Master Seminar WS 18/19
Landnutzungsklassifikation - als Wettbewerb

Patrick Schäfer

URL: https://hu.berlin/landnutzung
ToC

• Today:
  • Today 26.11.: Introduction and Group Formation

• Next Friday:
  • By next Friday 2.11.: Choice of Topic
  • Next Friday 2.11.: Primer on Time Series Analytics
Motivation

- **Temporal data** is common in many data mining applications
- Application domains range from:
  - Sensor data: environmental sensors measure temperature, pressure humidity
  - Medical devices: electrocardiogram (ECG) and electroencephalogram (EEG)
  - Financial market: stock prices, economic indicators, product sales
  - Meteorological data: sediments from drill holes, **earth observation satellite data**
- See **UCR time series archive** for sample datasets
Time Series Definition

• Definition: A Time Series is a sequence (ordered collection) of \( n \) real values at time stamps \( (t_1, ..., t_n) \):

\[
T = (y_1, ..., y_n)
\]

• Time Series may be univariate or multivariate
  • Univariate: a single value \( y_i \) is associated with each time stamp \( t_i \).
  • Multivariate: \( m \) values \( y_i = (k_1, ..., k_m) \) are associated with each time stamp \( t_i \).

• The dimensionality of a time series refers to the number of values at each time stamp
Land Cover Mapping

- Land cover is the physical material that covers the surface of the earth like grass, asphalt, urbanized areas, trees, bare soil, water, ...
- The are two primary methods for capturing information on land cover:
  - Small-scale: Field survey
  - Large-scale: Remote sensed imaginary from satellites
- Land cover changes relate to (natural) processes:
  - Disasters: flooding, forest disturbance and degradation, wildfire
  - Anthropogenic activities: urbanization, agriculture, deforestation
  - Climate change

Satellite-based Earth Observation

- Satellites periodically observe Earth’s surface and acquire large, temporal, multi-spectral image sets
  - American Landsat 8, 30 m spatial resolution, 8 spectral bands, every 16 days
  - European Sentinel-2, 10 to 20 m spatial resolution, every 5 days

- Satellites enable the identification of the nature of spatial-temporal changes from space
Change of annual land cover of China

![Land cover maps](image)

**Land cover classes**
- Water
- Evergreen Needleleaf Forest
- Evergreen Broadleaf Forest
- Deciduous Broadleaf Forest
- Mixed Forest
- Open Shrublands
- Woody Savannas
- Grasslands
- Croplands
- Urban and Built-up
- Cropland and Natural Vegetation Mosaic
- Snow and Ice
- Barren or Sparsely Vegetated
- No Data

**Fig. 12.** Annual LULC maps of China, produced by random forest classification. (a) 1982, (b) 1992, (c) 2002, and (d) 2012.

From satellite images to pixel time series

Figure 2: Production of a time series datasets from satellite image series.

Periodic behaviour which can be slightly modulated by weather artifacts. These modulations result in distortions of canonical temporal profiles that are well handled by DTW. (2) Time series are too short for Bag-of-word-type approaches to perform best. NN-DTW cannot scale to the typical size of satellite datasets where it is common to have 100 million example time series. This is because to classify each query time series, we have to scan the entire 100 million training dataset. Even making the most of lower-bounding, this is completely infeasible. Figure 3 illustrates this point: while all datasets of the standard archive of time series can be classified in less than 30 minutes, creating a temporal land-cover map for just a city like Houston (16 million time series) assuming a bare minimum of 1 million training examples would take about a year to complete. To create a land-cover map of Texas (7 billion time series) with a reasonable training dataset of 100 million samples would require 30k years of computation.

With these motivations, this work tackles Contract Time Series Classification, where we would like to produce the most accurate classifier under a contracted time (obviously significantly smaller than running the NN-DTW). We propose a new algorithm that efficiently indexes the training database using a hierarchical K-means tree structure specifically designed for DTW. We will show that our algorithm reduces the time per query while retaining similar error to the state of the art, NN-DTW.

This paper is organized as follows. In section 2, we review some background and define the problem statement for our work. Then in section 3 we introduce and describe our approach. Section 4 shows the empirical evaluation for our approach. Lastly, section 5 offers some direction for our future work and we conclude our work in section 6.

2 Background and Motivation

2.1 Time Series Classification

Many time series classification algorithms in the literature such as Shapelets, 1-NN BOSS and SAX-VSM have been shown to be competitive (and sometimes superior) to the state of the art, NN-DTW. Nonetheless, as explained in the introduction, classification of the Satellite Image Time Series (SITS) is better tackled by NN-DTW. NN-DTW has been shown to be extremely competitive for many other applications. It has been argued that the widespread utility of NN-DTW is due to time series data having autocorrelated values, resulting in high apparent but low intrinsic dimensionality. Experimental comparison of DTW to most other highly cited distance measures on many datasets concluded that DTW almost always outperforms other measures.

A smoothed NDVI time-series

Fig. 3. Raw, and cleaned and smoothed NDVI time-series of mixed forest, croplands, and grasslands.

Why time series?

• We could also work with single images (single date)
• **Large gaps**: In some areas, the number of observations is low (due to shadows, clouds or haze):
  • Thus, repeated observations of the same location increase the chance of **cloud-free observations**
  • Repeated observations allow for identifying **temporal trends** (flooding, wildfire, anthropogenic activities (urbanization, agriculture), forest disturbance and degradation)
• Time series analysis has shown to be **superior** (more accurate) when compared to single-date methods
Spectral Sensors / Features

• Land surfaces absorb and reflect sunlight differently
• Satellite are equipped with multi-spectral sensors
  • Visible spectrum: Blue, green, red (absorbed by green vegetation)
  • Near-infrared (absorbed by water)
  • Short-wave infrared
• Derived Features: 3 computed indices:
  • for vegetation NDVI,
  • for water NDWI,
  • for brightness BI
• This allows machine-assisted mapping of land cover from space
Land cover usage classification

• Typically, experts label some reference land cover samples apriori
• Then we can train classifiers in a supervised manner
• However, selecting an appropriate number of samples is crucial and very time consuming as it is typically done manually (and error-prone)
• Thus, acquiring reference data is very expensive and time consuming, thus large labelled datasets are rare and very valuable
• Luckily, we are given the labelled data and only need to worry about choosing the right classifier
Classification (I)

• Given a set D of samples and a set of classes C. A classifier is a function f: D→C

• Supervised learning:
  • Training data: Obtain a set S of samples with their classes
  • Feature Space: Function v mapping a sample to a vector of features.
  • Model the characteristics of the samples in each class
  • Encode the model in a classifier function f operating on the feature vector: v: D→V, and f: V→C
  • Classification: Compute f(v(d))
Classification (II)

- **Non-time-series approaches** do not capture model **temporal dependencies** and treat values independently.
- **Time series** approaches explicitly capture temporal dependencies, which captures periodic changes over time (seasonality, agriculture, or harvesting).
- **Multivariate ~**: use all sensors as features but scalability might be a major issue (2d vector).
- **Univariate ~**: dimensionality reduction on features / use a single sensor or computed index (1d vector).
Time Series Classification TSC

• There are hundredths of base-classifiers
  • k-nearest neighbour, Naïve Bayes, Bayesian Networks, Graphical models, Decision Trees and Random Forests, Support Vector Machines, Neural Networks, ...
• Differences when using different base-classifiers on the same data/representation are often astonishing small
• Including time series models into classification has a larger impact on accuracy than the choice of classifier
• Despite progress in technology and data availability, training time series models on large-scale data is very challenging in practice
• Overall, effectiveness of classification depends on many variables: labelled data, classifier, representation, feature selection and engineering, evaluation method
Time Series Approaches

• Time series approaches are composed of a time series representation and a classifier

• Representations of time series can be divided into:
  • Using the whole time series (global trends)
  • Using sub-sequences of a time series (local trends)
    • Shapelets: absence or presence of (seasonal) patterns
    • Bag-of-Patterns (Dictionaries): use frequency of occurrences

• Base-Classifiers can then trained on this (new) representation
Subsequence vs Whole Series

• Suppose, we wish to build a classifier to distinguish between two kinds of plants: what features should one use?
• The contour of a leaf can in fact be interpreted as a time series
• Instead of comparing the entire shapes, it can be better to only compare small subsections (“Shapelets”)
• Here: the defining difference is that Urtica dioica has a stem that connects to the leaf at almost 90 degrees

Bag-of-Patterns: single occurrence vs frequency of occurrences

• Many signals are inherently periodic/repetitive (heartbeats, network traffic, weather, ...)
• We describe a signal by the frequency of occurrence of patterns
• Similar to the bag-of-words representation for documents, which is a histogram of word counts
• Problem: how to count the occurrences of real-valued subsequences?

What is a good Classifier?

- Problem: Finding a good classifier
  - Assigning as many samples as possible to their correct class
  - Involves proper feature engineering

- How do we know?
  - Use a (separate) gold standard test data set
  - Split the training data (beware of overfitting)
    - Train data for training the model
    - Validation data for evaluating the model

- A classifier $f$ is the better, the more samples it assigns to their correct classes on the validation data
Beware of Overfitting

• **Training data:** Let \( S \) be a set of instances with their classes
  - We can easily build a perfect classifier for a train set \( S \)
  - Model: map a sample to its nearest neighbour in \( S \) using identity.
  - This will produce perfect results for every sample \( s \in S \)
  - What about samples \( s \notin S \) (validation/test set)?
  - **So, don’t train a model and evaluate the model on the same data**

• **Overfitting**
  - If the model strongly depends on \( S \), \( f \) overfits – it will only work well if all future samples are very similar to the samples in \( S \)
  - You cannot detect overfitting when evaluation is performed on training set \( S \) only
Towards Overfitting

• f must generalize: Capture features that are typical for all samples in D, not only for the samples in S
• Still, often we only have a train set S for evaluation ...
  • We need to extrapolate the quality of f to unknown samples
• Usual method: Cross-validation (leave-one-out)
  • Divide S into k disjoint partitions (typical: k=10)
    • Leave-one-out: k=|S|
  • Learn model on k-1 partitions and evaluate on the k-th
  • Performed k times, each time evaluating on another partition
  • Estimated quality on new samples = average performance over k runs
What we’ll address here

• Working with **massive** pixel time series series
  • Feature engineering
  • (Time Series) Representations
  • (Time Series) Classification
  • Pre-processing
  • Scalability
  • Competition
Who should be here

• Master Informatik
  • Also: Wirtschaftsinformatik, Ms.Edu, Diplom informatik

• Ability to read English papers

• Ideally:
  • Knowledge on machine learning
  • Knowledge in statistics, probability theory, math
  • Or willingness to learn this
How it will work – competition part

• Every group (2-3 people) has to implement a (times series) classification method
  • Free choice pre-processing (normalization, interpolation, feature engineering, ...)
  • Your topic: Every group uses (a) different classifier(s)
  • Published codes and libraries (Weka (Java), scikit-learn (Python), R, matlab,...) allowed

• I will release a training set in November
  • A set of pixels time series with landcover classes assigned
  • Program, build, test and optimize your model
  • Include approach description in seminar talk and presentation

• We will evaluate your method on held-back test data
  • You will be given an unlabeled test set
  • We will use an automated evaluation web platform (Kaggle)
  • Submissions will be possible in January
  • Small price for best average accuracy among all groups
Competition part

• A massive land cover pixel time series (TS) dataset
  • 46 geometrically and radiometrically corrected images taken by FORMOSAT-2
  • Train data: 6,091,037 pixels TS, 2,4GB
  • Test data (hold-back): 2,614,122 pixels TS, 1,0GB

• In total 3x46 values per pixel time series
  • 46 time stamps between 06.02 and 29.11.2006
  • 3 surface reflectances: Near-Infra-Red, Red, Green

• Contains missing values ‘?’

• Overall, 24 land cover classes, labelled by experts

• Note: This data is provided for the class only and it has to be deleted once the seminar is over
## Competition: Excerpt of the data

<table>
<thead>
<tr>
<th>Data Excerpt</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,226,57,57,285,47,50,401,49,49,408,43,51,459,45,53,320,20,25,460,69,82,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,156,51,52,154,30,36,292,30,33,316,30,43,417,38,44,288,16,19,433,64,70,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,254,71,63,285,66,64,347,58,57,365,54,57,378,58,59,248,30,30,431,69,82,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,221,64,56,302,48,54,397,44,48,412,41,48,450,43,53,312,20,25,467,72,83,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,153,51,50,157,28,34,32,37,354,30,44,426,36,46,312,19,19,433,64,69,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,262,69,63,325,59,68,336,55,57,369,50,57,375,51,57,262,27,27,403,67,78,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,235,61,57,323,48,54,397,47,49,418,41,49,454,37,53,315,20,26,462,69,85,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,226,57,57,285,47,50,401,49,49,408,43,51,459,45,53,320,20,25,460,69,82,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,156,51,52,154,30,36,292,30,33,316,30,43,417,38,44,288,16,19,433,64,70,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,254,71,63,285,66,64,347,58,57,365,54,57,378,58,59,248,30,30,431,69,82,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,221,64,56,302,48,54,397,44,48,412,41,48,450,43,53,312,20,25,467,72,83,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,153,51,50,157,28,34,32,37,354,30,44,426,36,46,312,19,19,433,64,69,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,262,69,63,325,59,68,336,55,57,369,50,57,375,51,57,262,27,27,403,67,78,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,235,61,57,323,48,54,397,47,49,418,41,49,454,37,53,315,20,26,462,69,85,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,226,57,57,285,47,50,401,49,49,408,43,51,459,45,53,320,20,25,460,69,82,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,156,51,52,154,30,36,292,30,33,316,30,43,417,38,44,288,16,19,433,64,70,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,254,71,63,285,66,64,347,58,57,365,54,57,378,58,59,248,30,30,431,69,82,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,221,64,56,302,48,54,397,44,48,412,41,48,450,43,53,312,20,25,467,72,83,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,153,51,50,157,28,34,32,37,354,30,44,426,36,46,312,19,19,433,64,69,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,262,69,63,325,59,68,336,55,57,369,50,57,375,51,57,262,27,27,403,67,78,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
<tr>
<td>0,235,61,57,323,48,54,397,47,49,418,41,49,454,37,53,315,20,26,462,69,85,?,?,?</td>
<td>Excerpt of the data</td>
</tr>
</tbody>
</table>

[...]
Direction: Features

• Scalability will be a major issue [1]!

• Directions:
  • How to handle missing values (interpolation, ...)?
  • Use additional indices: brightness, normalized difference vegetation index
  • Test additional features: minimum, maximum, average, range, ...
  • Address Scalability: feature selection, dimensionality / noise reduction, sampling
  • Use a time series representation: Shapelets / Dictionary-based, ...

---

A List of Possible Approaches

• **(non-time series) based-Classifiers**
  - SVM, logistic regression, random forests/decision trees, gradient boosting trees, XGBoost

• **Whole-Series-based Classifiers**
  - 1-NN Dynamic Time Warping
  - 1-NN Euclidean Distance
  - Proximity Forests

• **Shapelet-based Classifiers**
  - Univariate: Fast Shapelets (FS), Learning Shapelets (LS), Shapelet Transform (ST)
  - Multivariate: gRSF

• **Dictionary-based Classifiers**
  - Univariate: BoP, SAX VSM, TSBF, BOSS, BOSS VS, WEASEL
  - Multivariate: SMTS, WEASEL+MUSE, LPS

• **Deep Learning Classifiers**
  - ResNet, FCN, Encoder, MLP, Time-CNN, TWIESN, MCDCNN, MCNN, t-LeNet

• **Ensembles of Core Classifiers**
  - Univariate: EE PROP, COTE
A List of Non-Time Series Classifiers

Table 4
Strengths and weaknesses of algorithms used for large-area land cover characterization with time-series optical data.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Strengths/characteristics</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Artificial Neural Networks</strong></td>
<td>• Manage well large feature space</td>
<td>• Needs parameters for network design</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>• Indicate strength of class membership</td>
<td>• Tends to overfit data</td>
</tr>
<tr>
<td></td>
<td>• Generally high classification accuracy</td>
<td>• Black box (rules are unknown)</td>
</tr>
<tr>
<td></td>
<td>• Resistant to training data deficiencies—requires less training data than DT</td>
<td>• Computationally intense</td>
</tr>
<tr>
<td></td>
<td>• Slow training</td>
<td>• Looping and aliasing occurrence</td>
</tr>
<tr>
<td><strong>Clustering (partitioning)</strong></td>
<td>• Do not need previous knowledge</td>
<td>• Cluster-class correspondence not assured</td>
</tr>
<tr>
<td></td>
<td>• Do not need samples</td>
<td>• Complex identification of classes</td>
</tr>
<tr>
<td><strong>Decision trees</strong></td>
<td>• No need of any kind of parameter</td>
<td>• Computationally intensive</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>• Easy to apply and interpret</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Handle missing data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Handle data of different types (e.g. continuous, categorical)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and scales</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Handle non-linear relationships</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Insensitive to noise</td>
<td></td>
</tr>
<tr>
<td><strong>Gaussian Maximum likelihood</strong></td>
<td>• Simple application</td>
<td>• Parametric</td>
</tr>
<tr>
<td>Parametric</td>
<td>• Easy to understand and interpret</td>
<td>• Assumes normal distribution of data</td>
</tr>
<tr>
<td></td>
<td>• Predicts class membership probability</td>
<td>• Large training sample needed</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td>• Manages well large feature space</td>
<td>• Needs parameters: regularization and kernel</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>• Insensitive to Hughes effect</td>
<td>• Poor performance with small feature space</td>
</tr>
<tr>
<td></td>
<td>• Works well with small training dataset</td>
<td>• Computationally intense</td>
</tr>
<tr>
<td></td>
<td>• Does not overfit</td>
<td>• Designed as binary, although variations exist</td>
</tr>
<tr>
<td><strong>Random Forests</strong></td>
<td>• Capacity to determine variable importance</td>
<td>• Decision rules unknown (black box)</td>
</tr>
<tr>
<td>Non-parametric</td>
<td>• Robust to data reduction</td>
<td>• Computationally intense</td>
</tr>
<tr>
<td></td>
<td>• Does not over-fit</td>
<td>• Needs input parameters (#trees and #variables per node)</td>
</tr>
<tr>
<td></td>
<td>• Produces unbiased accuracy estimate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Higher accuracy than DT</td>
<td></td>
</tr>
<tr>
<td><strong>Bagging</strong></td>
<td>• Provides measures of classification confidence</td>
<td>• Complex incomprehensible classifiers</td>
</tr>
<tr>
<td></td>
<td>• Does not overfit</td>
<td></td>
</tr>
<tr>
<td><strong>Boosting</strong></td>
<td>• Provides measures of classification confidence</td>
<td>• Stops if a classifier achieves zero training set error</td>
</tr>
<tr>
<td></td>
<td>• Does not overfit</td>
<td>• Complex incomprehensible classifiers</td>
</tr>
<tr>
<td></td>
<td>• Robust to noise</td>
<td>• Ineffective if excessive error in training sample</td>
</tr>
</tbody>
</table>

General Literature

• On Landcover Classification - To be read by everyone

• On Univariate Time Series Classification

• On Deep Learning for Time Series Classification

• Webseiten / Wettbewerbe:
  • TiSeLaC: Time Series Land Cover Classification Challenge
  • AALTD’16 Challenge : on multivariate time series data
  • http://timeseriesclassification.com : a website dedicated to univariate time series classifiers

• On classification / machine learning in general
  • Alpaydin: Introduction to Machine Learning, MIT press 2014
  • Ertel: Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung, Springer 2016
Allgemeine Hinweise

• Literaturrecherche ist notwendig
  • Die ausgegebenen Arbeiten sind Anker
  • Weiterführende Arbeiten müssen herangezogen werden
  • Auch Grundlagen nachlesen

• Kritisch lesen
  • Keine Angst vor nicht ganz zutreffenden Aussagen – solange gute Gründe vorhanden sind
  • Begründen und argumentieren
  • Kritikloses Abschreiben ist fehl am Platz
Next steps...

• Today: Introduction and group formation
• Friday 2.11.: Lecture “Primer on Time Series Analytics”
• Between today and Friday 2.11.: choose a topic
• Flash presentation (7.12. 15-16 Uhr, RUD 25 4.410):
  • Before 30.11.18: meet me to discuss topic
  • Present ideas and method in 5min
• Blockseminar (1.2. 15-18 Uhr, RUD 25 4.410)
  • Before 31.01.19: meet me to discuss slides
  • Present your topic (30-40min) at the Blockseminar
• Seminar Thesis before 31.03.2019!
  • write seminar thesis (~20 pages)
## Topics and Groups

<table>
<thead>
<tr>
<th>Topic</th>
<th>Assigned to (groups of 2-3)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(non-time series) based-Classifiers</strong></td>
<td>Alexej</td>
</tr>
<tr>
<td>SVM, logistic regression, random forests/decision trees, gradient boosting trees, XGBoost, Bayesian methods</td>
<td></td>
</tr>
<tr>
<td><strong>Whole-Series-based Classifiers</strong></td>
<td></td>
</tr>
<tr>
<td>Dynamic Time Warping, Euclidean Distance, Proximity Forests</td>
<td></td>
</tr>
<tr>
<td><strong>Shapelet-based Classifiers</strong></td>
<td></td>
</tr>
<tr>
<td>Univariate: Fast Shapelets (FS), Learning Shapelets (LS), Shapelet Transform (ST)</td>
<td></td>
</tr>
<tr>
<td>Multivariate: gRSF</td>
<td></td>
</tr>
<tr>
<td><strong>Dictionary-based Classifiers</strong></td>
<td>Arik</td>
</tr>
<tr>
<td>Univariate: BoP, SAX VSM, TSBF, BOSS, BOSS VS, WEASEL</td>
<td></td>
</tr>
<tr>
<td>Multivariate: SMTS, WEASEL+MUSE, LPS</td>
<td></td>
</tr>
<tr>
<td><strong>Deep Learning Classifiers</strong></td>
<td></td>
</tr>
<tr>
<td>ResNet, FCN, Encoder, MLP, Time-CNN, TWIESN, MCDCNN, MCNN, t-LeNet</td>
<td></td>
</tr>
<tr>
<td><strong>Ensembles of Core Classifiers</strong></td>
<td></td>
</tr>
<tr>
<td>Univariate: EE PROP, COTE</td>
<td></td>
</tr>
</tbody>
</table>

---

Siehe Webseite: [https://hu.berlin/landnutzung](https://hu.berlin/landnutzung)
Questions?
Hinweise zum Vortrag

- 30-40 Minuten plus Diskussion
- Klare Gliederung
- Themenauswahl: Lieber verständlich als komplett
- Bilder und Grafiken; Beispiele
- Eher Stichwörter als lange Sätze
- Vorträge können auch unterhaltend sein
  - Gimmicks, Rhythmuswechsel, Einbeziehen der Zuhörer, etc.
- Adressat sind alle Teilnehmer, nicht nur die Betreuer
- Technik: Laptop? Powerpoint? Apple?
- Siehe: [https://hu.berlin/checkliste_seminar](https://hu.berlin/checkliste_seminar)
Hinweise zur Ausarbeitung

• Eine gedruckte Version abgeben
  • Selbstständigkeitserklärung unterschreiben
• Eine elektronische Version schicklen
• Referenzen: Alle verwendeten und nur die
  • Im Text referenzieren, Liste am Schluss
• Korrekt zitieren
  • Vorsicht vor Übernahme von kompletten Textpassagen; wenn, dann deutlich kennzeichnen
  • Aussagen mit Evidenz oder Verweis auf Literatur versehen
• Verwendung von gefundenen Arbeiten im Web
  • Möglich, aber VORSICHT
  • Eventuell Themenschwerpunkt verschieben – Betreuer fragen
• Siehe: https://hu.berlin/checkliste_seminar
Format

• Benutzung unserer Latex-Vorlage
• Nur eine Schriftart, wenig und konsistente Wechsel in Schriftgröße und –stärke
• Inhaltsverzeichnis
• Bilder: Nummerieren und darauf verweisen
• Referenzen:

• Darf man Wikipedia zitieren?
  • Ja, aber nicht dauernd

• Siehe: https://hu.berlin/checkliste_seminar
Sehr geehrte Institutsmitglieder,

am Dienstag, den 30.10.2018, verteidigt ab 10.00 Uhr c.t. in Raum 4.410 Herr Arik Ermshaus seine Bachelorarbeit mit dem Titel

Deep Spatio-Temporal Time Series Land Cover Classification

Alle Interessierten sind herzlich eingeladen.

Beste Grüße,
Ulf Leser