



TECHNISCHE  
UNIVERSITÄT  
WIEN  
Vienna | Austria

# Self-Supervised Learning for Cyber Security Applications

**Optional Subtitle of the Thesis**

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Embedded Systems**

by

**Jonas Ferdigg, BSc**

Registration Number 01226597

to the Faculty of Electrical Engineering and Information Technology  
at the TU Wien

Advisor: Univ. Prof. Dipl.-Ing. Dr.-Ing. Tanja Zseby

Assistance: Univ.Ass. Dott.mag. Maximilian Bachl

Vienna, 1<sup>st</sup> January, 2001



# Erklärung zur Verfassung der Arbeit

Jonas Ferdigg, BSc

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct der Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, 1. Jänner 2001



# Acknowledgements

Enter your text here.



# Kurzfassung

---

Ihr Text hier.





# Abstract



# Contents

<b>Kurzfassung</b>	<b>vii</b>
<b>Abstract</b>	<b>ix</b>
<b>Contents</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Research Questions . . . . .	3
1.3 Approach . . . . .	3
1.4 Contribution . . . . .	3
1.5 Structure . . . . .	3
1.6 Support (optional) . . . . .	3
<b>2 Background</b>	<b>5</b>
<b>3 State of the art</b>	<b>7</b>
<b>4 Methodology</b>	<b>9</b>
<b>5 Experiments</b>	<b>11</b>
<b>6 Results</b>	<b>13</b>
<b>7 Discussion</b>	<b>15</b>
<b>8 Conclusion</b>	<b>17</b>
<b>A Rules for writing the Thesis</b>	<b>19</b>
A.1 General Rules . . . . .	19
A.2 Writing the Thesis . . . . .	19
A.3 Tools and Infrastructure . . . . .	21
A.4 Communication . . . . .	21
A.5 Reproducibility . . . . .	22
A.6 Publishing Papers . . . . .	22
	xi

A.7	Open Issues . . . . .	22
<b>B</b>	<b>Introduction to L<sup>A</sup>T<sub>E</sub>X</b>	<b>23</b>
B.1	Installation . . . . .	23
B.2	Editors . . . . .	23
B.3	Compilation . . . . .	24
B.4	Basic Functionality . . . . .	25
B.5	Bibliography . . . . .	26
B.6	Table of Contents . . . . .	27
B.7	Acronyms / Glossary / Index . . . . .	27
B.8	Tips . . . . .	27
B.9	Resources . . . . .	28
	<b>List of Figures</b>	<b>31</b>
	<b>List of Tables</b>	<b>33</b>
	<b>List of Algorithms</b>	<b>35</b>
	<b>Bibliography</b>	<b>37</b>

# Introduction

## 1.1 Motivation

Give a brief overview of the motivation for the thesis.

Provide the Problem Statement:

- Why is your topic an important topic?
- Why is it not yet solved?
- Why has nobody solved it so far (was it not relevant or not needed so far? was it too difficult so far?)
- Why is it not easy to be solved? (why does it need research and a skilled person to solve it?)
- Why do you think you can and should solve it now? (e.g., because now it became relevant, or we now have faster computers to make it possible to solve it, or you have a unique idea to solve the problem that no one has tried so far)

With the progressing digitalization of evermore aspects of society, cyber security will always be a relevant issue as no system will ever be fully secure. Preventing possible cyber attacks by developing more robust systems is one way to mitigate the issue, the other is preventing already existing faults from being exploited as not every vulnerability can be patched easily as it is the case with e.g. DoS and brute force attacks. To stop such attacks it is necessary to identify them within the vast flow of ordinary network traffic which gives rise to the need of Intrusion Detection Systems (IDS). State-of-the-art IDSs apply two methods to

With the TODO format you can mark open issues or comments during editing. It will automatically generate a TODO List at the end of the document

insert reference to state of the art ids

give examples for IDSs lacking accuracy

give examples for NN based IDSs

give examples of self supervised machine learning

detect occurring attacks: Signature-based detection and statistical anomaly-based detection. Signature-based detection looks for known patterns or signatures within packets and data streams to identify incoming attacks . Statistical anomaly-based detection focuses on differentiating between normal and abnormal behavior in the system and raises an alert if the latter is identified. The problem with signature-based detection is that unknown attacks are ignored and anomaly-based detection is still not sufficiently accurate and prone to false positives . The rise of Machine Learning (ML) gave opportunity to use the mighty pattern recognition capabilities of Neural Networks (NN) for intrusion detection. As ML is a rapidly developing field its steady improvement fueled the advance of NN based IDSs which start to show promising results . NNs however are still mostly trained in a supervised fashion, namely by providing labeled examples of cyber attacks for the NN to learn from. This again poses the problem, that only known attacks can be identified, but new attacks that are sufficiently similar to old attacks can also be identified, which is not the case with mere signature-based detection. As with every form of supervised training on NNs, labeled data is harder to come by while unlabeled data is often abundant and certainly so for network traffic data. For this reason, self-supervised training/pretraining is seeing increased use in the realm of ML , as unlabeled data can be used to boost the performance without the need for expensive labeled data. One of the most noteworthy examples of the effectiveness of self-supervised pre-training for Neural Networks in the realm of Natural Language Processing (NLP) is Bidirectional Encoder Representations from Transformers (BERT) [DCLT18] developed by Jacob Devlin *et al* from Google AI Language. BERT is based on the state-of-the-art Transformer architecture [VSP<sup>+</sup>17] and uses a series of proxy tasks like word masking and next sentence prediction to teach the network about syntax and grammar in a self-supervised fashion. The pre-trained network can then be fine-tuned for more specific tasks like question answering or text classification. Analogous, it would be highly beneficial if these or similar pre-training mechanisms could be used to bolster performance of ML based IDSs by improving the classification of network flows, at the most basic level, into cyber attack vs. no cyber attack.

As the technologies mentioned above are fairly recent (Transformers Dec 2017, BERT May 2019) and the design space for solutions in the context of ML for cyber security is substantial, there has not yet been sufficient inquiry into the possibilities of these new methods when applied to the problems posed by Intrusion Detection and cyber attack classification. NN performance also improves with the steadily increasing capabilities of modern Graphics Processing Units which makes this a promising concept which can be improved upon by future more powerful hardware. As I am both versed in the domain of Machine Learning and Cyber Security, i find myself able to contribute to this narrow field of research by writing this thesis.

## 1.2 Research Questions

Here state the exact research questions that you try to answer with the thesis

Good research questions are specific and measurable. For example if you want to show that an anomaly detection method is better than another one, do not just say "better" but rather provide details of what you mean by better (e.g., higher speed, lower computational complexity, better detection performance, etc)

- R1:
- R2:
- R3:

## 1.3 Approach

Here describe briefly what is your approach to solve the problem or to answer the research questions. What methodology did you choose and why (briefly)? e.g., theoretical work, simulations, experiments,...

## 1.4 Contribution

Here provide a list of the contributions of your work.

Suggestion (especially for dissertations): provide a table with research questions, methods used to answer each, and major findings and the section in which to find details.

## 1.5 Structure

Describe the structure of the thesis in 1-2 paragraphs.

In section XXX I provide state of the art...

## 1.6 Support (optional)

In case your research was supported by a project, you can here mention the project and its objectives





# CHAPTER 2

## Background

Provide some background information about your work. Here also introduce the terminology and notation used.

In addition: Abbreviations and mathematical notation should be put in a list in the beginning of the thesis

### 2.0.1 Terminology



## State of the art

*Notice of adoption from previous publications in section 1*

*Parts of the contents of this chapter have been published in the following papers:*

*[P1]*

*[P2]*

*Explanation text, on what parts were adopted from previous publications:*

*e.g. "The statistical anomaly detection algorithm published in the above mentioned papers and described in this Chapter is based on the work done in [29]."*

Here provide an overview of the related state of art. Look for papers that are closest to the research you are doing Suggestion: make a table with the related papers and compare them wrt to different criteria, for instance

- Findings: What do they claim (main findings)
- Data: What data set they are using
- Methods: Which methods did they use?
- Reproducibility: Is it possible to reproduce the results? (e.g., is the data available? are all parameter settings provided? Is source code provided?)
- Relevance (How relevant is it for your work)

In the last paragraph explain how your work differs from the existing works.



# CHAPTER 4

## Methodology

Here describe the methodology you use and why you decided to use it. e.g., theoretical considerations, simulations, experiments, measurements, testbeds, emulations, etc. What concepts are used.

Also explain which metrics you use to measure success or failure (e.g., detection performance with accuracy, recall, precision, f1 score, RocAUC, etc.)

Provide a figure (see example figure 4.1) to describe the processing steps

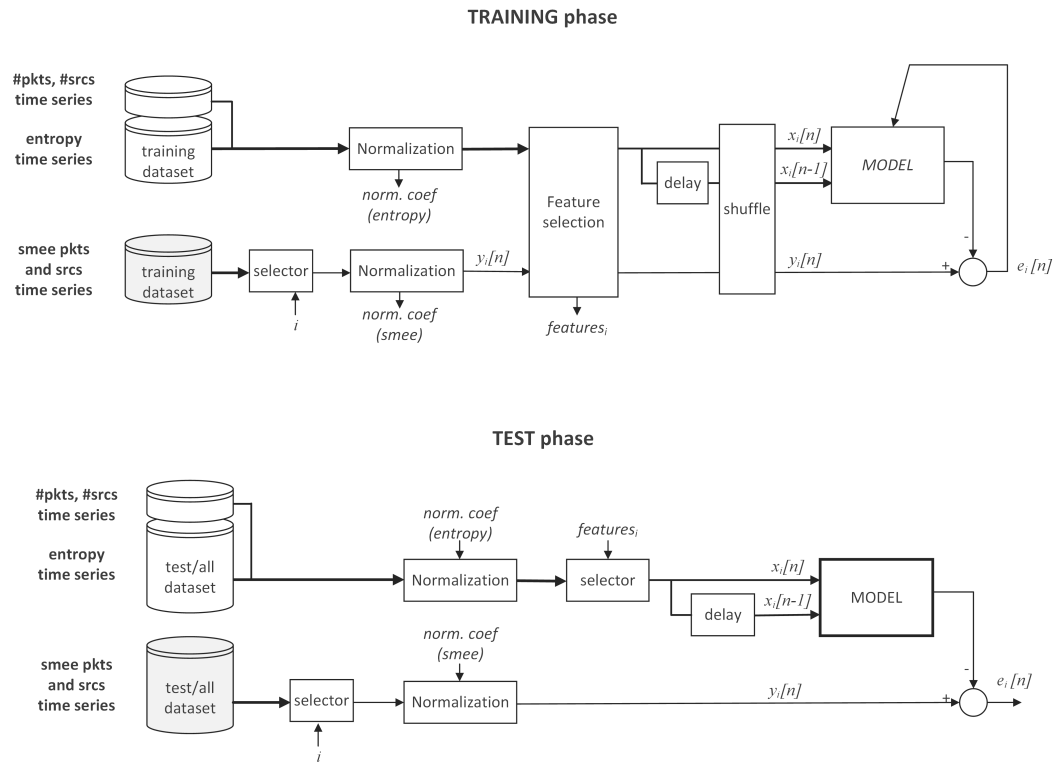


Figure 4.1: Describe in the caption exactly what can be seen in the figure

# CHAPTER 5

## Experiments

Describe all details of the experiments in a way that the reader is able to reproduce your experiments. Provide an overview table (or multiple tables) with all experiments, data sets and all parameters used. Give the different experiment runs a unique name and refer to this name in the results section.

Keep interim data from your experiments (e.g. predicted labels per instance, confusion matrices, etc.) in case you need to come back to the data to calculate further metrics.





# CHAPTER 6

## Results

Show the results. Link them to the experiment (e.g. by providing a unique naming per experiment). Then first describe the results (what can be seen, is there anything unusual or all as expected) and then interpret them (do you have an explanation why the results look like this? do you have ideas why it did not look as expected?)



# CHAPTER 7

## Discussion

Discuss any open issues and give a critical reflection of your work. E.g., what could be problems to deploy your method or do you have an idea how your findings could be generalized or what could be a hindrance for generalization?

Also discuss strange things you observed or results you could not completely explain.



## CHAPTER 8

# Conclusion

Conclude your work. Stress again what was the contribution. Provide an outlook what could be further improvements and what could future research do to continue your work.



# Rules for writing the Thesis

## A.1 General Rules

- Code of Conduct: You need to understand and sign the TU Code of Conduct before working on a thesis at TU. You can find it at [https://www.tuwien.at/fileadmin/Assets/dienstleister/Datenschutz\\_und\\_Dokumentenmanagement/Code\\_of\\_Conduct\\_fuer\\_wissenschaftliches\\_Arbeiten.pdf](https://www.tuwien.at/fileadmin/Assets/dienstleister/Datenschutz_und_Dokumentenmanagement/Code_of_Conduct_fuer_wissenschaftliches_Arbeiten.pdf)
- Time Planning: Plan your thesis realistically. Check how much time you need for studies and work and other obligations to estimate how much time you can spend per week on your thesis. Especially if you have to learn new things (theoretical knowledge in a new field, a new tool, a new programming language), plan sufficient time for this. Keep in mind that always unforeseen problems can occur. So plan some buffer time.
- External Deadlines: Make all deadlines clear before you start the thesis. E.g. if you have any time constraints wrt. projects, visa applications, planned employment or any other time restrictions in your studies, let the supervisor know this before you start working on the thesis. Last minute request will not be accepted.
- 
- 

## A.2 Writing the Thesis

- Use the CN group latex Master Thesis template
- Continuously document what you are doing

- Make notes about papers you read
- Document all experiment details. Also if experiments are not successful it is important to document what you did and which errors occurred
- Document your software in a way that others can continue to understand and modify/extend the software
- Use US english
- Consider to write a paper from your results

### A.2.1 Tenses

- Use present tense for state of art
- 

**TZ TODO: add rules and references about tenses**

### A.2.2 References, Copyright and Citations

- citations need to be clearly marked (see code of conduct)
- no re-phrasing
- Ideally use no figures copied from somewhere else. If figure are copied, a) the copyright must allow use it and b) they have to be correctly cited
- You may use sherpa to identify the copyright rules for particular Journal. <http://www.sherpa.ac.uk/romeo/index.php?la=en&fIDnum=|&mode=simple>
- Some useful definitions and rules for plagiarism and self-plagiarism can be found at <https://www.fsdr.at/plagiarism>
- Rules how to correctly cite a creative commons figure see [https://commons.wikimedia.org/wiki/Commons:Reusing\\_content\\_outside\\_Wikimedia](https://commons.wikimedia.org/wiki/Commons:Reusing_content_outside_Wikimedia)
- References: Use books or scientific papers as reference instead of web pages or blog entries
- If you have to cite a web page you have to provide the date when you last accessed the page , last accessed at YYYY-MM-DD

**TZ TODO: add example for creative commons reference**

**TZ TODO: add references to code of conduct and plagiarism rules**



### A.2.3 Latex Tools

**TZ TODO: Add links**

## A.3 Tools and Infrastructure

The following tools are useful:

- thesis template
- zotero ([zotero.org](https://www.zotero.org)): Tool for collecting papers and sharing papers with others (creating a zotero group)
- SVN or git for joint paper editing
- Overleaf for short term joint editing of latex files

Open Issues

- Getting data sets from CN group
- Getting access to CN infrastructure (compute cluster, GPU, storage)
- Access to NTARC?
- provide a template for describing experiments

## A.4 Communication

The first rule is to stay in contact and inform the supervisor(s) about your progress, questions and difficulties.

So always ask:

- If anything is not clear about what you should do
- If you do not understand something (e.g., a paper, an equation, a statement)
- If you have problems with software, programming, etc.
- If you don't know which papers are relevant and which not
- If you have a new idea or want to take a different path.

Further rules:

- Friday updates: send a brief update to your supervisor(s) every Friday. You can include any ideas, questions or difficulties that you had during the week. If you did not make any progress in the week just send an email saying that you did not make progress.
- Use an SVN or git repository to store the latest version of your document
- Use meaningful file names: Example: YYYY-MM-DD-YourLastName-DocumentName-version
- Send an email to supervisors(s) if a new version to be reviewed is in the SVN
- clearly mark all changes in the document that you made compared to the last version. Show how you addressed comments.

### A.5 Reproducibility

### A.6 Publishing Papers

#### A.6.1 Finding suitable Conferences

Top Conferences and Journals

Conferences and Journal Rankings

#### A.6.2 Using arxiv

### A.7 Open Issues

- put change marking method in template

# Introduction to L<sup>A</sup>T<sub>E</sub>X

Since L<sup>A</sup>T<sub>E</sub>X is widely used in academia and industry, there exists a plethora of freely accessible introductions to the language. Reading through the guide at <https://en.wikibooks.org/wiki/LaTeX> serves as a comprehensive overview for most of the functionality and is highly recommended before starting with a thesis in L<sup>A</sup>T<sub>E</sub>X.

## B.1 Installation

A full L<sup>A</sup>T<sub>E</sub>X distribution consists not only of the binaries that convert the source files to the typeset documents, but also of a wide range of packages and their documentation. Depending on the operating system, different implementations are available as shown in Table B.1. **Due to the large amount of packages that are in everyday use and due to their high interdependence, it is paramount to keep the installed distribution up to date.** Otherwise, obscure errors and tedious debugging ensue.

## B.2 Editors

A multitude of T<sub>E</sub>X editors are available differing in their editing models, their supported operating systems and their feature sets. A comprehensive overview of editors can be

Distribution	Unix	Windows	MacOS
TeX Live	<b>yes</b>	yes	(yes)
MacTeX	no	no	<b>yes</b>
MikTeX	(yes)	<b>yes</b>	yes

Table B.1: T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X distributions for different operating systems. Recommended choice in **bold**.

Description	
1	Scan for refs, toc/lof/lot/loa items and cites
2	Build the bibliography
3	Link refs and build the toc/lof/lot/loa
4	Link the bibliography
5	Build the glossary
6	Build the acronyms
7	Build the index
8	Link the glossary, acronyms, and the index
9	Link the bookmarks
Command	
1	<code>pdflatex.exe example</code>
2	<code>bibtex.exe example</code>
3	<code>pdflatex.exe example</code>
4	<code>pdflatex.exe example</code>
5	<code>makeindex.exe -t example.glg -s example.ist</code> <code>-o example.gls example.glo</code>
6	<code>makeindex.exe -t example.alg -s example.ist</code> <code>-o example.acr example.acn</code>
7	<code>makeindex.exe -t example.ilg -o example.ind example.idx</code>
8	<code>pdflatex.exe example</code>
9	<code>pdflatex.exe example</code>

Table B.2: Compilation steps for this document. The following abbreviations were used: table of contents (toc), list of figures (lof), list of tables (lot), list of algorithms (loa).

found at the Wikipedia page [https://en.wikipedia.org/wiki/Comparison\\_of\\_TeX\\_editors](https://en.wikipedia.org/wiki/Comparison_of_TeX_editors). TeXstudio (<http://texstudio.sourceforge.net/>) is recommended. Most editors support a synchronization of the generated document and the L<sup>A</sup>T<sub>E</sub>X source by Ctrl clicking either on the source document or the generated document.

### B.3 Compilation

Modern editors usually provide the compilation programs to generate Portable Document Format (PDF) documents and for most L<sup>A</sup>T<sub>E</sub>X source files, this is sufficient. More advanced L<sup>A</sup>T<sub>E</sub>X functionality, such as glossaries and bibliographies, needs additional compilation steps, however. It is also possible that errors in the compilation process invalidate intermediate files and force subsequent compilation runs to fail. It is advisable to delete intermediate files (`.aux`, `.bbl`, etc.), if errors occur and persist. All files that are not generated by the user are automatically regenerated. To compile the current document, the steps as shown in Table B.2 have to be taken.

## B.4 Basic Functionality

In this section, various examples are given of the fundamental building blocks used in a thesis. Many  $\text{\LaTeX}$  commands have a rich set of options that can be supplied as optional arguments. The documentation of each command should be consulted to get an impression of the full spectrum of its functionality.

### B.4.1 Floats

Two main categories of page elements can be differentiated in the usual  $\text{\LaTeX}$  workflow: *(i)* the main stream of text and *(ii)* floating containers that are positioned at convenient positions throughout the document. In most cases, tables, plots, and images are put into such containers since they are usually positioned at the top or bottom of pages. These are realized by the two environments `figure` and `table`, which also provide functionality for cross-referencing (see Table B.3 and Figure B.1) and the generation of corresponding entries in the list of figures and the list of tables. Note that these environments solely act as containers and can be assigned arbitrary content.

### B.4.2 Tables

A table in  $\text{\LaTeX}$  is created by using a `tabular` environment or any of its extensions, e.g., `tabularx`. The commands `\multirow` and `\multicolumn` allow table elements to span multiple rows and columns.

Position		
Group	Abbrev	Name
Goalkeeper	GK	Paul Robinson
Defenders	LB	Lucas Radebe
	DC	Michael Duburrry
	DC	Dominic Matteo
	RB	Didier Domi
Midfielders	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Forward	FW	Jamie McMaster
Strikers	ST	Alan Smith
	ST	Mark Viduka

Table B.3: Adapted example from the  $\text{\LaTeX}$ guide at <https://en.wikibooks.org/wiki/LaTeX/Tables>. This example uses rules specific to the `booktabs` package and employs the multi-row functionality of the `multirow` package.

### B.4.3 Images

An image is added to a document via the `\includegraphics` command as shown in Figure B.1. The `\subcaption` command can be used to reference subfigures, such as Figure B.1a and B.1b.

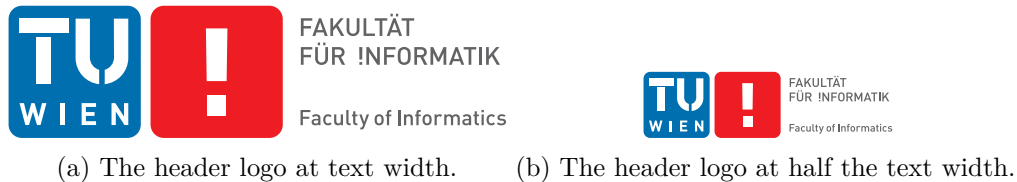


Figure B.1: The header logo at different sizes.

### B.4.4 Mathematical Expressions

One of the original motivation to create the T<sub>E</sub>X system was the need for mathematical typesetting. To this day, L<sup>A</sup>T<sub>E</sub>X is the preferred system to write math-heavy documents and a wide variety of functions aids the author in this task. A mathematical expression can be inserted inline as  $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$  outside of the text stream as

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

or as numbered equation with

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}. \tag{B.1}$$

### B.4.5 Pseudo Code

The presentation of algorithms can be achieved with various packages; the most popular are `algorithmic`, `algorithm2e`, `algorithmicx`, or `algpseudocode`. An overview is given at <https://tex.stackexchange.com/questions/229355>. An example of the use of the `algorithm2e` package is given with Algorithm B.1.

## B.5 Bibliography

The referencing of prior work is a fundamental requirement of academic writing and well supported by L<sup>A</sup>T<sub>E</sub>X. The B<sub>I</sub>B<sub>T</sub>E<sub>X</sub> reference management software is the most commonly used system for this purpose. Using the `\cite` command, it is possible to reference entries in a `.bib` file out of the text stream, e.g., as [Tur36]. The generation of the formatted bibliography needs a separate execution of `bibtex.exe` (see Table B.2).

---

**Algorithm B.1:** Gauss-Seidel

---

**Input:** A scalar  $\epsilon$ , a matrix  $\mathbf{A} = (a_{ij})$ , a vector  $\vec{b}$ , and an initial vector  $\vec{x}^{(0)}$

**Output:**  $\vec{x}^{(n)}$  with  $\mathbf{A}\vec{x}^{(n)} \approx \vec{b}$

```
1 for  $k \leftarrow 1$  to maximum iterations do
2   for  $i \leftarrow 1$  to  $n$  do
3      $x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j < i} a_{ij} x_j^{(k)} - \sum_{j > i} a_{ij} x_j^{(k-1)} \right);$ 
4   end
5   if  $|\vec{x}^{(k)} - \vec{x}^{(k-1)}| < \epsilon$  then
6     break for;
7   end
8 end
9 return  $\vec{x}^{(k)}$ ;
```

---

## B.6 Table of Contents

The table of contents is automatically built by successive runs of the compilation, e.g., of `pdflatex.exe`. The command `\setsecnumdepth` allows the specification of the depth of the table of contents and additional entries can be added to the table of contents using `\addcontentsline`. The starred versions of the sectioning commands, i.e., `\chapter*`, `\section*`, etc., remove the corresponding entry from the table of contents.

## B.7 Acronyms / Glossary / Index

The list of acronyms, the glossary, and the index need to be built with a separate execution of `makeindex` (see Table B.2). Acronyms have to be specified with `\newacronym` while glossary entries use `\newglossaryentry`. Both are then used in the document content with one of the variants of `\gls`, such as `\Gls`, `\glspl`, or `\Glspl`. Index items are simply generated by placing `\index{<entry>}` next to all the words that correspond to the index entry `<entry>`. Note that many enhancements exist for these functionalities and the documentation of the `makeindex` and the `glossaries` packages should be consulted.

## B.8 Tips

Since  $\text{\TeX}$  and its successors do not employ a What You See Is What You Get (WYSIWYG) editing scheme, several guidelines improve the readability of the source content:

- Each sentence in the source text should start with a new line. This helps not only the user navigation through the text, but also enables revision control systems

(e.g. Subversion (SVN), Git) to show the exact changes authored by different users. Paragraphs are separated by one (or more) empty lines.

- Environments, which are defined by a matching pair of `\begin{name}` and `\end{name}`, can be indented by whitespace to show their hierarchical structure.
- In most cases, the explicit use of whitespace (e.g. by adding `\hspace{4em}` or `\vspace{1.5cm}`) violates typographic guidelines and rules. Explicit formatting should only be employed as a last resort and, most likely, better ways to achieve the desired layout can be found by a quick web search.
- The use of bold or italic text is generally not supported by typographic considerations and the semantically meaningful `\emph{...}` should be used.

The predominant application of the L<sup>A</sup>T<sub>E</sub>X system is the generation of PDF files via the PDFL<sup>A</sup>T<sub>E</sub>X binaries. In the current version of PDFL<sup>A</sup>T<sub>E</sub>X, it is possible that absolute file paths and user account names are embedded in the final PDF document. While this poses only a minor security issue for all documents, it is highly problematic for double blind reviews. The process shown in Table B.4 can be employed to strip all private information from the final PDF document.

	Command
1	Rename the PDF document <code>final.pdf</code> to <code>final.ps</code> .
2	Execute the following command: <pre>ps2pdf -dPDFSETTINGS#/prepress ^       -dCompatibilityLevel#1.4 ^       -dAutoFilterColorImages#false ^       -dAutoFilterGrayImages#false ^       -dColorImageFilter#/FlateEncode ^       -dGrayImageFilter#/FlateEncode ^       -dMonoImageFilter#/FlateEncode ^       -dDownsampleColorImages#false ^       -dDownsampleGrayImages#false ^       final.ps final.pdf</pre>
	On Unix-based systems, replace <code>#</code> with <code>=</code> and <code>^</code> with <code>\</code> .

Table B.4: Anonymization of PDF documents.

## B.9 Resources

### B.9.1 Useful Links

In the following, a listing of useful web resources is given.



**<https://en.wikibooks.org/wiki/LaTeX>** An extensive wiki-based guide to  $\text{\LaTeX}$ .

**<http://www.tex.ac.uk/faq>** A (huge) set of Frequently Asked Questions (FAQ) about  $\text{\TeX}$  and  $\text{\LaTeX}$ .

**<https://tex.stackexchange.com/>** The definitive user forum for non-trivial  $\text{\LaTeX}$ -related questions and answers.

### B.9.2 Comprehensive TeX Archive Network (CTAN)

The CTAN is the official repository for all  $\text{\TeX}$  related material. It can be accessed via <https://www.ctan.org/> and hosts (among other things) a huge variety of packages that provide extended functionality for  $\text{\TeX}$  and its successors. Note that most packages contain PDF documentation that can be directly accessed via CTAN.

In the following, a short, non-exhaustive list of relevant CTAN-hosted packages is given together with their relative path.

**algorithm2e** Functionality for writing pseudo code.

**amsmath** Enhanced functionality for typesetting mathematical expressions.

**amssymb** Provides a multitude of mathematical symbols.

**booktabs** Improved typesetting of tables.

**enumitem** Control over the layout of lists (`itemize`, `enumerate`, `description`).

**fontenc** Determines font encoding of the output.

**glossaries** Create glossaries and list of acronyms.

**graphicx** Insert images into the document.

**inputenc** Determines encoding of the input.

**l2tabu** A description of bad practices when using  $\text{\LaTeX}$ .

**mathtools** Further extension of mathematical typesetting.

**memoir** The document class on upon which the `vutinfth` document class is based.

**multirow** Allows table elements to span several rows.

**pgfplots** Function plot drawings.

**pgf/TikZ** Creating graphics inside  $\text{\LaTeX}$  documents.

**subcaption** Allows the use of subfigures and enables their referencing.

**symbols/comprehensive** A listing of around 5000 symbols that can be used with  $\text{\LaTeX}$ .

**voss-mathmode** A comprehensive overview of typesetting mathematics in  $\text{\LaTeX}$ .

**xcolor** Allows the definition and use of colors.



# List of Figures

4.1	Describe in the caption exactly what can be seen in the figure . . . . .	10
B.1	The header logo at different sizes. . . . .	26



# List of Tables

B.1	T <sub>E</sub> X/L <sup>A</sup> T <sub>E</sub> X distributions for different operating systems. Recommended choice in <b>bold</b> . . . . .	23
B.2	Compilation steps for this document. The following abbreviations were used: table of contents (toc), list of figures (lof), list of tables (lot), list of algorithms (loa). . . . .	24
B.3	Adapted example from the L <sup>A</sup> T <sub>E</sub> Xguide at <a href="https://en.wikibooks.org/wiki/LaTeX/Tables">https://en.wikibooks.org/wiki/LaTeX/Tables</a> . This example uses rules specific to the booktabs package and employs the multi-row functionality of the multirow package. . . . .	25
B.4	Anonymization of PDF documents. . . . .	28



# List of Algorithms

B.1	Gauss-Seidel . . . . .	27
-----	------------------------	----





# Bibliography

- [DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
- [Tur36] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58:345–363, 1936.
- [VSP<sup>+</sup>17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.