Predict Missing Links in Citation Networks

INF 554 – Machine Learning 1

4our
Fabrizio Indirli • Leon Kloten • Seongbin Lim • Martin Wohlfender

Content

École Polytechnique
Université Paris-Saclay

- A citation network is a social network which contains paper sources that are linked by co-citation relationships
- Represented by a graph, research papers are nodes and there is an edge between two nodes if one paper cites the other

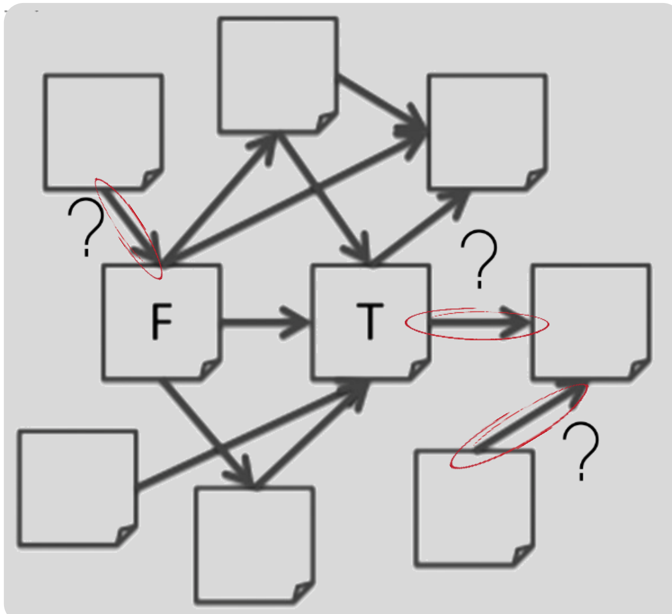
- In the data challenge, a citation network is given of which edges have been randomly deleted
- The given citation network is defined as a graph where research papers are nodes that are linked by an edge if one of the two papers cites the other

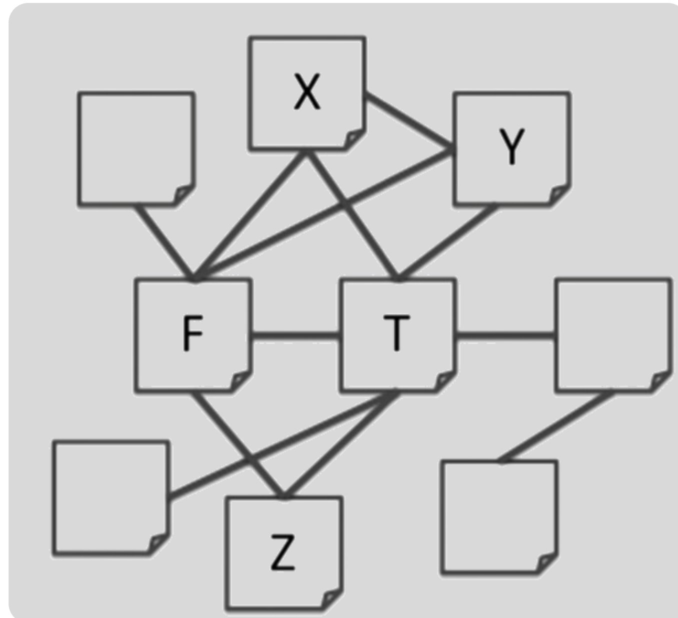
Background

Reconstruction of the Initial Citation Network

Citation Networks…

… As Directed Graph	**… As Undirected Graph**





- ➢ Given is a **citation network, defined as** an **a graph,** of which **certain edges** have been **randomly deleted**
- ➢ **Accurate reconstruction of** the initial **citation network using graph-theoretical, textual, and other information**

1 **2** 3 4 5 6 7

- Clear and well-structured methodological approach in order to fully understand the given problem, analyze it and find the correct and best suitable solution

Approach

Methodological Approach

Methodology

1 Data Preprocessing and Analysis

- Detailed analysis of publication years as well as examination of certain types of edges
- Distribution analysis of the number of citations and references
- Data Preprocessing in order to clean faulty data

2 Engineering of Features Using Given Information

- 17 implemented features in the final model with additional 3 discarded features
- For performance measurement, new features have been tested individually first before being tested together with the already implemented features
- Final feature selection based on our main model (XGBoost model)

3 Classifiers

- Definition of our main classifier models (XGBoost, Neural Network, Random Forest as well as Linear Regression)
- Further classifiers have been defined with standard parameters as baselines (Support Vector Machine, KNeighbors, Decision Tree, One Versus Rest)

4 Tuning & Results

- Features engineering and selection
- Definition of our models
- Tuning with K-fold cross validation
- Overfitting avoidance with *Early stopping, Dropout (NN)*

ÉCOLE POLYTECHNIQUE UNIVERSITÉ PARIS-SACLAY

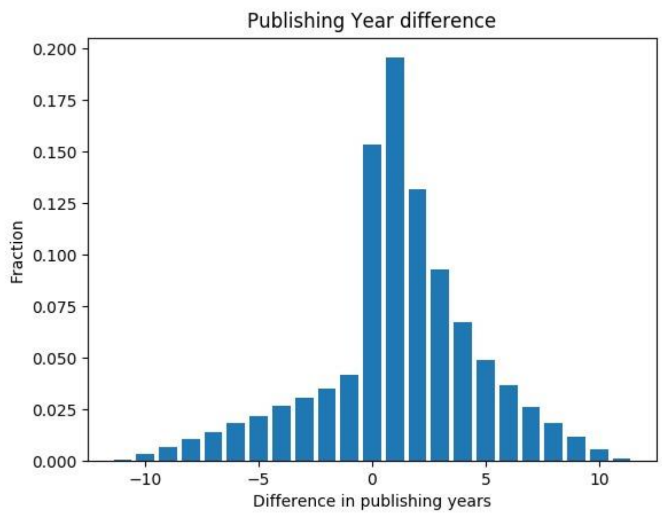
1 2 **3** 4 5 6 7

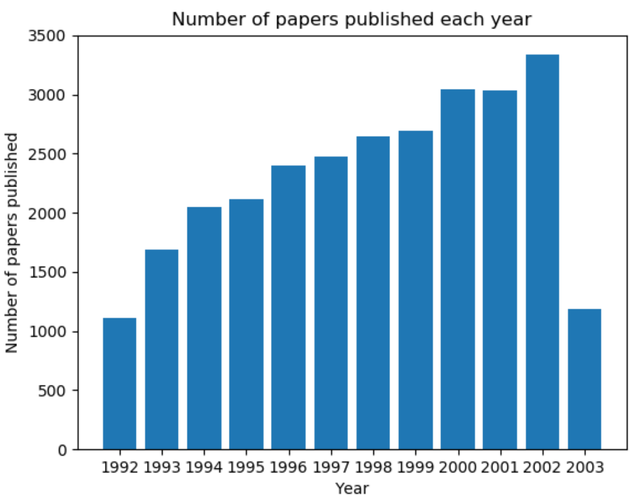
1. **Publication year difference**
2. Edges analysis
3. Distribution of number of citations and references

Data Analysis

From Preliminary Analysis on Data Some Useful Information has Been Extracted

Publication Year difference distribution





- **Most of** the **linked articles** in the training set have a **temporal difference between 0 and 4 years**
- In almost **20% of the citations** in the training set **the cited paper was published after the citing article**
- **The number of articles published per year has increased on a year by year base**

Data Analysis

Analysis of Edges as Being Rather Directed or Undirected

Considering the Citation Network as *Directed* vs *Undirected*

In literature, citation networks are often considered **directed graphs.**
However, from our data analysis on *publication year difference* it follows that, considering the network as a **directed graph,** some articles would cite other resources from the future. This led to the formulation of 3 hypotheses:

I. This citation network should be intended as an *undirected graph*. However, this would not follow the usual definition of citations networks

II. The direction of the edges should be inferred from the sign of the Publishing Years difference. However, implementing this solution had a negative impact on the prediction score; in addition, there isn't a clear method to decide the direction of edges whose difference in their publishing years is 0

III. Citations may have been done in the future on purpose. This may happen, for example, when the papers are published online before their official publication on a journal, or if the authors know each other and share their results before the official publication

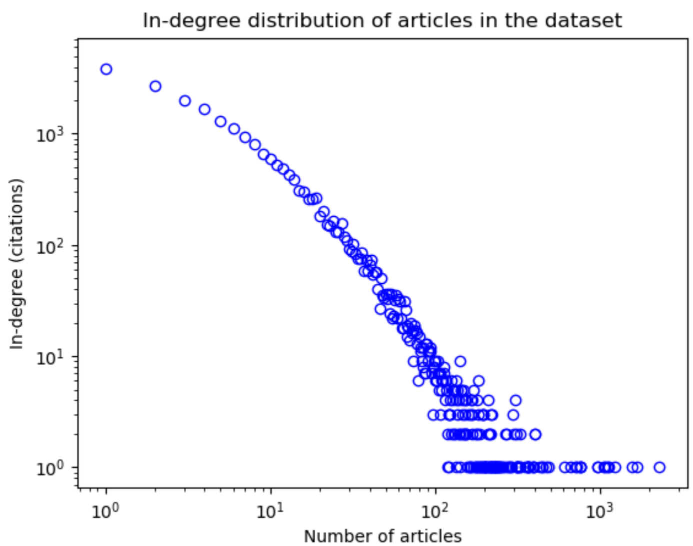
- **In our project, we** have **used an hybrid approach** that **mainly considers the network as** an ***undirected*** *graph*
- The **graph** was **considered as *directed* only** for the **calculations of the 2 *HITS* features** and for **the *authors-to-authors citations* feature**

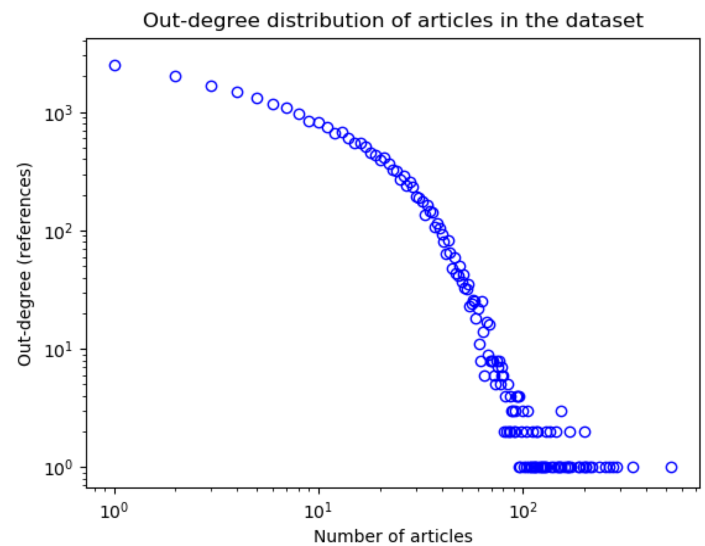
Data Analysis

From Preliminary Analysis on Data Some Useful Information has Been Extracted

1. Publication year difference
2. Edges analysis
3. **Distribution of number of citations and references**

Distributions of number of citations and references





- The **in-degree** (number of received citations) **and out-degree** (number of references) **distributions follow** the ***Power law distribution***
- This distribution is **very common in social phenomena** such as **social networks** as well as **citations networks**

- **Authors Preprocessing:** In the dataset, the authors' data was polluted with additional characters that were removed to ensure that the features regarding the authors are computed correctly
- **Titles & Abstracts Preprocessing:**
  - ✓ Stopwords removal
  - ✓ Stemming

Data Preprocessing

Data of Authors, Titles and Abstract has been cleaned to guarantee a correct feature computation

Authors Preprocessing

- By using **regular expressions**, the authors column was cleaned from special characters such as // /; \' \~ \" and others
- Also, the parentheses and their content were removed

Verena Sch\"on, Michael Thies
Aram A. Saharian (Yerevan State Univ)
Philippe Droz-Vincent (Meudon, France)

Verena Schon, Michael Thies
Aram A. Saharian
Philippe Droz-Vincent

Titles & Abstracts Preprocessing

- Using the NLTK package, the following pre-processing steps have been executed on *titles* and *abstracts*:
  - ✓ **Stopwords removal:** Removal of articles/ prepositions/ etc.
  - ✓ **Stemming:** Using a PorterStemmer, to reduce inflected or derived words to their root
- This was done to reduce the size of the TF-IDF matrix and to reduce noise in the *abstracts similarities* and *titles overlap* features

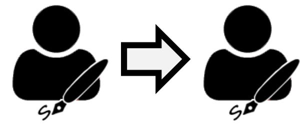
- ➢ **This Preprocessing was important to correctly compute the features *common authors, authors-to-authors citations, authors-to-journal citations, abstracts similarities, titles overlap***
- ➢ **In fact, using the cleaned datas, instead of the raw ones, slightly increased the importance of those features**

Overview of Features

Various Features Have Been Tested With the Goal to Achieve the Highest Prediction

- For each edge *(i,j)* in the citation network, various features have been computed:
  - **Topological Properties** of the involved nodes/ edges
  - **Intrinsic Properties** of the involved articles
- Afterwards each feature was tested for its impact with the goal of performance improvement (also see section 5)

$$Cits(i,j) = \sum_{a \in Auths_i} \Big(\sum_{b \in Auths_j} M(a,b) \Big)$$

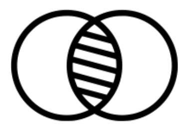


$$Jac(i,j) = \frac{|Neighbors_i \bigcap Neigbors_j|}{|Neighbors_i \bigcup Neigbors_j|}$$

$$C(i) = \frac{1}{\sum_{\forall j} distance(i,j)}$$

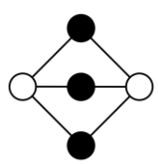
$$DifCentrality(i,j) = C(i) - C(j)$$

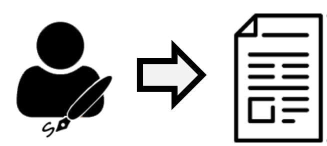
$$SumCentrality(i,j) = C(i) + C(j)$$

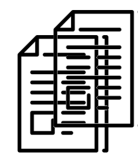


$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

$$aa(i,j) = \sum_{w \in (\Gamma(i) \bigcap \Gamma(j))} \frac{1}{\log(\Gamma(w))}$$







$$Pas(i,j) = |\Gamma(u)||\Gamma(v)|$$

$$w_p[i] = tf_{i,p} \cdot \log \frac{numPapers}{df_i}$$

$$cosineSim(i,j) = w_i \cdot w_j$$

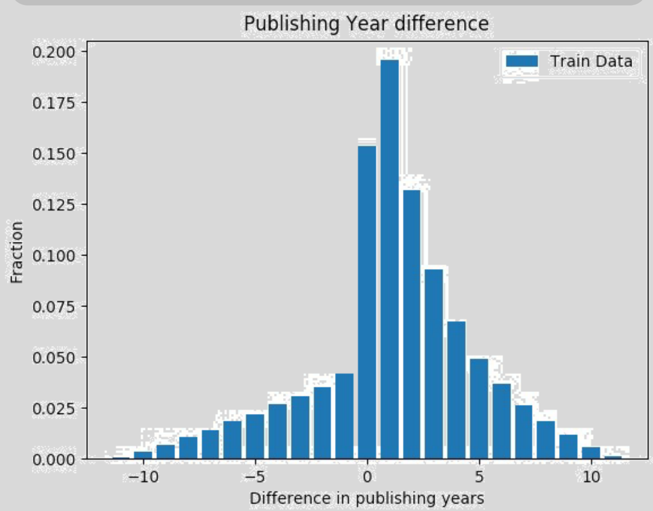
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Main Features

1 2 3 **4** 5 6 7

Various Features Have Been Tested With the Goal to Achieve the Highest Prediction

Temporal Difference



- Difference between the second article's year and the first article's year
- Most of the linked articles with temporal difference between 0 and 4 years

Number of Paths

- For an edge *(i,j)* this feature is the number of simple (acyclic) paths of length 3 from *i* to *j*

Abstracts TF-IDF Similarity

$$w_p[i] = tf_{i,p} \cdot \log \frac{numPapers}{df_i}$$

$$cosineSim(i,j) = w_i \cdot w_j$$

- Cosine similarity of term frequency (inverse document frequency vectors extracted from source and target abstracts)
- Reduction of the weight of terms with meaningless information

Source Hub & Target Authority

For each edge (i,j):

✓ **Hub Score of source i:** estimates the value of links from i to other nodes

✓ **Authority Score of target j:** estimates the value of paper j by valuing the Hub Scores of other papers citing j

- Values are calculated using the HITS algorithm
- Originally a link analysis algorithm developed to rank webpages

Resource Allocation

$$s(i,j) = \sum_{w \in (\Gamma(i) \bigcap \Gamma(j))} \frac{1}{\Gamma(w)}$$

- Inspired by the resource allocation process taking place in networks
- Each node i is considered as producer of a resource

Preferential Attachment

$$Pas(i,j) = |\Gamma(u)||\Gamma(v)|$$

- Based on the assumption that a regularly cited paper is likely to be cited even more in future papers
- Accordingly, barely cited papers are likely to be "forgotten"

Features

Various Features Have Been Tested With the Goal to Achieve the Highest Prediction

Adamic/ Adar Index

$$aa(i,j) = \sum_{w \in (\Gamma(i) \bigcap \Gamma(j))} \frac{1}{\log(\Gamma(w))}$$

- A measure introduced to predict links in a social network according to the amount of shared links between the two nodes
- The $\Gamma(w)$ is the set of w's neighbors

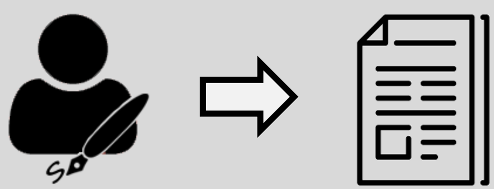
Sum and Difference of Closeness Centralities

$$C(i) = \frac{1}{\sum_{\forall j} distance(i,j)}$$

$$SumCentrality(i,j) = C(i) + C(j) \qquad DifCentrality(i,j) = C(i) - C(j)$$

- The Closeness of a node is a measure of centrality in a network, calculated as the reciprocal of the sum of the length of the shortest path between the node and all the other nodes in a graph
- Assumption: prob. of an edge between two nodes with high centrality is higher than the prob. of an edge between less central nodes
- Measures the difference of two nodes in terms of centrality

Source Authors to Target Journal Citations



- Sum of the number of citations that the authors of the source paper have made on the target's paper journal

Source Authors to Target Authors Citations

$$Cits(i,j) = \sum_{a \in Auths_i} (\sum_{b \in Auths_j} M(a,b))$$

- Calculates for each edge (i,j), how often an author of i cited an author of j
- A m*m matrix M is calculated, with m being the total number of diff. authors and M(i,j) the number of citations from author i of j

Titles Overlap

- Number of identical words in the titles of the two papers

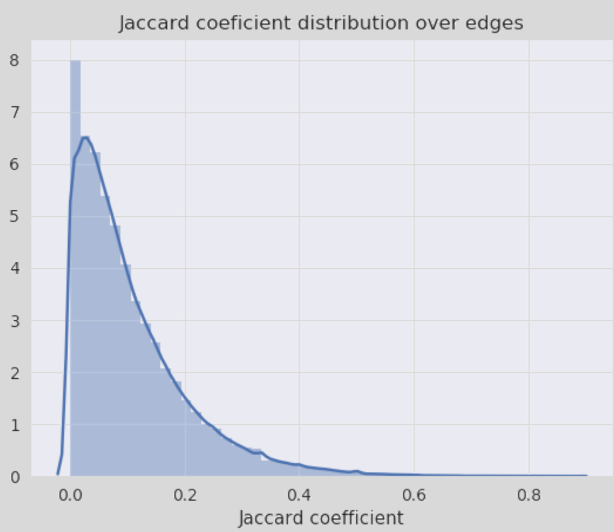
Features

Various Features Have Been Tested With the Goal to Achieve the Highest Prediction

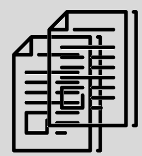
Jaccard Coefficient (Neighbors)

$$Jac(i,j) = \frac{|Neighbors_i \bigcap Neigbors_j|}{|Neighbors_i \bigcup Neigbors_j|}$$

- Index of similarity over common neighbors that does not penalize nodes with less neighbors



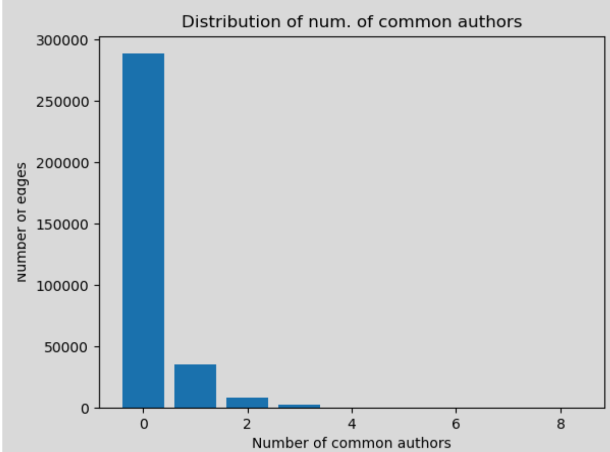
Same Journal



- A Boolean feature that is equal to 1 if the two papers have been published on the same journal.
- We thought that authors who publish on a journal, also read the same journal and hence have an higher probability of citing articles from that journal
- However, this feature seems to have a very marginal impact on prediction performance.

Number of Common Authors

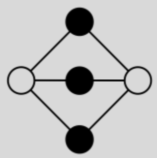
- Number of common authors of the two documents.
- Before calculating this feature, the authors are preprocessed to remove parentheses (which often contain unnecessary information) or special characters (which are often typos)
- This feature had a limited impact on prediction because for most of the citations the number of common authors = 0



Discarded Features

Various Features Have Been Tested With the Goal to Achieve the Highest Prediction

Number of Common Neighbors



- Number of nodes that are directly linked to both i and j
- It shares the same idea with Jaccard coefficient

LSA (Latent Semantic Analysis) of TF-IDF abstract

$$X \approx X_k = U_k \Sigma_k V_k^T$$

- LSA, known as **truncated singular value decomposition**, is a dimensionality reduction technique that preserves the variance as much as possible
- Solution for **synonymy** and **polysemy**

- A LSA model (# of components = 100)
- Sum of explained variance = 0.187
- Too small to be more useful.

Connectivity

$$Cond(i,j) = \sum_{p \in Paths(i,j)} \frac{1}{len(p)}$$

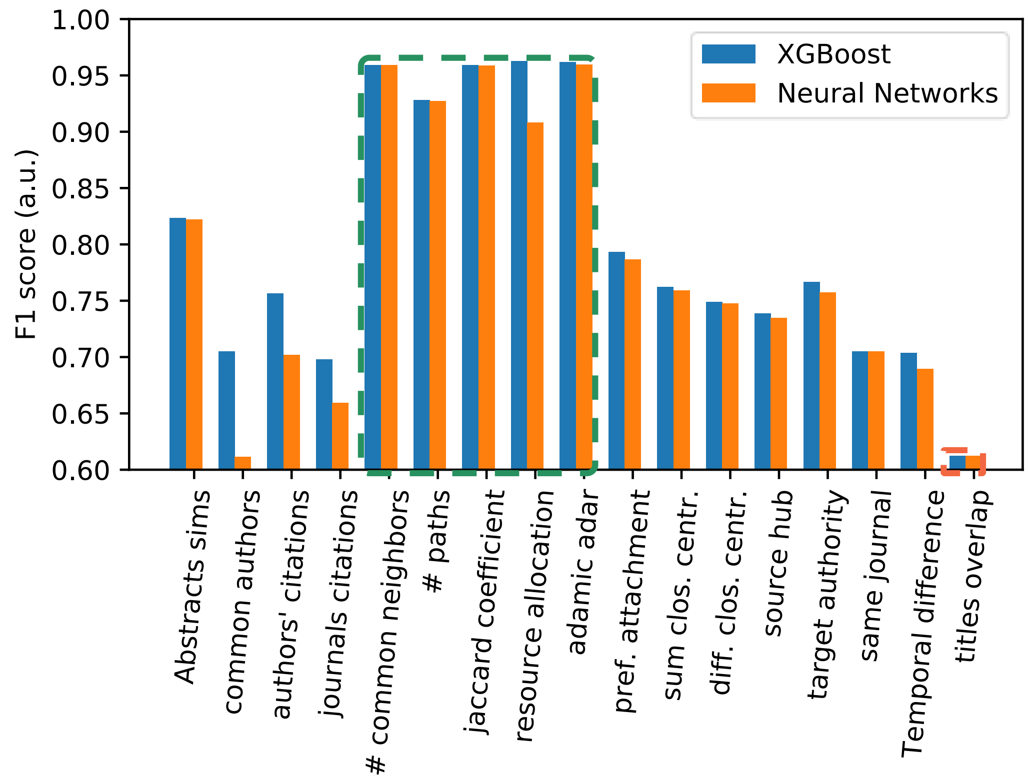
- This feature considers the graph as an **electrical circuit**: in a circuit, the **conductivity** of a dipole is the inverse of its resistance
- We define the **resistance of a simple path** between 2 nodes as the length of the path
- If there exist multiple paths between nodes *i* and *j*, the **total resistance** between *i* and *j* is calculated as in the case of **parallel resistors** in an electrical circuit
- This feature was eventually discarded because it didn't improve the prediction score; this may be due to the presence of many other graph-related features

- Single feature importance
- 5-fold cross-validation
- Two main classifiers
  - XGBoost
  - Neural Networks
- **Tier 1:**
  - # common neighbors
  - # paths (len=3)
  - Jaccard coefficient
  - Resource allocation
  - Adamic adar
- **Tier 2:**
  - Rest
- **Least important:**
  - Titles overlap

The Importance of the Implemented Features

Titles Overlap as Least Predictive Features

Single Feature Score (w/5-fold cross-validation)



- The **Number of Common Neighbors**, the **Number of Paths** (length=3), **Jaccard Coefficient**, **Resource Allocation** and the **Adamic/ Adar Index** yielded the **highest F1 scores**
- **Titles Overlap** turned out the **least important** feature while **Common Authors** led to **highly different results**

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

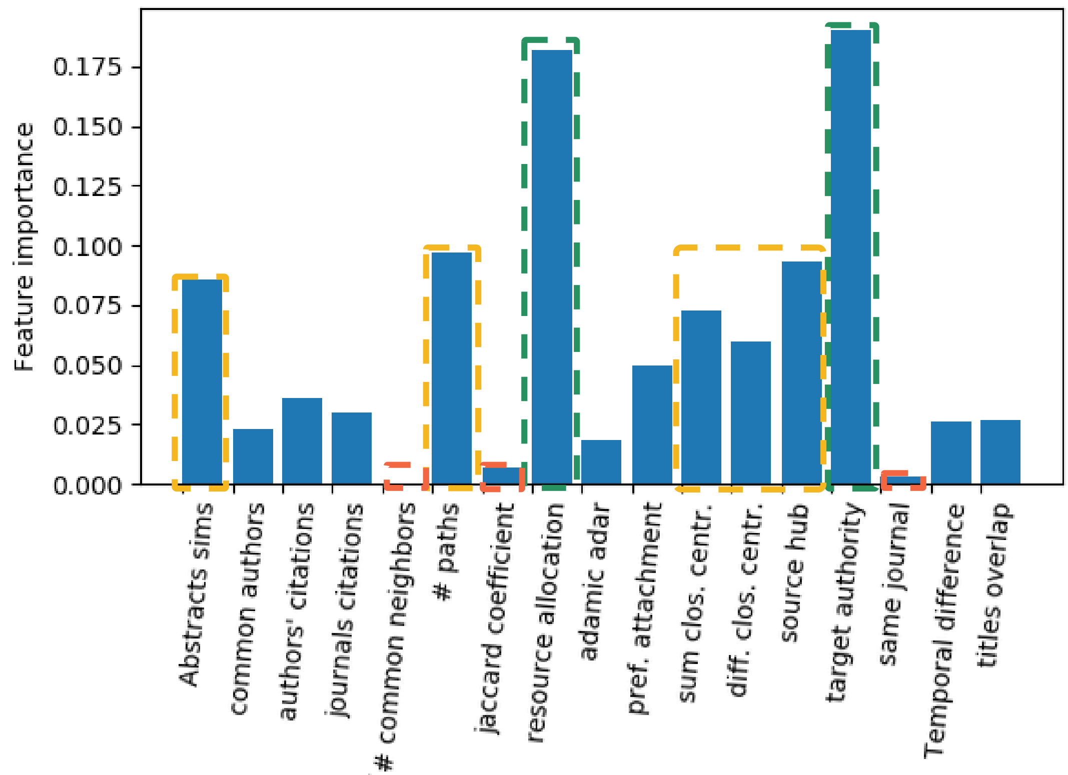
1 2 3 **4** 5 6 7

The Importance of the Implemented Features

Target Authority and Resource Allocation as Most Predictive Features

- The different features have been tested for their predictive power in order to find the best combination
- The features can be grouped into four different Tiers according to their importance
- **Tier 1:**
  - Target Authority
  - Resource Allocation
- **Tier 2:**
  - Abstract Similarity
  - Number of Paths
  - Source Hub
  - Closeness Centralities
- **Tier 3**
  - Preferential Attachment
  - Authors' Citations
  - Journal Citations
  - Common Authors
  - Temporal Difference
  - Titles Overlap
- **Tier 4**
  - Rest

Feature Importance



- ➢ Identification of **Target Authority** and **Resource Allocation** as **most predictive features with** the **highest impact**
- ➢ The **Number of Common Neighbors, Same Journal** and **Jaccard Coefficient** as **least important** features
- ➢ **Some features** are **controversial** $\Rightarrow$ **Trial & Error** (ex. # Common Neighbors and Jaccard Coef.)

ÉCOLE POLYTECHNIQUE UNIVERSITÉ PARIS-SACLAY

Main Classifiers

With the Goal to Achieve the Best Results, Various Models Have Been Tested

- Four main classifiers have been tested in order to achieve the highest prediction
- As the simplest classifier, linear regression was chosen to compare with the other models
- The random forest model was used as practical performance measurement tool for newly implemented features as its training is fast and reliable
- XGBoost was found to be the strongest classifier

➢ **XGBoost** used **as main model**

➢ **Random forest** for **performance measurement** of newly implemented **features**

Linear Regression

- Assumes linear relationship between dependent and independent variables
- Simplest classifier

Random Forest

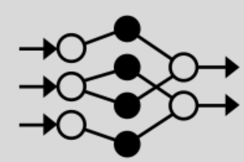
- Ensemble algorithm using the bagging technique as well as decision trees
- Very robust against overfitting
- Easy to tune

XGBoost

XGBoost

- Based on gradient boosted decision trees
- Boosting is applied on a trees ensemble with an incremental policy
- Used as main model

Neural Network



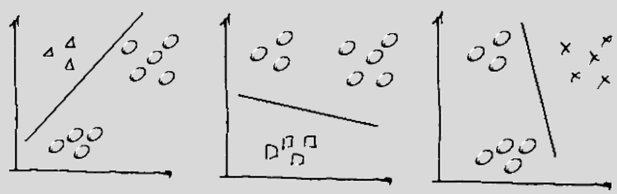
- Best known for recognition tasks as they have multiple layers allowing to learn multiple layers of abstraction
- High potential due to hyperparameters used to tune the model

- Further classifiers have been used with standard parameters as baselines
- Mainly served as references in order to compare and evaluate the results of the main classifiers

Further Classifiers

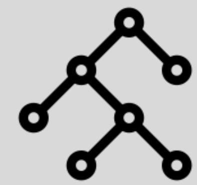
With the Goal to Achieve the Best Results, Various Models Have Been Tested

One Versus Rest (with Lin. Reg.)



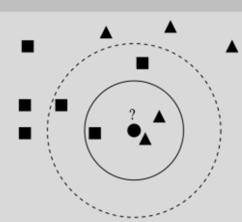
- Involves training a single classifier per class, with the samples of that class as positive and all the others as negatives

Decision Tree



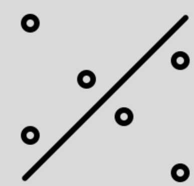
- Deals with dif. parameters and, depending on the response over each parameter, splits the data until a final answer is reached
- However, the more the data is split the higher the risk of overfitting the data

KNeighbors



- One of the simplest classifiers based on the k-nearest neighbors algorithm (KNN)
- Can be used for both classification and regression predictive problems

Support Vector Machine



- A discriminative classifier formally defined by a separating hyperplane
- Given labelled training data, the algorithm outputs an optimal hyperplane categorizing new examples

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Models

Fitting and Tuning of the Models

Fitting of the Models

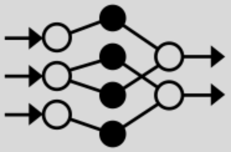
- K-fold cross-validation (usually K = 5)
- Training – Validation split
- Early stopping
- Dropout (Neural Network)

Tuning of the Models

XGBoost

- # estimators
- Maximal depth
- Learning rate

Neural Network

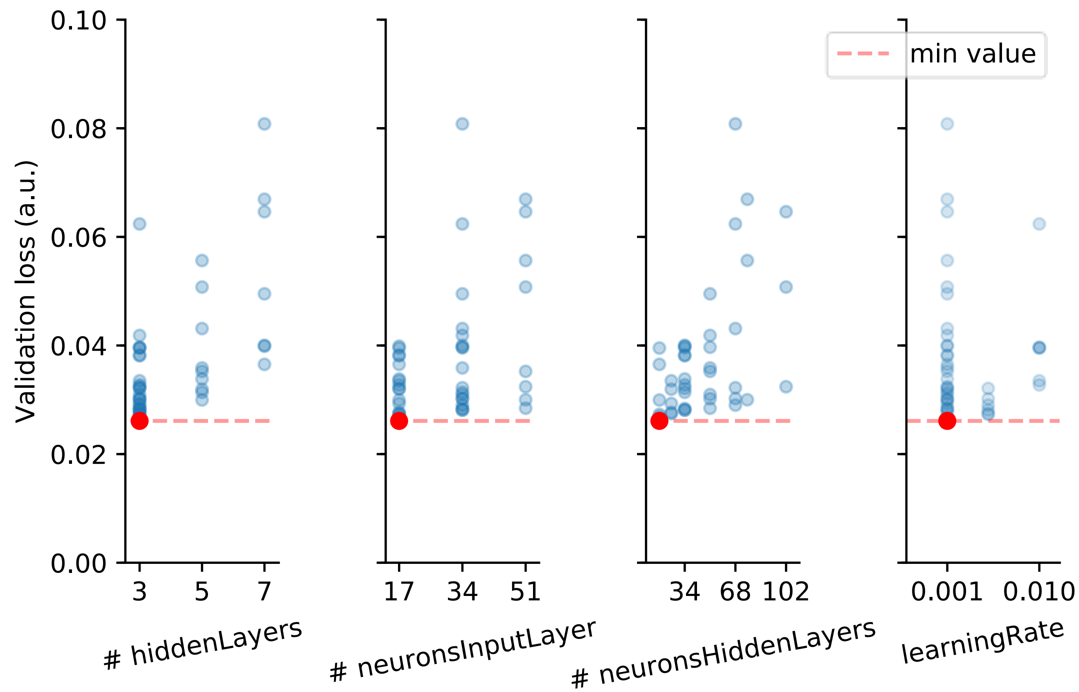


- # hidden layers
- # neurons in hidden layers
- # neurons in input layer
- Learning rate (Adam Optimizer)

Parameter Tuning

Effects During the Tuning Process of the Neural Network

Validation Loss Value of Models During Tuning Process

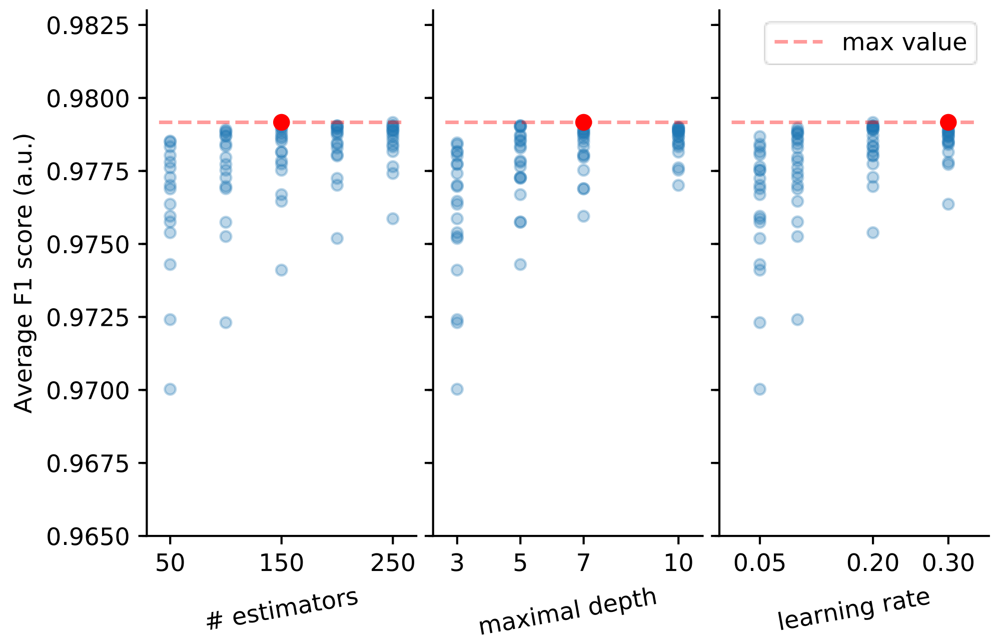


- **Higher validation loss by increasing number of hidden layers in the Neural Network**
- **Lowest validation loss while number of neurons per layer being low**
- **Learning rate of 0.001 works best**

Parameter Tuning

Effects During the Tuning Process of the XGBoost Classifier

Validation Loss Value of Models During Tuning Process



- **Higher F1 scores by increasing the *learning rate***
- **The other optimal parameters: *# estimators = 150* and *max_depth* = 7**

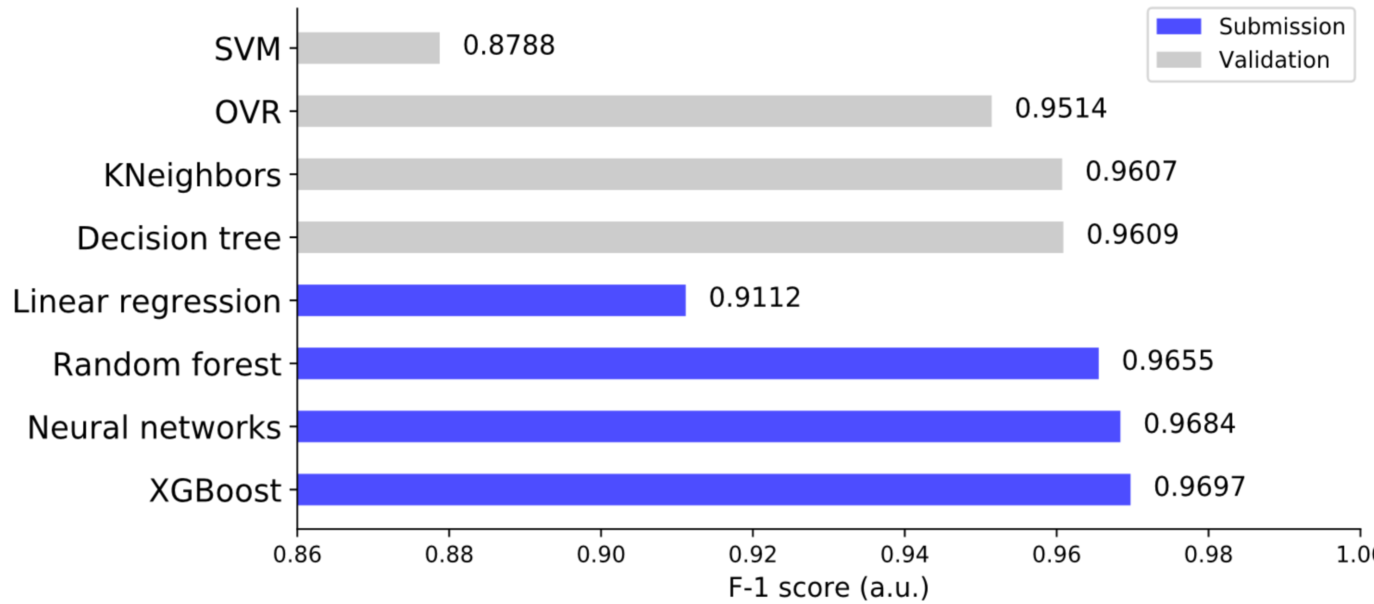
ÉCOLE POLYTECHNIQUE UNIVERSITÉ PARIS-SACLAY

1 2 3 4 5 **6** 7

Results

Comparison of the Performance of Different Classifiers

Performance Comparison of Various Classifiers



- The **Boosted Tree model** (XGBoost) yielded the **best results with** a maximal **F-1 Score** of almost **0.97**
- **Random Forest** as well as the **Neural Network** achieved **almost as good results**

1 2 3 4 5 6 7

Results

The XGBoost Classifier Achieved the Best Results With the Highest Prediction Capabilities

Linear Regression

✓ **F-1 Score** (on test dataset)**:** 0.911154

- The coefficient of determination (R^2) has been around 0.57
- This indicates that the linear regression model can explain approx. 57% of the variance

➢ **Linear Regression** is **not suitable for** the accurate **reconstruction of** the **initial citation network**

Random Forest

✓ **F-1 Score** (on test dataset)**:** 0.96544

- Parameters were tuned by varying the number of estimators (decision trees) and maximum depth of each tree
- F-1 score improved by increasing maximal depth until the value reached 16. The parameter was set to 8 to avoid overfitting

➢ **Random Forest** achieved the **third best** result **with number of estimators** being **40** and **maximal depth** being **8**

XGBoost

✓ **F-1 Score** (on test dataset)**:** 0.96973

- Best result was achieved with 20% of the training set data being used as validation set
- The early stopping parameter was set to 20 rounds to avoid overfitting; for the other parameters the default values were left untouched

➢ **XGBoost** classifier as **best model for** the **prediction of missing links** in a citation network

Neural Network

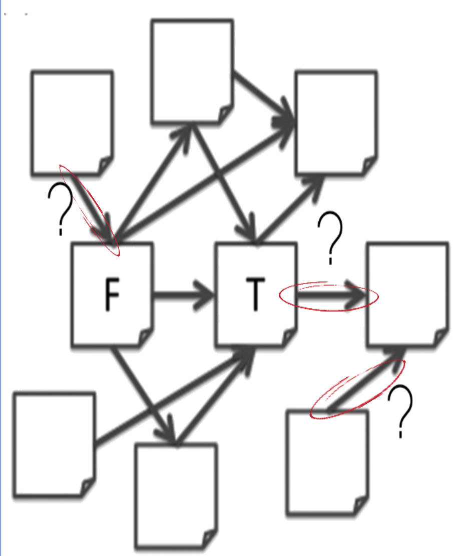
✓ **F-1 Score** (on test dataset)**:** 0.96839

- Several attempts to tune neural networks by varying the number of hidden layers, neurons at each layer as well as the learning rate
- Usage of several techniques to avoid overfitting (e.g. simple validation splits with early stopping and dropout)

➢ **Neural Network** yielded **almost as good** results **as XGBoost**
➢ Slightly **better** results **without overfitting-avoidance**

Conclusion

The XGBoost Classifier Together With Selected Features is the Most Suitable Model



Prediction of Missing Links in Citation Networks

1	▪ The Boosted Tree model (XGBoost) yielded the best results with an F-1 Score of approx. 0.97, Random Forest and Neural Network achieved nearly as good results
2	▪ *Target Authority score* and *Resource Allocation Index* had the highest impact on the prediction of the XGB Classifier while *Number of Common Neighbors*, *Jaccard Coefficient and Same Journal* were the features with the least impact.
3	• Features based on properties of the graph were in general more important than features based on properties of the papers.
4	▪ Parameter tuning improved the Neural Network model. ▪ The XGBoost model could not be improved by varying the parameters.