

# Paper on Privacy-Preserving Machine Learning: Draft for Salzburg and FWF-Proposal

Bernd Malle, Peter Kieseberg, Andreas Holzinger

Holzinger Group HCI-KDD  
Institute for Medical Informatics, Statistics & Documentation  
Medical University Graz, Austria  
[a.holzinger@hci-kdd.org](mailto:a.holzinger@hci-kdd.org)

**Abstract.** This is a *general purpose template* for all types of Springer LNCS and LNAI submissions following the `llncs.cls` styleset. It includes hints for  $\text{\LaTeX}$ typesetting of theory, algorithms, tables, formulas etc. s based on the preparation of a research article for submission to the *Transactions on Rough Sets* (TRS), which is a journal subline of the *Lecture Notes in Computer Science* (LNCS) published by Springer<sup>1</sup>. However, this template can be used for any LNCS or LNAI contribution using the `llncs` style. This document illustrates parts of a manuscript (*e.g.*, abstract, keywords, introduction, organization of sections, notation, algorithms, theory, formulas, figures, tables, conclusion, acknowledgements and references) of a typical Springer LNCS/LNAI article.  $\text{\LaTeX}$  and the Springer *llncs* document class are to be used to prepare an article. Details on how to obtain and use  $\text{\LaTeX}$  in principal are also outlined. BTW: A very interesting idea is to prepare a graphical abstract, which outlines the content of a paper within a single image. Moreover, a good idea is to boil down the three (3) most important issues on how this paper contributes to the international scientific community. Finally, consider that what other researches can use - will be referenced, thus will increase your impact. This template has zero (0) errors, zero warnings, and zero (0) bad boxes.

**Keywords:** Springer Article, guidelines,  $\text{\LaTeX}$ , LNCS, LNAI.

## 1 Introduction and Motivation for Research

At the beginning provide a motivation on why your paper is interesting. Example: ML is very broad and deals with the problem of extracting features from data to solve predictive tasks, including decision support, forecasting, ranking, classifying (*e.g.*, in cancer diagnosis), detecting anomalies (*e.g.*, virus mutations) or sentiment analysis [44]. The challenge is to discover relevant *structural* patterns and/or *temporal* patterns (“knowledge”) in such data, which are often hidden and not accessible to the human expert. The problem is that a majority of the

---

<sup>1</sup> See <http://www.springer.com/west/home/computer/lncs?SGWID=4-164-6-99627-0>

data sets in the biomedical domain are weakly-structured and non-standardized [45], and most data is in dimensions much higher than 3, and despite human experts are excellent in pattern recognition for dimensions  $\leq 3$ , such data make manual analysis often impossible.

Most colleagues from the ML community are concentrating on *automatic* machine learning (aML), with the grand goal of bringing humans-out-of-the-loop, and a best practice real-world example can be found in autonomous vehicles.

However, biomedical data sets are full of uncertainty, incompleteness etc. [46], they can contain missing data, noisy data, dirty data, unwanted data, and most of all, some problems in the medical domain are hard, which makes the application of fully automated approaches difficult or even impossible, or at least the quality of results from automatic approaches might be questionable. Moreover, the complexity of sophisticated machine learning algorithms has detained non-experts from the application of such solutions. Consequently, the integration of the knowledge of a domain expert can sometimes be indispensable and the interaction of a domain expert with the data would greatly enhance the knowledge discovery process pipeline. Hence, *interactive* machine learning (iML) puts the “human-in-the-loop” to enable what neither a human nor a computer could do on their own. This idea is supported by a synergistic combination of methodologies of two areas that offer ideal conditions towards unraveling such problems: Human-Computer Interaction (HCI) and Knowledge Discovery/Data Mining (KDD), with the goal of supporting human intelligence with machine intelligence to discover novel, previously unknown insights into data (HCI-KDD approach [47]).

**We define iML-approaches as algorithms that can interact with both computational agents and human agents \*) and can optimize their learning behaviour through these interactions. [48]**

\*) In Active Learning such agents are referred to as so-called “oracles”.

This template is based on the TRS: The *Transactions on Rough Sets* (TRS journal is an outgrowth of the pioneering work on rough sets by Zdzisław Pawlak (see, e.g., [21–26]) and many others [31] during the past three decades.

This Styleguide for preparing an article for submission to the TRS has the following organization: The  $\text{\LaTeX}$  document class to be used by every author is specified in Sect. 3. In addition, information about  $\text{\LaTeX}$  as well as information about public domain software for  $\text{\LaTeX}$  users is also given in Sect. 3. A very detailed set of style guidelines that must be used in preparing an article for the TRS can be found in Sect. [style]. Notice that this document illustrates how each of the article components described in the Style Guideline can be performed in  $\text{\LaTeX}$ . In effect, this document can be used as a template for a TRS article. The method of submission of an article to the TRS is given in Sect. [method]. Finally, the specification for the concluding section of a TRS article is given in Sect. 4.

## 2 Glossary and Key Terms

**Note: This is only for use when producing a Springer LNCS SOTA State-of-the-Art-Analysis paper**

*Open data* is any error, anomaly and/or undesired modification in the perception or representation of information from data.

*Gaussian process* is a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions.. Carl Edward Rasmussen

etc. etc. etc. define your key terms here

etc. etc. etc. define your key terms here

etc. etc. etc. define your key terms here

etc. etc. etc. define your key terms here

## 3 L<sup>A</sup>T<sub>E</sub>X Requirements

Prepare your TRS paper using **L<sup>A</sup>T<sub>E</sub>X** and the Springer-Verlag document class llncs.cls (see [35]). There are a number of books that are helpful in showing how to use L<sup>A</sup>T<sub>E</sub>X (see, *e.g.*, [7, 5, 17]). A tutorial to help you get started in preparing a L<sup>A</sup>T<sub>E</sub>X document is available at [13]. To find the latest version of L<sup>A</sup>T<sub>E</sub>X, try [14]. Articles written in Microsoft <sup>®</sup> Word will not be accepted.

MiKTeX (pronounced mik-tech) is an up-to-date implementation of T<sub>E</sub>X [16], which is available for different platforms<sup>2</sup>. For example, T<sub>E</sub>X and related programs is available for Windows on a PC (all current variants), MacOS X, Linux, and Unix. T<sub>E</sub>X is a typesetting system invented by D.E. Knuth [10], who has written that L<sup>A</sup>T<sub>E</sub>X is “intended for the creation of beautiful books—and especially for books that contain a lot mathematics”. More information about T<sub>E</sub>X (its history, novel aspects of T<sub>E</sub>X, examples, software, notes, references as well as external links) can be found at [38]. The MiKTeX system is public domain software that is available from [15]. The **graphicx**, **algorithm2e**, and **subfigure** packages used in this document, are among the files that can be downloaded from the web (see, *e.g.*, [1, 6, 36]).

It is also important to have access to a L<sup>A</sup>T<sub>E</sub>X editor such as TeXnicCenter, which is an Integrated Development Environment for writing TeX/LaTeX-documents on Microsoft Windows <sup>®</sup> platforms[40]. TeXnicCenter provides a lot of enhanced features like structure parsing, project management, syntax highlighting, and so on. TeXnicCenter is also public domain software available from [39].

## ~~Symbols in~~ L<sup>A</sup>T<sub>E</sub>X

The abstract for your article should briefly describe the problem(s) solved in your article. Include in your description, a brief indication of the context for as the corresponding L<sup>A</sup>T<sub>E</sub>X commands used to produce the symbols is available from [12]. Tables 1-3 present some of the more commonly used symbols in L<sup>A</sup>T<sub>E</sub>X.  
See <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=TeXsystems>.

**Table 1.** Lowercase Greek Letters

$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$o$	<code>\o</code>	$\tau$	<code>\tau</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$\upsilon$	<code>\upsilon</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\phi$	<code>\phi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\varphi$	<code>\varphi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\chi$	<code>\chi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\psi$	<code>\psi</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>	$\omega$	<code>\omega</code>
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>				

**Table 2.** Uppercase Greek Letters

$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		

**Table 3.** Relation Symbols

$\in$	<code>\in</code>	$\notin$	<code>\notin</code>	$\cap$	<code>\cap</code>	$\cup$	<code>\cup</code>
$\approx$	<code>\approx</code>	$\rightarrow$	<code>\rightarrow</code>	$\prec$	<code>\prec</code>	$\subset$	<code>\subset</code>
$\asymp$	<code>\asymp</code>	$\Join$	<code>\Join</code>	$\preceq$	<code>\preceq</code>	$\subseteq$	<code>\subseteq</code>
$\bowtie$	<code>\bowtie</code>	$\leq$	<code>\leq</code>	$\propto$	<code>\propto</code>	$\succ$	<code>\succ</code>
$\cong$	<code>\cong</code>	$\ll$	<code>\ll</code>	$\sim$	<code>\sim</code>	$\succeq$	<code>\succeq</code>
$\dashv$	<code>\dashv</code>	$\mid$	<code>\mid</code>	$\simeq$	<code>\simeq</code>	$\supset$	<code>\supset</code>
$\doteq$	<code>\doteq</code>	$\models$	<code>\models</code>	$\smile$	<code>\smile</code>	$\supseteq$	<code>\supseteq</code>
$\equiv$	<code>\equiv</code>	$\neq$	<code>\neq</code>	$\vdash$	<code>\vdash</code>	$\forall$	<code>\forall</code>
$\gg$	<code>\gg</code>	$\ni$	<code>\ni</code>	$\frown$	<code>\frown</code>	$\perp$	<code>\perp</code>
$\geq$	<code>\geq</code>	$\parallel$	<code>\parallel</code>	$\exists$	<code>\exists</code>	$\emptyset$	<code>\emptyset</code>

your research. Then indicate if you have included such things as examples, sample experimental results, and data. Finally, clearly state the contribution of your article. Abstracts are usually between 100 and 150 words in length (longer abstracts are also acceptable).

### 3.2 Keywords

A *keyword* is a significant or descriptive word that helps identify a focal point in the subject matter (*i.e.*, principal topic) of a research article. Keywords provide a basis for searches for published articles on a particular subject, and have been the subject of considerable research (see, *e.g.*, [42]). After your Abstract in a TRS article, give a list of up to 8 keywords or phrases that indicate the principal topics covered in your article.

### 3.3 Introduction

The Transactions on Rough Sets (TRS) is the primary publication of the International Rough Set Society (IRSS [8]). Your article should begin with a thorough introduction. This Introduction should begin by clearly describing the problem(s) to be solved in your article. The description of the problem should be followed by an indication of the approach to solving the problem in your article. The problem-description should include a carefully crafted overview of the context for the problem. This context must include references to the principal (and, possibly, minor) works that are currently published on this subject (see, *e.g.*, [31] for a comprehensive list of papers on rough sets and their applications). After giving a description of the problem(s) solved in your article, clearly indicate the contribution(s) of your article. In doing this, you should compare and contrast your contribution(s) with related contributions in papers published by others. Finally, your introduction should conclude with an overview of the organization of your article.

### 3.4 Basic Concepts

After your introduction, introduce a section that gives an overview of the basic concepts that underly the contribution(s) of your article. This Basic Concepts section should be sufficiently detailed so that the connection between traditional basic concepts and notation and the new results presented by you in succeeding sections of your article, is made clear. Document the sources of the concepts and notation that you introduce in this Basic Concepts section.

### 3.5 Details

Include sections (and subsections) that give further details about the research introduced in the article. Such sections should make it clear what is intended by the research presented in the article. This can be accomplished by first giving

a detailed presentation of notation, data, experiments, algorithms, and theory related to what you present. After that, compare and contrast the results and methods presented in your article with results and methods presented by other researchers. This means that you must document the claims made in your article, and give a complete presentation of results found in works related to your research. By doing this, you will provide the reader with a clearcut view of your research and its context.

### 3.6 Notation

As an aid to understanding the intricacies of the research contribution presented in your article, be sure to introduce, explain, and document the notation that you use. It is preferable to use notation traditionally employed in rough set theory rather than using new notation that is equivalent to the common notation. In other words, the following rule-of-thumb should be followed: *be conservative in the introduction of new notation in connection with your research*. Otherwise, it becomes very difficult for reviewers to follow what you are writing. If you find it necessary to introduce new notation, then it should be clearly explained, illustrated, and distinguished from existing notation.

### 3.7 Data

Include sections (and subsections) that give a detailed description of the data used to obtain experimental results. This description should include sample data and an indication of the sources of your data. It will be very good if the data used in your article is made available either at [8] or at some other website.

### 3.8 Experiments

Include sections (and subsections) that give a detailed presentation of experimental results used in support of the claims made in your paper. It is recommended that a presentation of the experimental results include carefully labeled plots, charts, and tables. Document your experimental results by comparing and contrasting them to results obtained by other researchers.

### 3.9 Algorithms

Include detailed algorithms that specify the steps used in the research presented in the article. Each algorithm should be accompanied by an explanation of the principal steps, and the significance of the notation used in the algorithm. Document your algorithms by comparing and contrasting them with methods used by other researchers. Consider, for example, the reinforcement learning algorithm given in Alg. 1, which uses the `algorithm2e` package. This algorithm represents a form of off-policy Monte Carlo control (general case), which is explained in detail in [28, 37].

---

**Algorithm 1:** Off-Policy Monte Carlo Control Algorithm

---

**Input** : States  $s \in S$ , Actions  $a \in A(s)$  //  $A(s)$  is a set of actions in state  $s$ .  
**Output:** Policy  $\pi(s)$  // where  $\pi(s)$  is a policy in state  $s$  that controls the selection of a particular action in state  $s$ .

```
for (all  $s \in S, a \in A(s)$ ) do
     $\pi(s)$  is randomly chosen;
     $Q(s, a) \leftarrow$  arbitrary; // where  $Q$  is the value of an action  $a$  in state  $s$ 
     $N(s, a) \leftarrow 0$ ; // numerator of  $Q(s, a)$ 
     $D(s, a) \leftarrow 0$ ; // denominator of  $Q(s, a)$ 
end
while True do
    Select an action policy  $\pi'(s, a)$ , and use it to generate an episode:

     $s_0, a_0, r_1, s_1, a_1, r_2, \dots, s_{T-1}, a_{T-1}, r_T, s_T$ ; //  $r_i$  is the reward on action  $a_{i-1}$ 

     $\tau \leftarrow$  the latest time at which  $a_\tau \neq \pi(s_\tau)$ ;
    for each pair  $(s, a)$  appearing in the episode at time  $\tau$  or later do
         $t \leftarrow$  the time of first occurrence of  $s, a$  such that  $t \geq \tau$ ;
         $w \leftarrow \prod_{k=t+1}^{T-1} \frac{1}{\pi'(s_k, a_k)}$ ; // weight on  $R_t = \sum_{i=t+1}^T r_i$ ;
         $N(s, a) \leftarrow N(s, a) + wR_t$ ;
         $D(s, a) \leftarrow D(s, a) + w$ ;
         $Q(s, a) \leftarrow \frac{N(s, a)}{D(s, a)}$ ;
    end
    for each  $s \in S$  do
         $\pi(s) \leftarrow \operatorname{argmax}_a Q(s, a)$ ;
    end
end
```

---

Your algorithm should begin by specifying the input needed to perform the steps of the algorithm, and the output produced by the algorithm. Notice that Alg. 1 includes a specification of the input (pre-condition) and output (post-condition) for the algorithm. To specify your algorithm as shown in Alg. 1, you need to use the `algorithm2e` package. A very detailed description of the `algorithm2e` package with complete, detailed examples is available at [1].

Your algorithm should begin by specifying the input needed to perform the steps of the algorithm as well as the output produced by the algorithm. Notice that the L<sup>A</sup>T<sub>E</sub>X for Alg. 1 specifies the algorithm input as follows:

```
\Input{States  $s \in S$ , Actions  $a \in A(s)$  }
```

and the output produced by Alg. 1 is specified as follows:

```
\Output{Policy  $\pi(s)$  }
```

### 3.10 Theory

In the case where a new theory or theoretical results concerning rough sets is introduced, then sections (and subsections) should be introduced that present the notation, formal definitions, facts, lemmas, propositions and/or theorems that underly the new theory. Proofs of properties, principal lemmas, propositions and theorems should be included as part of the presentation of the underpinnings of the new theory. Document your theory by comparing and contrasting your definitions, lemmas, propositions and theorems with theories introduced by other researchers. By way of illustration of a formal definition as well as a theorem and its proof, consider the following brief example about a well-known property of the empty set.

**Definition 1. Truth Table for Implication.** If  $P$  and  $S$  are statements that can be only *true* or *false*, then the truth table for implication (denoted by  $\Rightarrow$ ) is shown in the table 4.

**Table 4.** Truth table for implication

$P$	$S$	$P \Rightarrow S$
<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>false</i>
<i>false</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>

**Definition 2. Equality of sets.** Let  $\forall x . x \in A$  denote for all  $x$  such that  $x$  belongs to an arbitrary universe  $A$ <sup>3</sup>. The sets  $A$  and  $B$  are equal if and only if (1) and (2) hold true.

$$\forall x . x \in A \Rightarrow x \in B. \quad (1)$$

$$\forall x . x \in B \Rightarrow x \in A. \quad (2)$$

**Definition 3. Empty Set** The *empty set* is a set which has no elements.

In other words, there exists no element that belongs to the empty set, or for each element of any universe of objects, a given element does not belong to the empty set, which is expressed by (3).

$$\forall x . x \notin \emptyset. \quad (3)$$

**Theorem 1.** (*The uniqueness of the null set*) *There is exactly one empty set.*

---

<sup>3</sup> The notation  $\forall$  (called a *universal quantifier*) was introduced by Gerhard Gentzen in 1935 [4]. The dot notation ‘.’ comes from [33], and serves as a means of eliminating parentheses in an assertion [30].



*Proof.* Let sets  $A$  and  $B$  be sets satisfying the definition of the empty set. From Def. 3, we obtain (4) and (5).

$$\forall x . x \notin A. \quad (4)$$

$$\forall x . x \notin B. \quad (5)$$

From (4), the premise in (6) is false, and from (5), the statement in the conclusion in (6) is false. Hence, from the truth table in Def. 1, the implication in (6) is true. Similarly, the implication in (7) is true.

$$\forall x . x \in A \Rightarrow x \in B. \quad (6)$$

$$\forall x . x \in B \Rightarrow x \in A. \quad (7)$$

Thus, sets  $A$  and  $B$  satisfy the requirements for the equality of sets (see Def. 2) and, therefore,  $A = B$ , which proves that there is exactly one empty set. ■

### 3.11 Formulas

All formulas should be numbered, and properly punctuated and formatted. Every equation should end with a period “.” unless there is a continuation which is marked by a comma “,”. Please punctuate a displayed equation in the same way as ordinary text, but with a small space in front of the end punctuation. Consider, for example, the problem of determining the value of an action  $a$  in state  $s$  with a function  $Q^\pi(s, a)$ , where a policy  $\pi$  is followed. During the episodic behavior of a system that learns from experience,  $Q^\pi(s, a)$  is defined relative to the expected value of returns. The return  $R = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T$  (*i.e.*, cumulative future discounted rewards) results from a sequence of actions, where  $\gamma \in [0, 1]$  is called a discount rate and  $r_i$  is the  $i^{th}$  reward. The basic idea is to choose actions during an episode that ends in a terminal state at time  $T$  so that the expected discounted return  $E^\pi(R)$  following policy  $\pi$  improves. A *policy*  $\pi(s, a)$  is a mapping from an environment state  $s$  to the probability of selecting a particular action. In choosing actions, it is necessary to estimate the expected value of  $R$ . Let  $Pr(X = x)$  denote the probability that  $X$  equals  $x$ . It is assumed that the return  $R$  (cumulative discounted future rewards) for a sequence of actions is a discrete random variable, and the probability  $Pr(R = r)$  is not known. In effect, if the episodic behavior of a swarm yields a sequence of returns  $R_1, \dots, R_n$  over  $n$  episodes, the value of the expectation  $E[R] = \sum_{j=1}^n x_j Pr(R_j = x_j)$  is not known. Monte Carlo methods (see, *e.g.*, [32]) offer an approach to estimating the expected value of  $R$ . Briefly, using the Monte Carlo method,  $Q^\pi(s, a)$  can be estimated using a weighted sum. Let  $w_i$  denote an importance sampling weight on  $R_i$ , and we obtain an approximate value of  $Q^\pi(s, a)$  using Eq. 8.

$$Q^\pi(s, a) \approx \frac{\sum_{i=1}^n w_i R_i}{\sum_{i=1}^n w_i}, \quad (8)$$

where  $\approx$  denotes *approximately equal to*. It is sometimes necessary to define a function with multiple lines. In that case, put a comma “,” at the end of each

line before the last line, and put a “.” at the end of the last line. For example, standard rough inclusion(SRI)  $\nu_{SRI}$  can be defined for any  $X, Y \subseteq U$  as in Eq. 9.

$$\nu_{SRI}(X, Y) = \begin{cases} \frac{|X \cap Y|}{|X|}, & \text{if } X \neq \emptyset, \\ 1, & \text{if } X = \emptyset. \end{cases} \quad (9)$$

In other words,  $\nu_{SRI}(X, Y)$  returns the degree of overlap between  $X$  and  $Y$ . In the case where  $X = Y$ , then  $\nu_{SRI}(X, Y) = 1$ . Under the assumption that  $X \neq \emptyset$ , the minimum inclusion value  $\nu_{SRI}(X, Y) = 0$ , is obtained when  $X \cap Y = \emptyset$  (*i.e.*,  $X$  and  $Y$  have no elements in common). There are two rules-of-thumb to observe relative to formulas. First, after you give a formula, give an explanation of each of the parts of the formula for the reader (*aka reviewer of your article*). Second, numbered formulas serve both as an aid in reading as well as writing a research article. Hence, whenever an article contains many formulas, numbering of the formulas is highly recommended.

### 3.12 Figures

Each figure should be accompanied by a short, meaningful caption. Either before or after a figure is given, give an explanation of the parts of the figure shown. For example, the results of reinforcement learning by a swarm inside an ecosystem is shown in Fig. [off]. The plot of normalized total  $Q$  values during 70 reinforcement learning episodes is shown in Fig. [off-policy] and RMS per episode for both the conventional, end-of-episode (EOE) and tail-of-episode (TOE) off-policy Monte Carlo algorithms is shown in Fig. [avgReward]. Even though the EOE algorithm does better than the conventional off-policy and TOE algorithms, the difference between the values in the plots in Fig. [off-policy] is not sufficient to conclude that one algorithm is better than the other. The oscillations in the two plots shown in Fig. [avg-reward] for both algorithms is not surprising, when one considers how  $Q$  values are estimated (see, *e.g.*, [37, 28]). Since Figures are usually not printed in color in the TRS, it is then necessary to convert color pictures to greyscale (see, *e.g.*, Fig. [greyscale]).

### 3.13 Images for Figures.

eps files are commonly used to supply images in Figures in a L<sup>A</sup>T<sub>E</sub>X article. A public domain utility for converting .jpg image files to .eps files can be downloaded from [9]. Included in the zip file for use with MS DOS is jpeg2ps.exe written by Thomas Merz. jpeg2ps can be used to convert JPEG files to PostScript Level 2 or 3 EPS. Merz writes that “jpeg2ps is not really a converter but a wrapper: it reads the image parameters (width, height, number of color components) in a JPEG file, writes the corresponding EPS header and then copies the compressed JPEG data to the output file”. To see how to convert jpg to eps files, put the jpeg2ps.exe file in the directory where you are working. Also put a jpg

file you wish to convert to eps format, in your working directory. Open the DOS Command Prompt window in your working directory.

### 3.14 References

A separate section entitled *References* should be included in your article. It is required that all references are proper and complete. The format of references should follow the guidelines contained in Springer's guide for LNCS authors [35]. The use of bibT<sub>E</sub>X is encouraged, especially in case of long reference lists. The authors should apply the bibliography (`splncls.bst`) style from Springer. It is very important that you follow the style for TRS references as illustrated in the following list.

- **Article in journal:** see, *e.g.*, [3, 21–24, 26, 28, 42],
- **Article in proceedings:** see, *e.g.*, [34],
- **Article in book:** see, *e.g.*, [11, 18, 27],
- **Book:** see, *e.g.*, [5, 7, 17, 19, 20, 29, 32, 37],
- **Research Report:** see, *e.g.*, [25, 2],
- **Website:** see, *e.g.*, [1, 8–10, 15, 16, 31, 35, 39, 41, 43].

## 4 Conclusion

The last section of your paper contains concluding remarks concerning what you have presented in the article. You should concisely state what you have done - straight to the end. Then highlight the main contribution of your article to the international research community. Finally, briefly state the future work, so that you or others can continue on this research.

## Acknowledgements

The author thanks the TRS team, in particular James F. Peters and Andrzej Skowron.

**Note:** Acknowledgements should appear in an unnumbered section immediately after the Conclusion section. In addition, to mentioning the names of those persons and/or organizations that have been helpful in writing your article. If appropriate, it is common to mention the sources of support of the research presented in the article. For example, you might write “This research has been supported by (enter the name of funding agency plus grant number(s))”.

## References

1. Algorithm2e: See documentation (includes sample algorithms) at [www.tug.org/tex-archive/macros/latex/contrib/algorithm2e/algorithm2e.pdf](http://www.tug.org/tex-archive/macros/latex/contrib/algorithm2e/algorithm2e.pdf)

2. Borkowski, M., Peters, J.F.: *Approximation Space Approach to Matching Image Segments*. Research Report, Computational Intelligence Laboratory, Department of Electrical & Computer Engineering, University of Manitoba (2005). See, also, Borkowski, M., Peters, J.F. (2006) in vol. V of the *Transactions on Rough Sets journal* [3].
3. Borkowski, M., Peters, J.F.: Matching 2D image segments with genetic algorithms and approximation spaces, *Transactions on Rough Sets journal* V (2006) 63-102.
4. Gentzen, G.: Untersuchen über das logische Schliessen I. *Mathematische Zeitschrift* 39 (1935) 176-210.  
See <http://dz-srv1.sub.uni-goettingen.de/cache/toc/D17147.html>
5. Goossens, M., Rahtz, S., Mittelbach: *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion*. Illustrating Documents with T<sub>E</sub>X and Postscript. Addison Wesley Longman, Inc., Harlow, England (1997).
6. graphicx package at  
<http://www.tug.org/tex-archive/macros/latex/contrib/graphicx-psmin/>.
7. Grätzer, G.: *Math into L<sup>A</sup>T<sub>E</sub>X*, 3rd Edition. Birkhäuser, Boston, MA, U.S.A., c/o Springer-Verlag, New York (2004).
8. International Rough Set Society: <http://roughsets.home.pl/www/>
9. jpeg2ps utility for converting jpg to eps file available at  
<http://gnuwin32.sourceforge.net/packages/jpeg2ps.htm>
10. Knuth, D.E.: Message to all users of TeX at  
<http://www-cs-faculty.stanford.edu/~knuth/cm.html>
11. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. In: [20] (1999) 3–98.
12. L<sup>A</sup>T<sub>E</sub>X symbols and corresponding L<sup>A</sup>T<sub>E</sub>X commands that produce the symbols<sup>4</sup>:  
<http://www.ctan.org/tex-archive/info/symbols/comprehensive/>
13. L<sup>A</sup>T<sub>E</sub>X tutorial to help you get started:  
<http://lahelper.sourceforge.net/mini-latex-tutorial.html>
14. For the latest version and current developments with L<sup>A</sup>T<sub>E</sub>X, see <http://www.latex-project.org/>
15. MikTeX implementation of T<sub>E</sub>X for the Windows operating system:  
<http://www.miktex.org/>
16. MikTeX 2.4 Manual. Revision 2.4.1779, 7 June 2006: <http://docs.miktex.org/manual/>
17. Mittelbach, F., Goossens, M.: *The L<sup>A</sup>T<sub>E</sub>X Companion*, 2nd Edition. Addison-Wesley, Boston, MA, U.S.A. (2004).
18. Pal, S.K., Peters, J.F., Polkowski, L., Skowron, A.: Rough-neural computing: An introduction. In: [19] (2004) 15–42.
19. Pal, S.K., Polkowski, L., Skowron, A. (Eds.): *Rough-Neural Computing. Techniques for Computing with Words*. Springer-Verlag, Berlin, Germany (2004).
20. Pal, S.K., Skowron, A. (Eds.): *Rough-Fuzzy Hybridization: A New Trend in Decision-Making*. Springer-Verlag, Singapore Pte. Ltd. (1999).
21. Peters, J.F., Skowron, A., Zdzisław Pawlak life and work (1906-2006), *Information Sciences* 177 (2007) 1-2.
22. Pawlak, Z., Skowron, A.: Rudiments of rough sets, *Information Sciences* 177 (2007) 3-27.
23. Z. Pawlak, A. Skowron, Rough sets: Some extensions, *Information Sciences* 177 (2007) 28-40.

---

<sup>4</sup> **Note:** some symbols require font files available from <http://www.ctan.org>.

24. Z. Pawlak, A. Skowron, Rough sets and Boolean reasoning, *Information Sciences* 177 (2007) 41–73.
25. Pawlak Z.: *Classification of Objects by Means of Attributes*, Institute for Computer Science, Polish Academy of Sciences, Report 429, March (1981).
26. Pawlak, Z.: Rough sets. *International Journal of Computer and Information Science* 11 (1982) 341–356.
27. Pawlak, Z., Skowron, A.: Rough membership functions. In: Yager, R., et al. (Eds.), *Advances in Dempster Shafer Theory of Evidence*. Wiley, N.Y., U.S.A. (1994) 251–271.
28. Peters, J.F., Henry, C.: Reinforcement learning with approximation spaces. *Fundamenta Informaticae* 71 (2006) 323–349.
29. Polkowski, L.: *Rough Sets. Mathematical Foundations*. Physica–Verlag, Berlin, Germany (2002).
30. Quine, W.V.O.: *Mathematical Logic*. Harper & Row, Publishers, NY (1951).
31. Rough Set Database System (RSDS): <http://rsds.wsiz.rzeszow.pl/rsds.php>
32. Rubinstein, R.Y.: *Simulation and the Monte Carlo Method*. John Wiley & Sons, Toronto, Canada (1981).
33. Russell, A.N., Whitehead, A.N.: *Principia Mathematica*, 3 vols. Cambridge University Press, London, GB, 2nd Ed. (1927).
34. Skowron, A., Stepaniuk, J.: Generalized approximation spaces. In: *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, San Jose, CA, U.S.A. (1994) 156–163.
35. Springer LNCS: See L<sup>A</sup>T<sub>E</sub>X files at  
<http://www.springer.com/east/home/computer/lncs?SGWID=5-164-7-72376-0>
36. subfigure package examples at  
<http://www.hep.man.ac.uk/u/jenny/jcwdocs/latex/figures.html>
37. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, U.S.A. (1998).
38. For detailed information about T<sub>E</sub>X, see  
<http://en.wikipedia.org/wiki/TeX>
39. TeXnicCenter system: <http://sourceforge.net/projects/texniccenter/>
40. TeXnicCenter system features, tools, add-ons, downloads:  
<http://www.texniccenter.org/>
41. Transactions on Rough Sets (TRS):  
<http://www.springer.com/west/home/computer/lncs?SGWID=4-164-6-99627-0>(see also <http://roughsets.home.pl/www/> for Guidelines for Special Issues of the TRS).
42. Turney, P.D.: Learning algorithms for keyphrase extraction. *Information Retrieval* 2/4 (2000) 303–336.
43. WITAS project (2001): <http://www.ida.liu.se/ext/witas/eng.html>
44. Petz, G., Karpowicz, M., Fuersch, H., Auinger, A., Stritesky, V., Holzinger, A.: Computational approaches for mining users opinions on the Web 2.0. *Information Processing & Management* 51(4), 510–519 (2015)
45. Holzinger, A., Dehmer, M., Jurisica, I.: Knowledge Discovery and interactive Data Mining in Bioinformatics - State-of-the-Art, future challenges and research directions. *BMC Bioinformatics* 15(S6), I1 (2014)
46. Holzinger, A.: *Biomedical Informatics: Discovering Knowledge in Big Data*. Springer, New York (2014)
47. Holzinger, A.: HumanComputer Interaction and Knowledge Discovery (HCI-KDD): What is the benefit of bringing those two fields to work together? In: *Lecture*

Notes in Computer Science LNCS 8127, pp. 319-328. Springer, Heidelberg, Berlin, New York (2013)

48. Holzinger, A.: Interactive Machine Learning for Health Informatics: When do we need the human-in-the-loop? Springer Brain Informatics (BRIN) 3 (2016) <http://dx.doi.org/10.1007/s40708-016-0042-6>