, the `\makecover` command accepts several additional options for customizing the layout of the cover. The most important of these is `back`. Supplying this option will generate a back cover as well as a front, including the spine. Since this requires a page size slightly larger than twice A4 (to make room for the spine), and LaTeX does not support different page sizes within the same document, it is wise to create a separate file for the cover. `cover.tex` contains an example. The recommended page size for the full cover can be set with

```
\geometry{papersize={1226bp,851bp}}
```

after the document class and before `\begin{document}`.

The other options `\makecover` accepts are

- `nospine`
  If a back cover is generated, the title will also be printed in a black box on the spine. However, for smaller documents the spine might not be wide enough. Specifying this option disables printing the title on the spine.

- `frontbottom`
  By default the black box on the front is situated above the blue box. Specifying this option will place the black box below the blue one.

- `spinewidth`
  If a back cover is present, this option can be used to set the width of the spine. The default is `spinewidth=1cm`.

- `frontboxwidth`, `frontboxheight`, `backboxwidth`, `backboxheight`
  As their names suggest, these options are used to set the width and height of the front (black) and back (blue) boxes. The default widths and heights are `4.375in` and `2.1875in`, respectively.

- `x`, `y`
  The blue and black boxes touch each other in a corner. The location of this corner can be set with these options. It is defined with respect to the top left corner of the front cover. The default values are `x=0.8125in` and `y=3in`.

- `margin`
  This option sets the margin between the borders of the boxes and their text. The default value is `12pt`.

For a thesis it is desirable to have a title page within the document, containing information like the thesis committee members. To give you greater flexibility over the layout of this page, it is not generated by a command like `\makecover`, but instead described in the file `title.tex`. Modify this file according to your needs. The example text is in English, but Dutch translations are provided in the comments. Note that for a thesis, the title page is subject to requirements which differ by faculty. Make sure to check these requirements before printing.

1.3. Contributions

Each chapter has its own file. For example, the LaTeX source of this chapter can be found in `chapter-1.tex`. A chapter starts with the command

```
\chapter{Chapter title}
```

This starts a new page, prints the chapter number and title and adds a link in the table of contents. If the title is very long, it may be desirable to use a shorter version in the page headers and the table of contents. This can be achieved by specifying the short title in brackets:

```
\chapter[Short title]{Very long title with many words which could not possibly
fit on one line}
```

Unnumbered chapters, such as the preface, can be created with `\chapter*{Chapter title}`. Such a chapter will not show up in the table of contents or in the page header. To create a table of contents entry anyway, add

```
\addcontentsline{toc}{chapter}{Chapter title}
```

after the `\chapter` command. To print the chapter title in the page header, add

```
\setheader{Chapter title}
```

Chapters are subdivided into sections, subsections, subsubsections, and, optionally, paragraphs and subparagraphs. All can have a title, but only sections and subsections are numbered. As with chapters, the numbering can be turned off by using `\section*{...}` instead of `\section{...}`, and similarly for the subsection.

1.4. Organization

2

Background and Related Work

3

Framework

3.1. modeling and training
3.2. Local Interpret Methods
3.3. Crowdsourcing
3.4. Data processing
3.5. Extract and characterize model unknowns

4

Dataset

4.1. LightDigit Dataset
4.2. Saliency Dataset

5

Algorithm

5.1. ViLiT(visible light transformer)
5.2. Temporal Saliency Rescaling

Bibliography

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[2] Agathe Balayn, Panagiotis Soilis, Christoph Lofi, Jie Yang, and Alessandro Bozzon. What do you mean? interpreting image classification with crowdsourced concept extraction and analysis. In *Proceedings of the Web Conference 2021*, pages 1937–1948, 2021.

[3] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916. PMLR, 2020.

[4] Hao Liu, Hanting Ye, Jie Yang, and Qing Wang. Through-screen visible light sensing empowered by embedded deep learning. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 478–484, 2021.

[5] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.

[6] Sisi Ma and Roshan Tourani. Predictive and causal implications of using shapley value for model interpretation. In *Proceedings of the 2020 KDD Workshop on Causal Discovery*, pages 23–38. PMLR, 2020.

[7] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431. Springer, 2020.