Chair of Communication Networks
Department of Electrical and Computer Engineering
Technical University of Munich

Boost your Communication Network with Machine Learning

Prof. Dr.-Ing. Wolfgang Kellerer

wolfgang.kellerer@tum.de

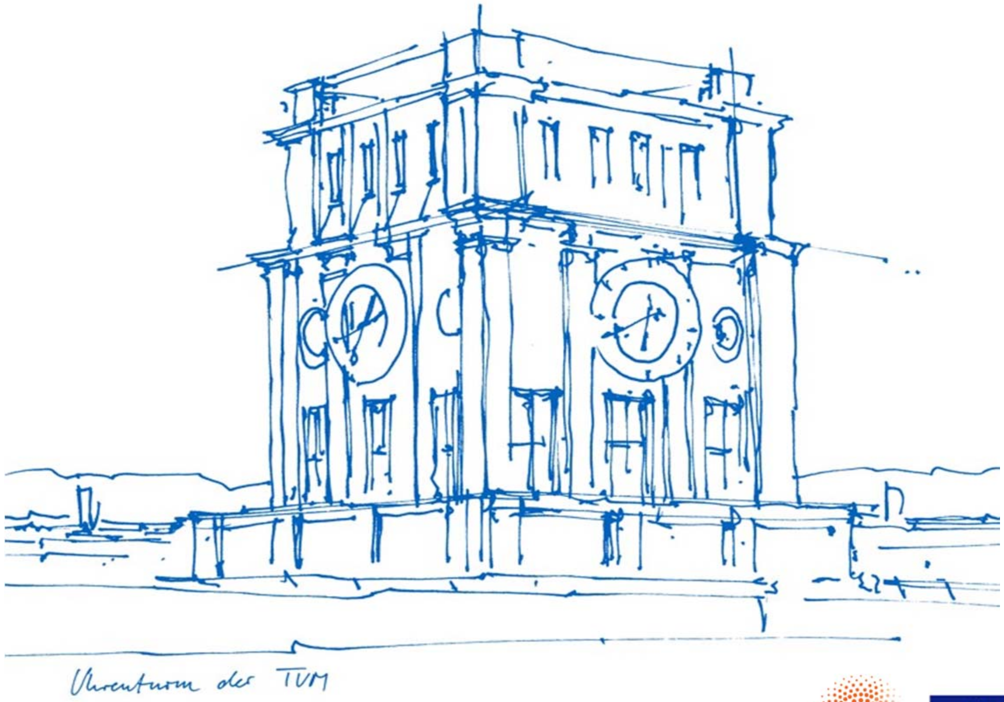
based on work done together with

Dr. Andreas Blenk

Patrick Kalmbach

Johannes Zerwas

and several others



Bundesministerium für Bildung und Forschung

SENDATE PLANETS

Project ID 16KIS0473

This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement No 647158 – **FlexNets (2015 – 2020)**.

European Research Council

Machine Learning (ML) in Communication Networking

receives a lot of attention recently, e.g.

- ML for flow classification and anomaly detection
- ML replacing optimization for virtual network embedding, function placement,…

in this talk, we show another application

- **ML to preprocess models leaving existing algorithms or optimizers untouched**

Boost your network algorithm with ML preprocessing

- ***Neurovine****: Hopfield neural network to preprocess (subgraph extraction) VNE algorithms*
- ***o‘zapft is****: supervised learning to learn from previous solutions of network algorithms*

This talk is mainly based on our following work

- Andreas Blenk, Patrick Kalmbach, Johannes Zerwas, Michael Jarschel, Stefan Schmid, Wolfgang Kellerer: ***NeuroViNE: A Neural Preprocessor for Your Virtual Network Embedding Algorithm*** IEEE INFOCOM 2018 (main conference), Honolulu, HI, USA, April 15-19, 2018.

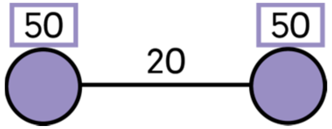
- Blenk, Andreas; Kalmbach, Patrick; Schmid, Stefan; Kellerer, Wolfgang: ***o'zapft is: Tap Your Network Algorithm's Big Data!*** ACM SIGCOMM 2017 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (Big-DAMA), 2017.

- Patrick Kalmbach, Andreas Blenk, Markus Klügel, Wolfgang Kellerer: ***Stochastic Block Models for Analysis and Synthetic Generation of Communication Networks*** 2nd IFIP/IEEE International Workshop on Analytics for Network and Service Management (AnNet), 2017.

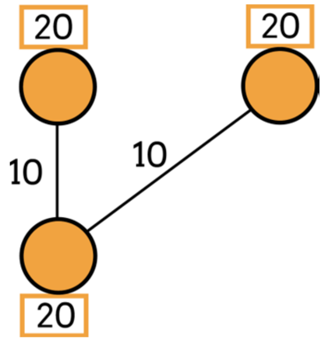
… and the Dissertation of Dr. Andreas Blenk: Towards Virtualization of Software-Defined Networks: Analysis, Modeling, and Optimization (defended March 2, 2018)

Our Use Case: Virtual Network Embedding (VNE)

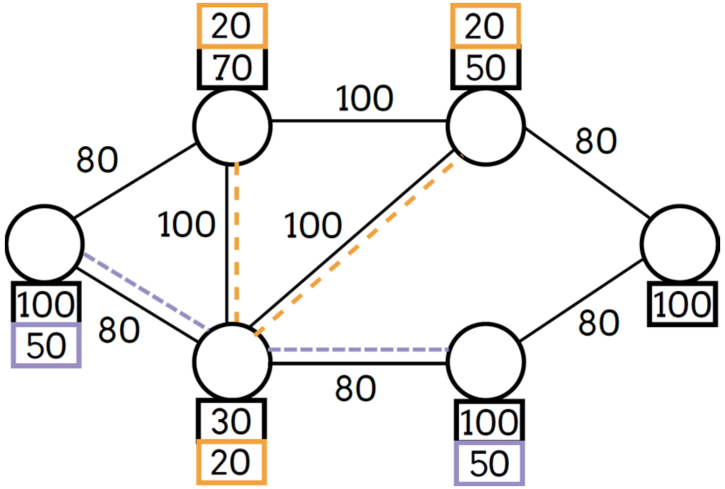
VN Request 1



VN Request 2



Substrate Network & Embedded Requests



NP-hard!

VN Request (VNR) requires node and link resources

Capacities of the substrate nodes and links are limited

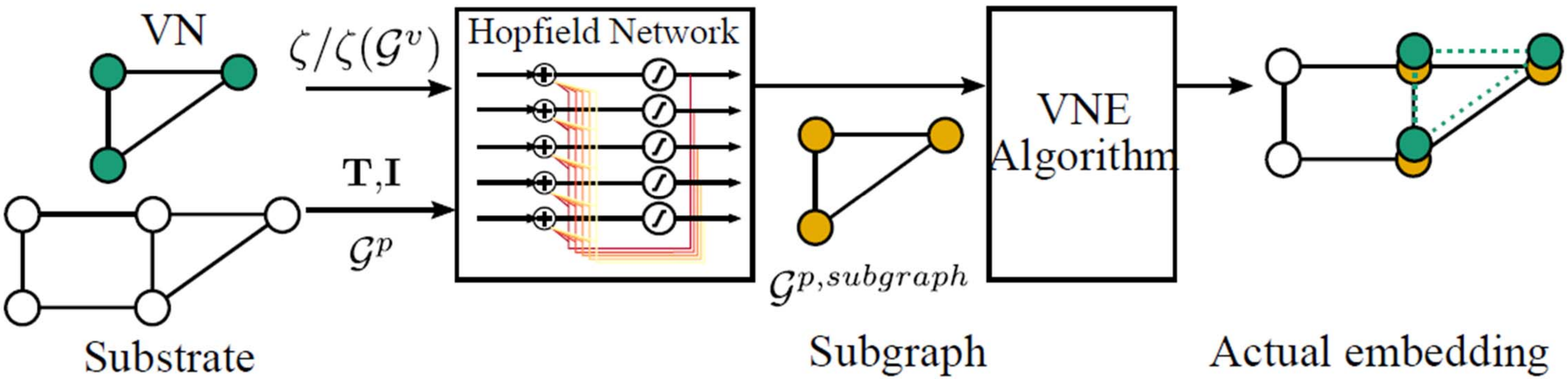
- Challenge: Regularly solving computation hard problems
- Goal: Speed-up and/or improve Virtual Network Embedding

Neurovine:

Hopfield neural network to preprocess (subgraph extraction) VNE algorithms

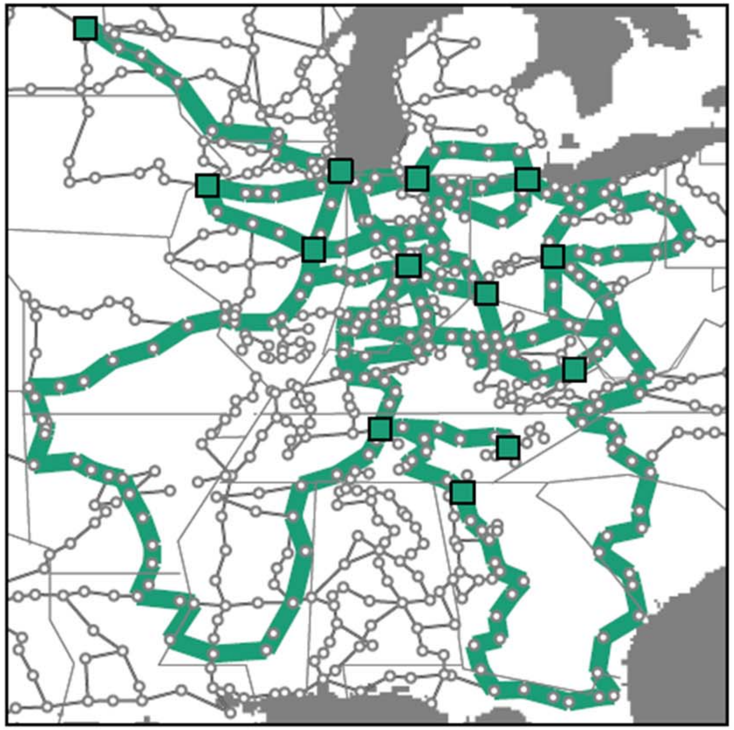
Andreas Blenk, Patrick Kalmbach, Johannes Zerwas, Michael Jarschel, Stefan Schmid, Wolfgang Kellerer:
NeuroViNE: A Neural Preprocessor for Your Virtual Network Embedding Algorithm
IEEE INFOCOM 2018 (main conference), Honolulu, HI, USA, April 15-19, 2018.

Neural Preprocessor for Virtual Network Embedding: NeuroViNE

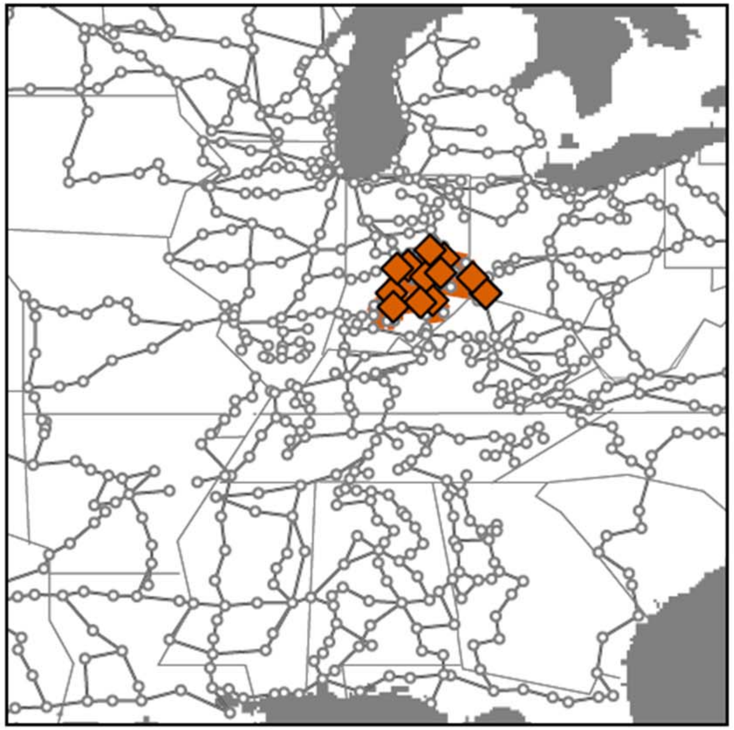


- State-of-the-art: Heuristics judge nodes independently from each other

Heuristics judge nodes independently from each other



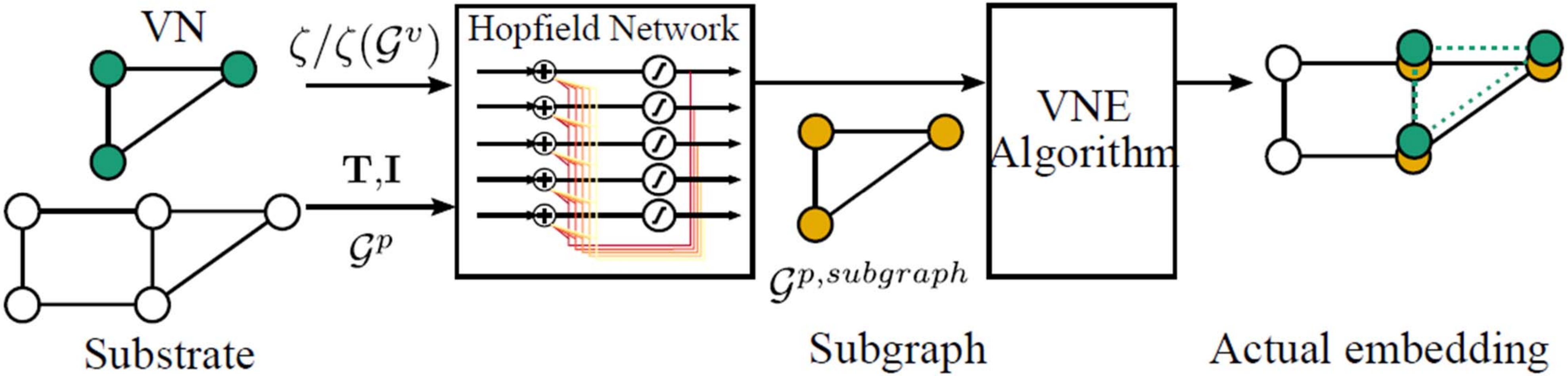
GRC



Our proposal: HF-GRC

Example: Comparison of node locations for a single VNR between GRC and HF-GRC

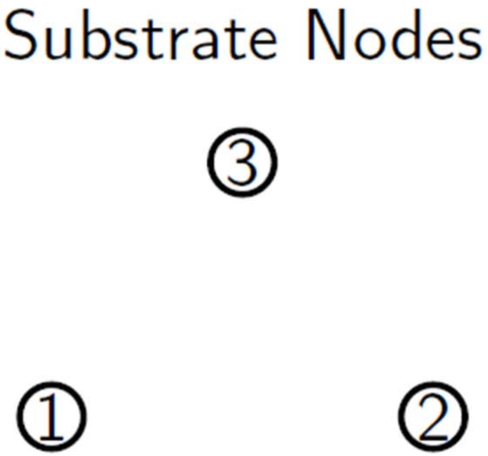
Neural Preprocessor for Virtual Network Embedding: NeuroViNE

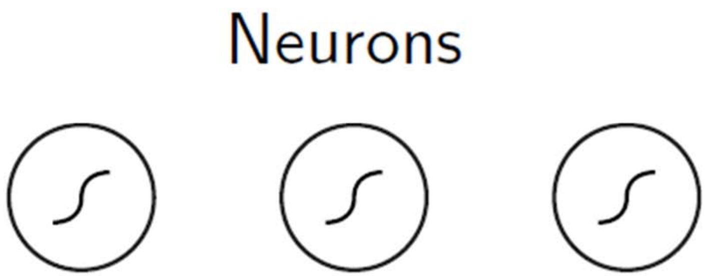


- State-of-the-art: Heuristics judge nodes independently from each other
- Idea: Extract subgraph with physical nodes close to each other and high available capacities

Optimization with Hopfield Neural Networks

Graph Inputs → Neural Network → Solution

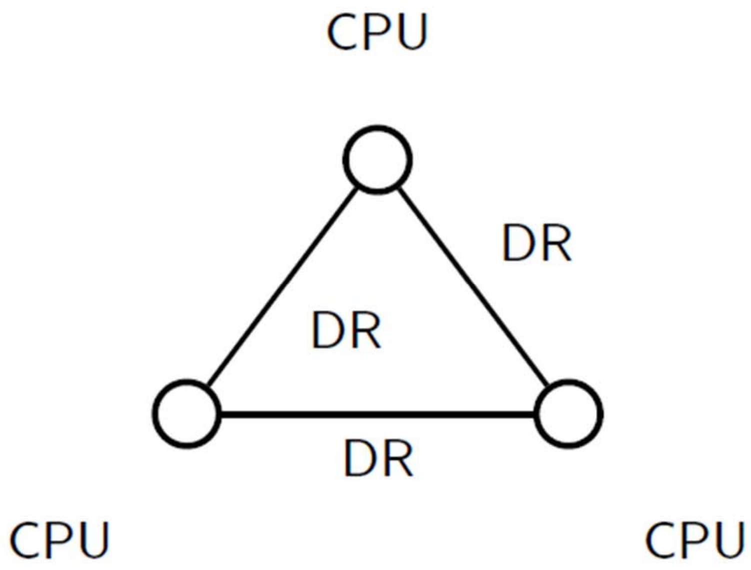


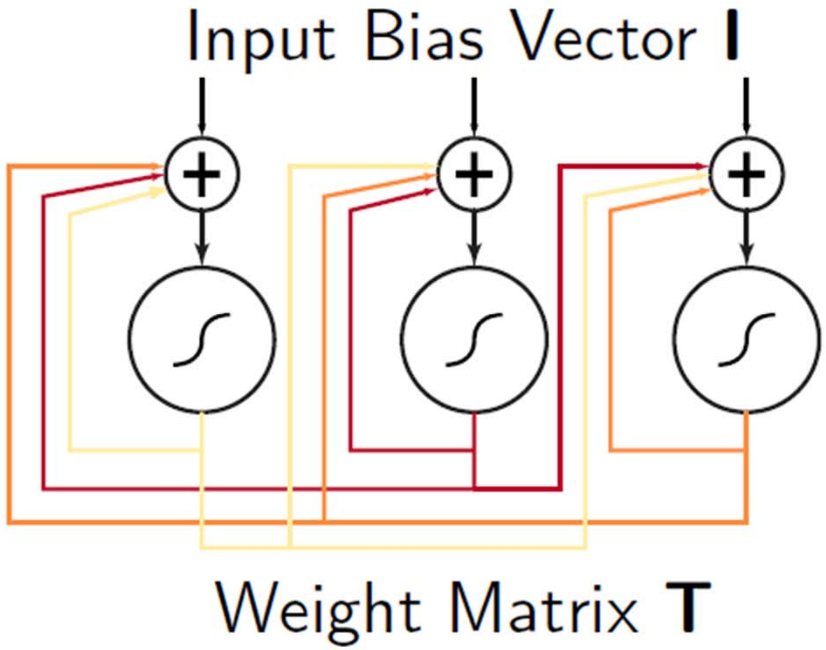


- According to optimization problem

Optimization with Hopfield Neural Networks

Graph Inputs → Neural Network → Solution

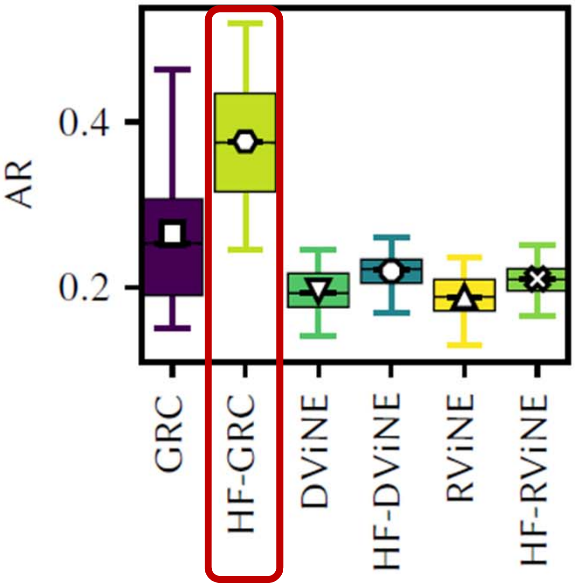




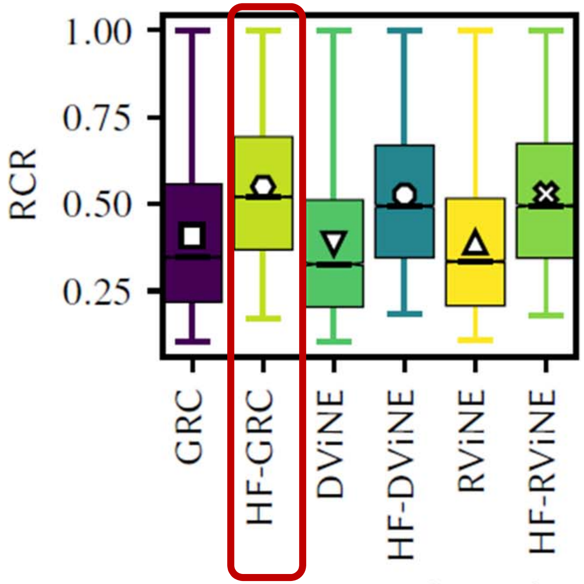
- According to optimization problem:
  - Input Bias Vector **I** integrates available node capacities (CPU)
  - Weight Matrix **T** integrates available datarate capacities (DR)
  - **I** and **T** take care of number of selected nodes (ζ) [1]
- Executing means solving: $\frac{d\mathbf{U}(t)}{dt} = -\frac{\mathbf{U}(t)}{\tau_{HF}} + \mathbf{TV}(t) + \mathbf{I}$

[1] G Tagliarini, J Christ, and E Page. "Optimization using neural networks". In: IEEE Trans. Comp. 40.12 (Dec. 1991), pp. 1347-1358.

Efficiency on Real Network Topologies



Acceptance Ratio (AR)



Revenue-cost-ratio (RCR)

- VNE algorithms (GRC, DViNE, RViNE) vs. Hopfield variants (HF-GRC, HF-DViNE, HF-RViNE)
- NeuroViNE accepts more networks with less costs

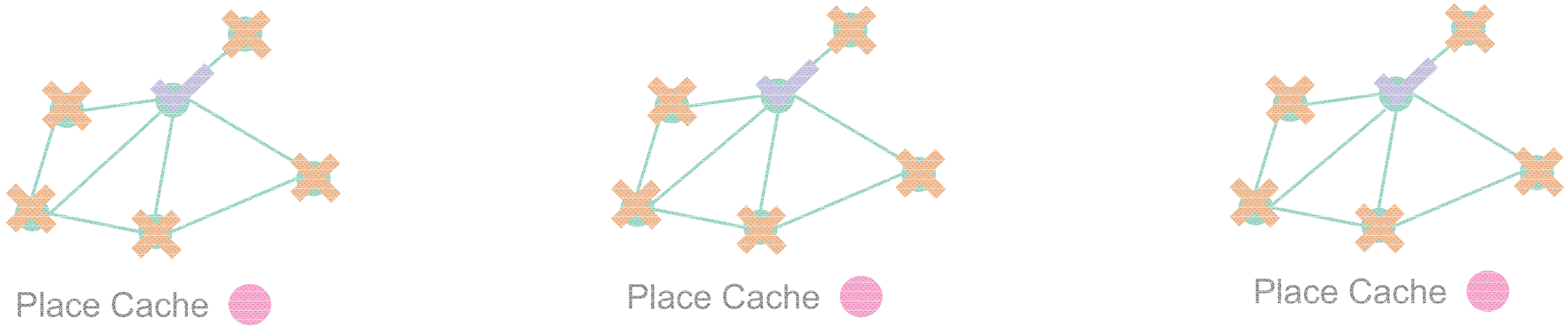
another way of filtering:
Data-driven Networking

o‘zapft is:

supervised learning
to learn from previous solutions (the data)
of (general) network algorithms

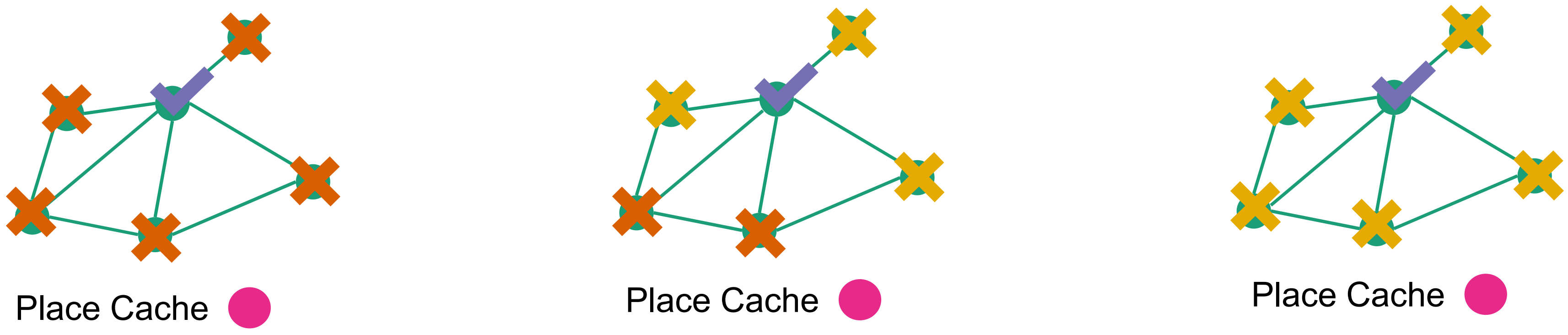
Blenk, Andreas; Kalmbach, Patrick; Schmid, Stefan; Kellerer, Wolfgang:
o'zapft is: Tap Your Network Algorithm's Big Data!
ACM SIGCOMM 2017 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (Big-DAMA), 2017.

The Limitation of today's network algorithms: *Fire and Forget*

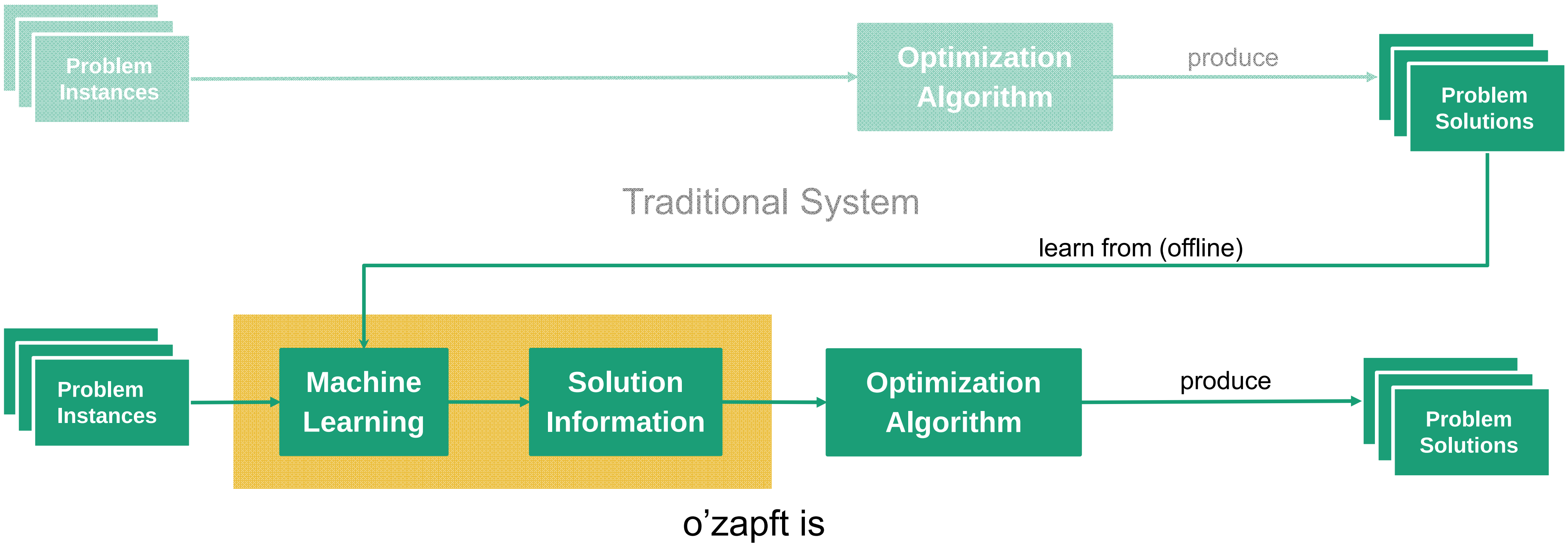


Algorithms repeatedly solve similar problems **from scratch.** This is not only boring for the algorithm but also a waste of information and resources

The Opportunity – Tap into your Algorithm's Big Data



Traditional vs. Proposed System

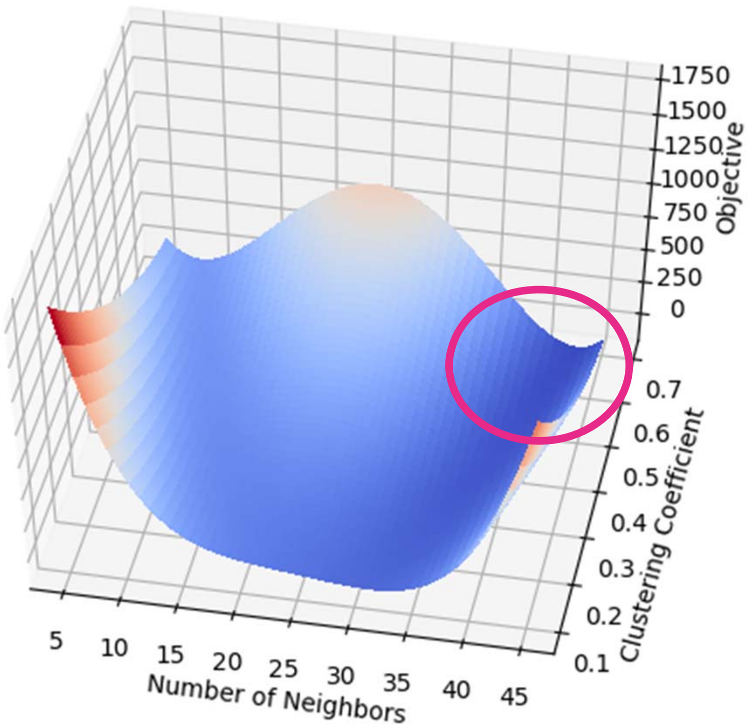


- State-of-the-art: Neglects produced data!
- Idea: Use problem/solution data generated by algorithms regularly solving problems

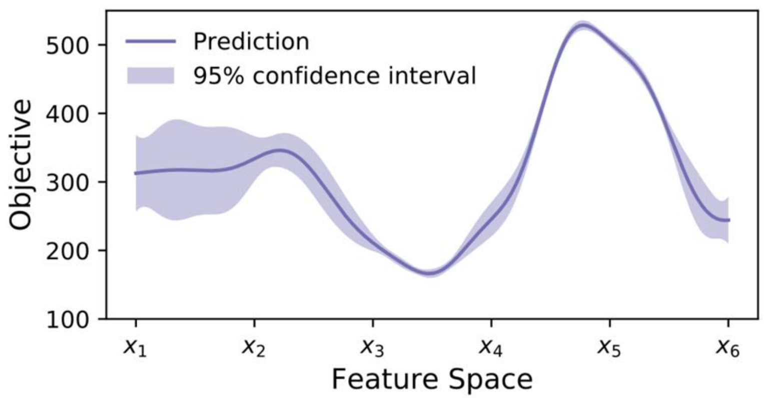
Data Available at: Patrick Kalmbach, Johannes Zerwas, Michael Manhart, Andreas Blenk, Stefan Schmid, and Wolfgang Kellerer. 2017. Data on "o’zapft is: Tap Your Network Algorithm’s Big Data!". (2017). https://doi.org/10.14459/2017md1361589

Potentials

- Potentials: (a) Reduce search space and (b) Predict problem outcome



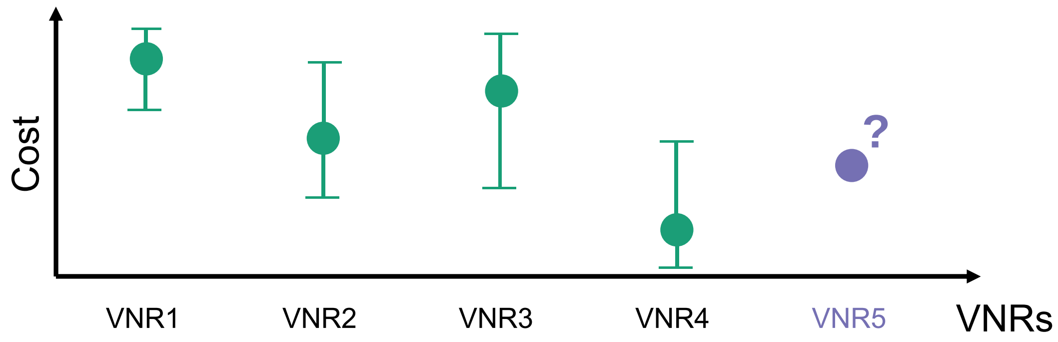
(a) Search Space Reduction/Initial Solutions



(b) Predict Value of Objective Function
→ admission control

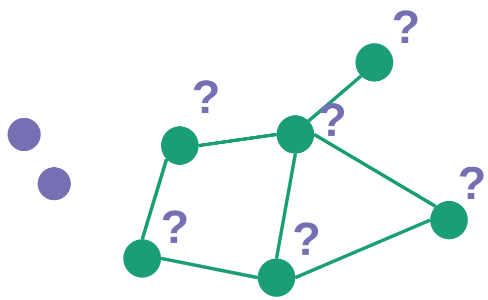
Use Cases

Virtual Network Embedding (VNE) – Predict Embedding Costs



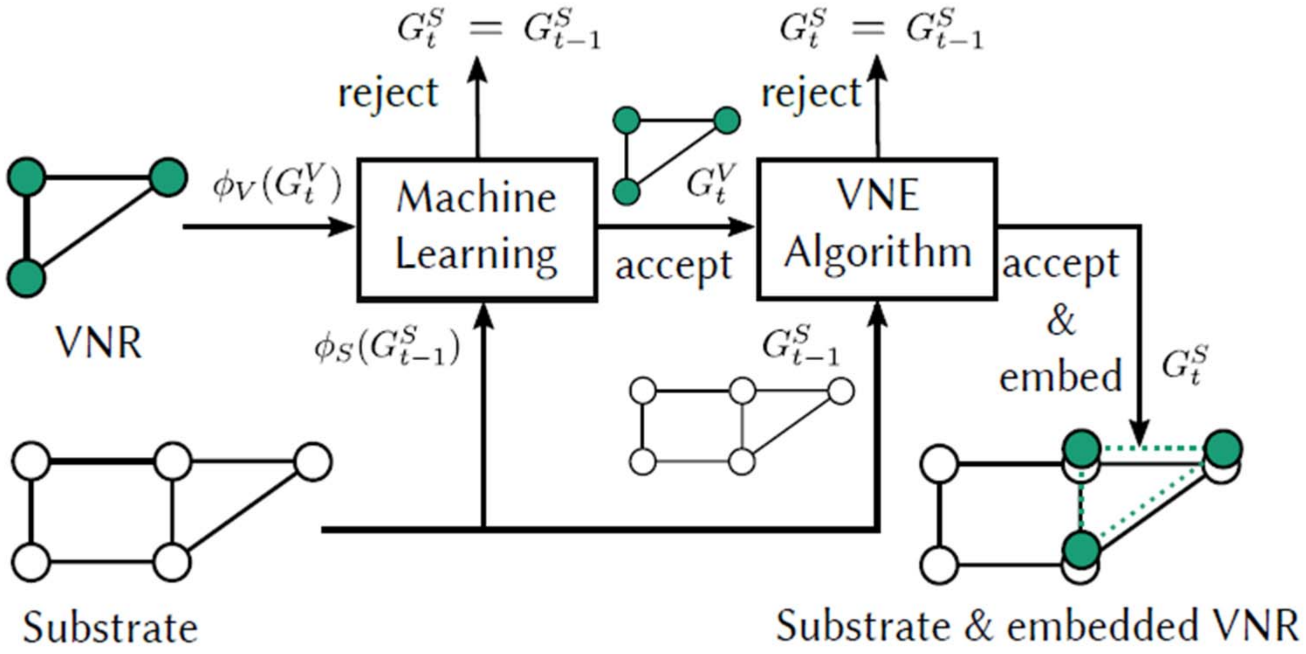
Problem: Given a VNR and Substrate, what will the cost be?

Facility Location (Controller Placement) – Guess Initial solutions



Problem: Given a network and a number of controllers, where to place the controllers?

Case Study I: Virtual Network Embedding Cost

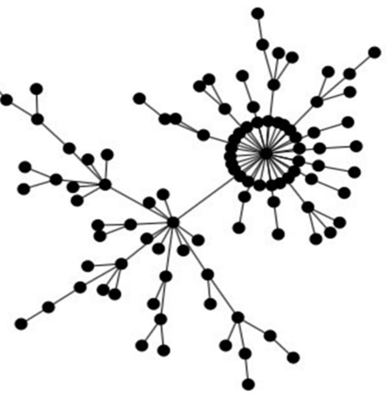


- Learn and predict the embedding cost of a VNR
- Embedding cost = total length of the virtual links interconnecting the requested virtual nodes
- Supervised learning: regressors predict the cost of to-be-embedded virtual networks
- Offline training!

Methodology

Optimization Algorithms

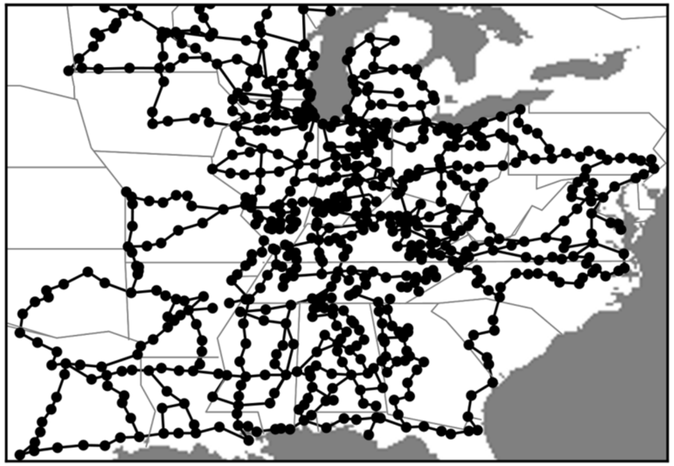
- Greedy [20]
- GRC (Global Resource Capacity) [8]
- SDP (optimal, Mixed Integer Program (MIP))
- Strawman (SM) (“VNR#nodes&links → cost”)



Barabasi-Albert Graph [3]

Substrates

- Erdős-Rényi (ER) 40 nodes
- Barabasi-Albert (BA) [2] 40 nodes
- Topology Zoo [1]: Kentucky Data Link (KDL) 734 nodes
- 6-ary Fat Tree (DC-FT)
- BCube2 (DC-BC)



Kentucky Data Link

Objective:

Minimize embedding cost

5 runs with 2500 VRs of each combination

[1] Knight et al., The Internet Topology Zoo. *IEEE J. on Sel. Areas in Communications 29*, 9 (2011)
[2] Saino et al., A Toolchain for Simplifying Network Simulation Setup, in *Procs. SIMUTOOLS '13*, Cannes, France, March 2013
[3] Picture taken from http://graphstream-project.org/media/img/generator_overview_barabasi_albert.png
[8] Long Gong, Yonggang Wen, Zuqing Zhu, and Tony Lee. 2014. Toward profitseeking virtual network embedding algorithm via global resource capacity. IEEE INFOCOM 2014
[20] Minlan Yu, Yung Yi, Jennifer Rexford, and Mung Chiang. 2008. Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration. SIGCOMM CCR 38, 2 (3/2008)

Learning embedding cost

Library:

- Sci-Kit Learn [1]

Graph features:

- Node degree
- Closeness

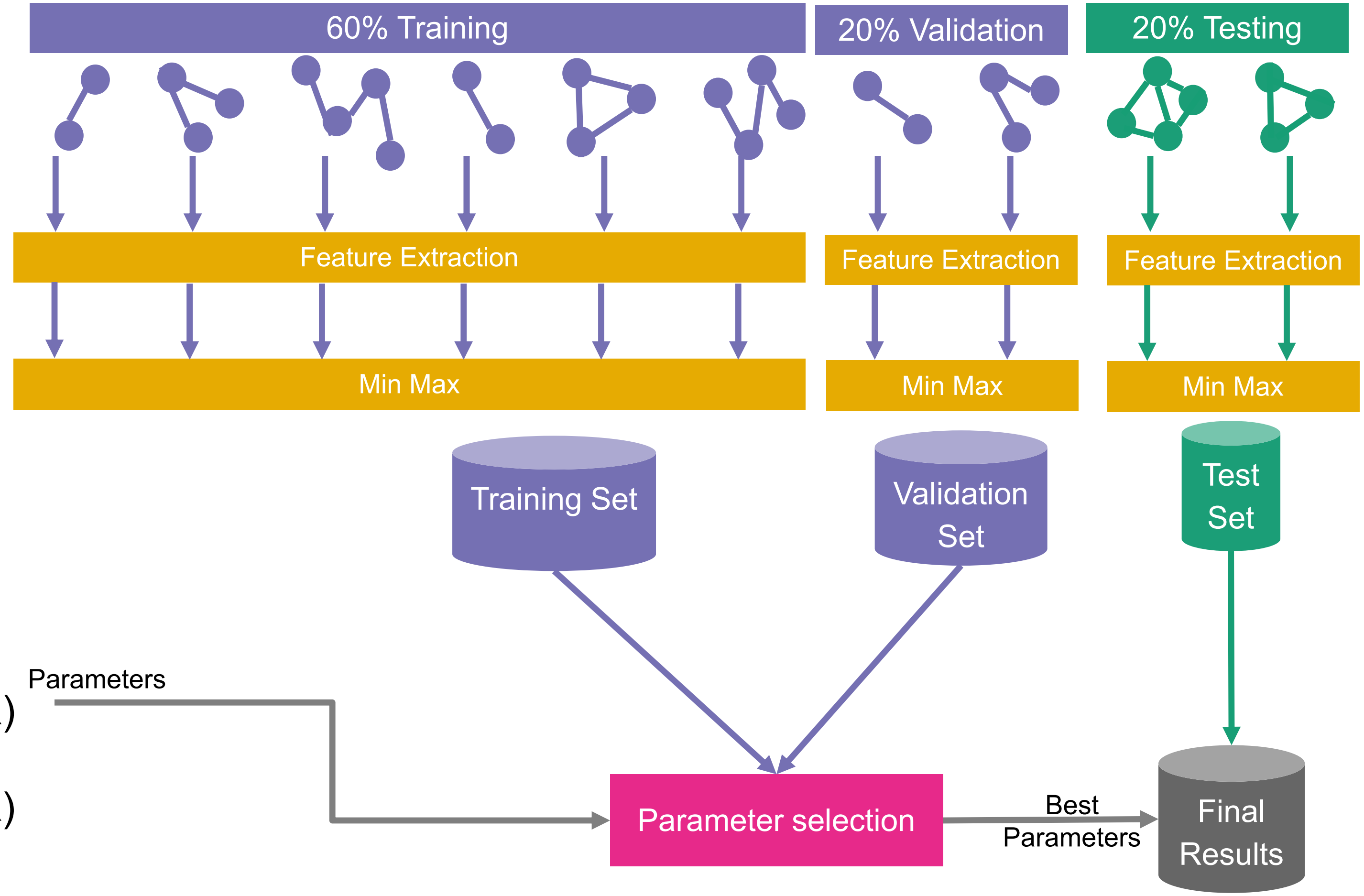
Measures:

- R^2 (goodness of fit for ML models)

Classifier:

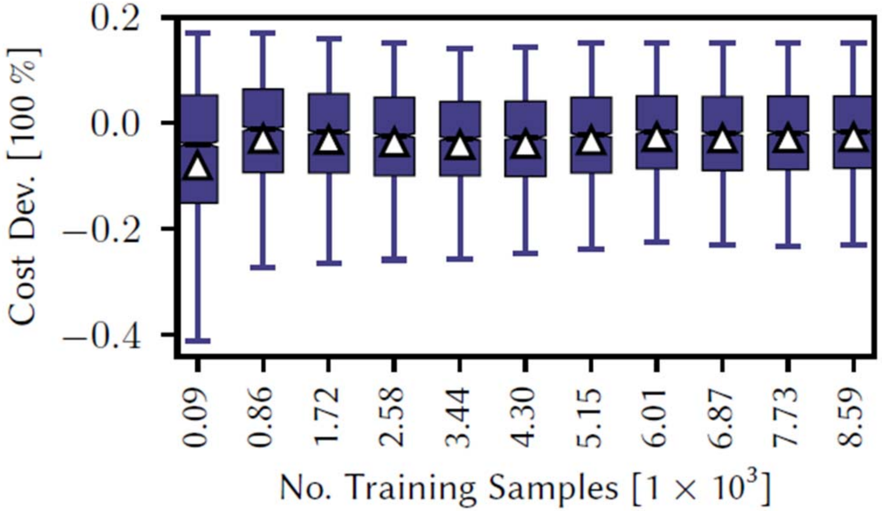
- Linear Regression (LR)
- Bayesian Ridge Regressor (BRR)
- Random Forest Regressor (RF)
- Support Vector Regression (SVR)

Model Training and Selection:

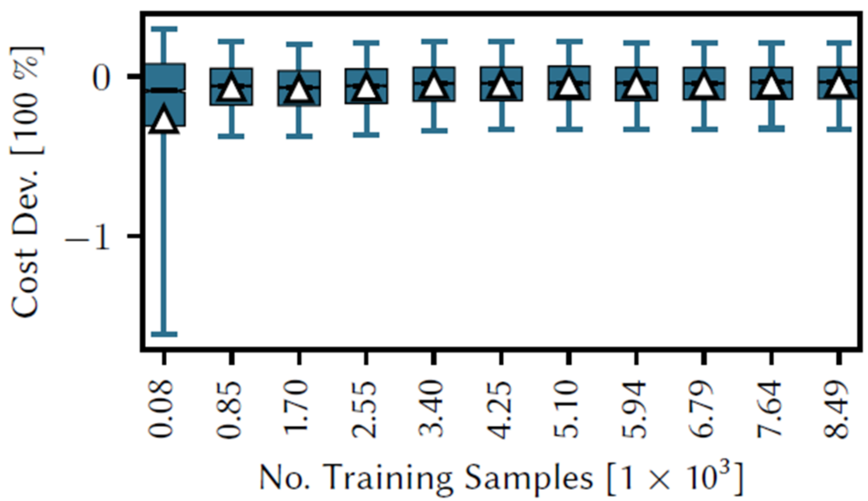


[1] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

How much learning is required?



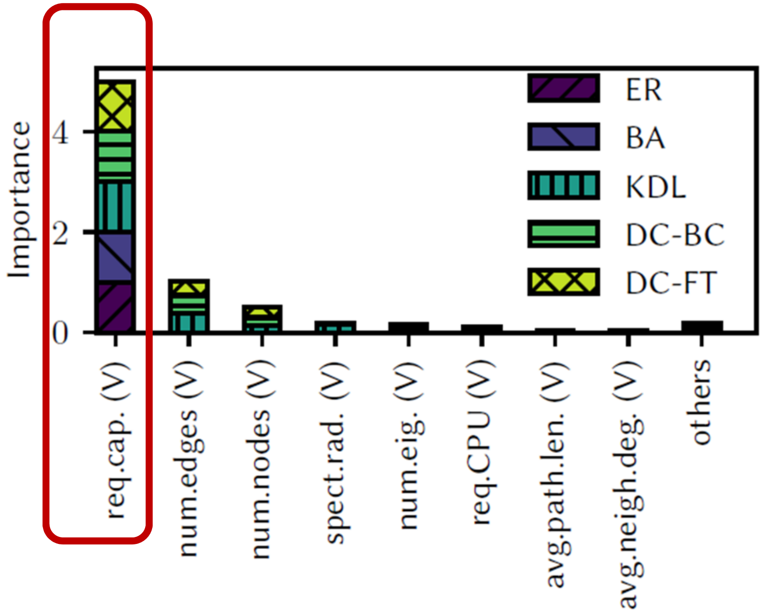
Greedy



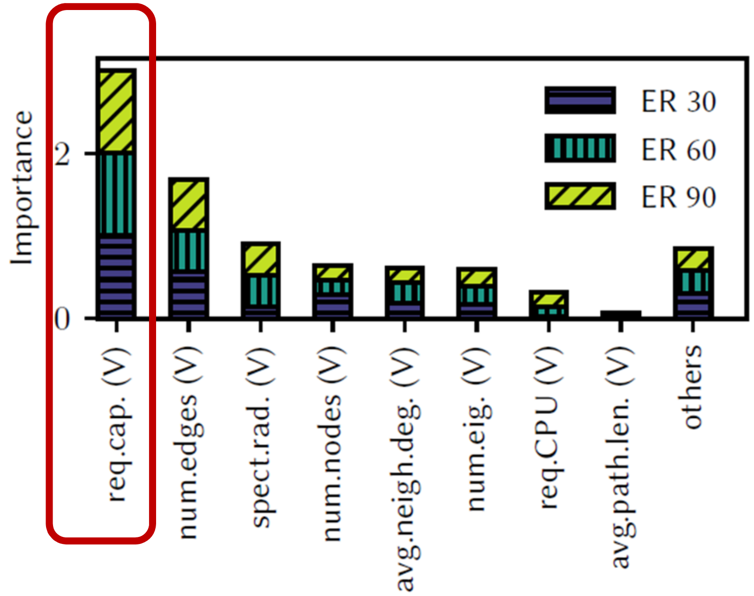
SDP

- VNR embedding costs can be estimated well after a short training period

Which graph features are important for solution learning?



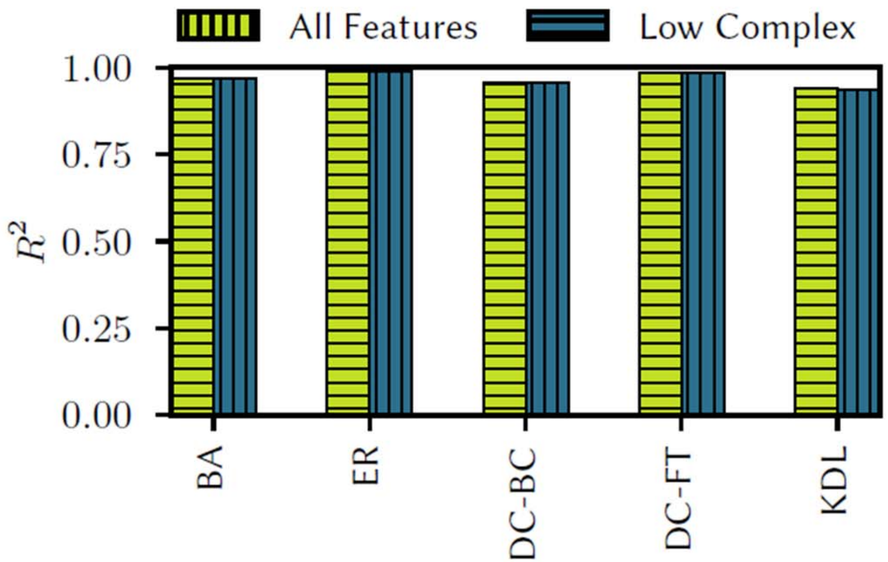
Greedy



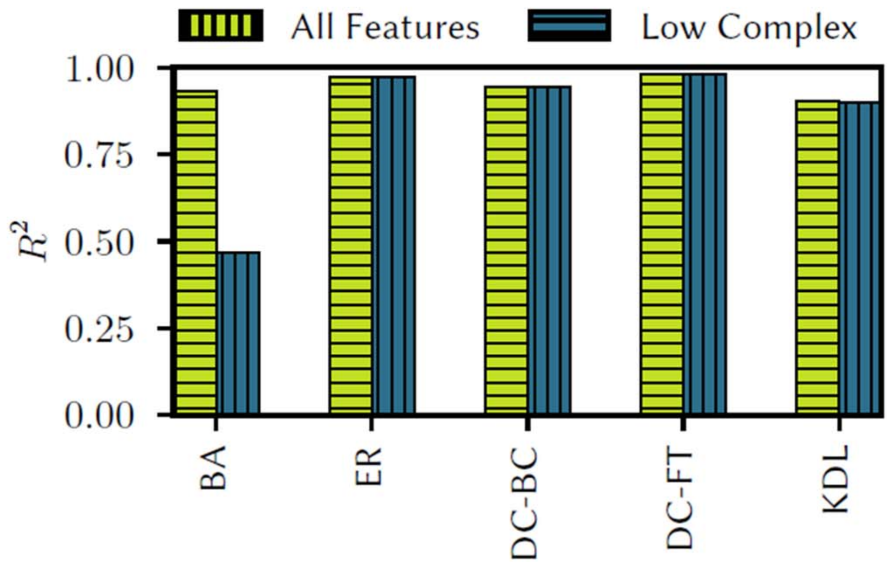
SDP

- Requested Link Capacity is most important
- For SDP the importance is more distributed (larger search space and variation of solutions)

Is one feature enough?



Greedy

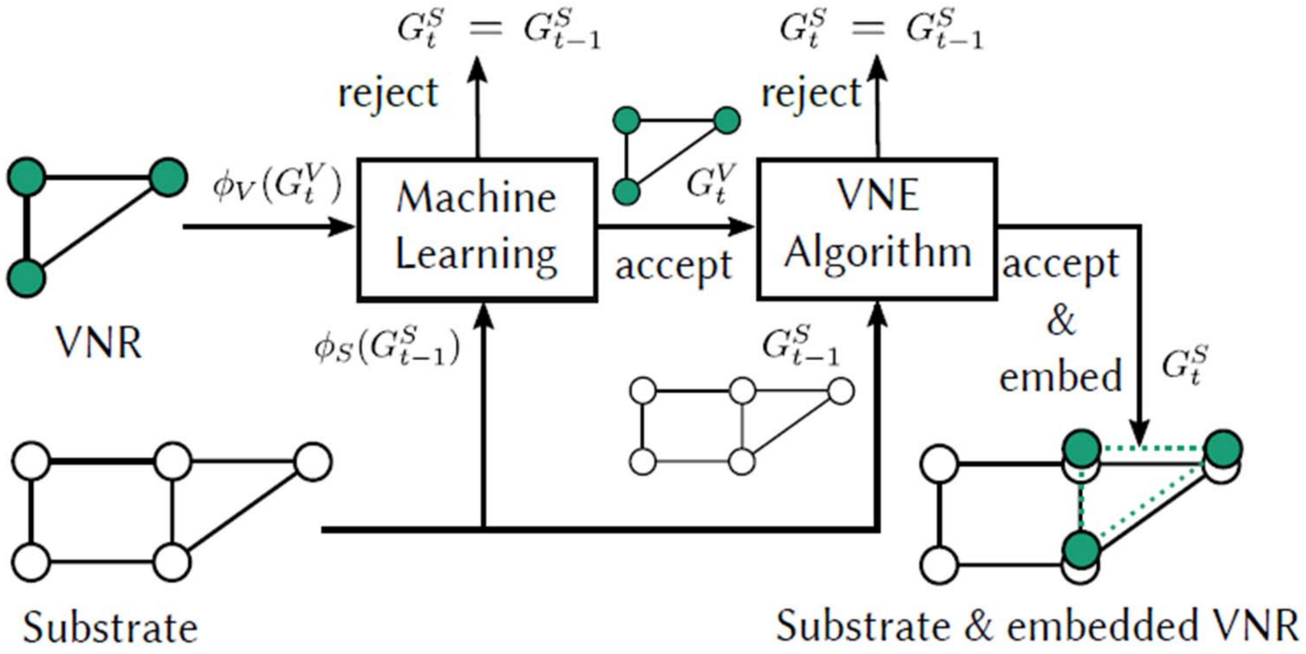


GRC

- Trained regressors with features of different complexity: O(n), O(n+m), O(n * log n) [14]
- Already low complexity features provide a high R^2 (goodness of fit for ML models)

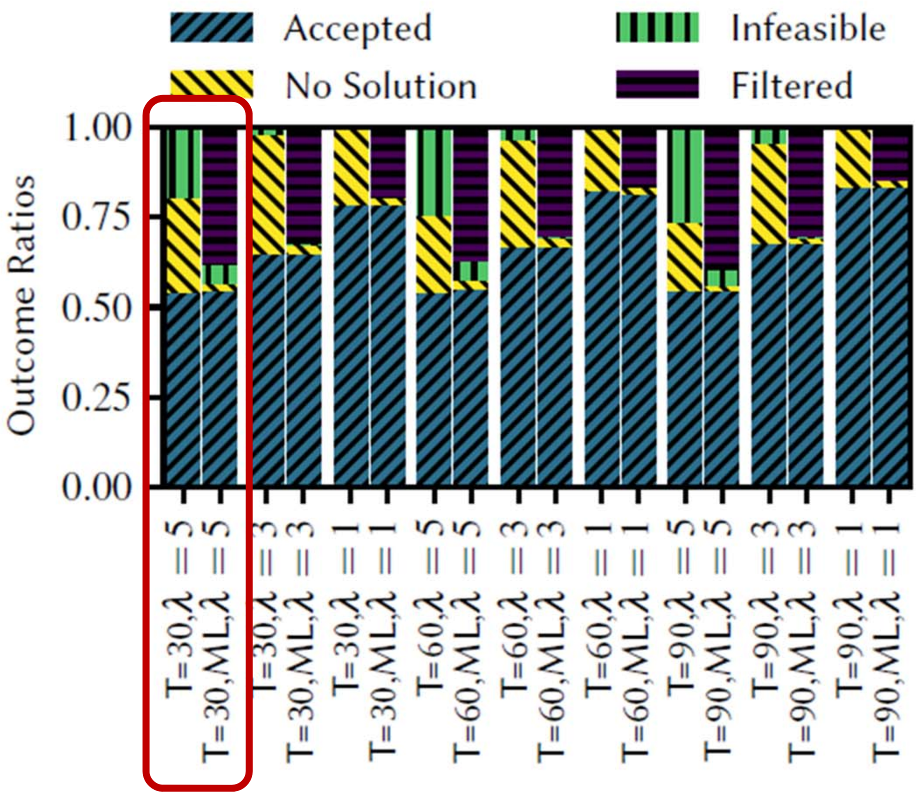
[14] Geng Li, Murat Semerci, Bülent Yener, and Mohammed J Zaki. 2012. Effective graph classification based on topological and label attributes. Statistical Analysis and Data Mining 5, 4 (Aug. 2012), 265–283.

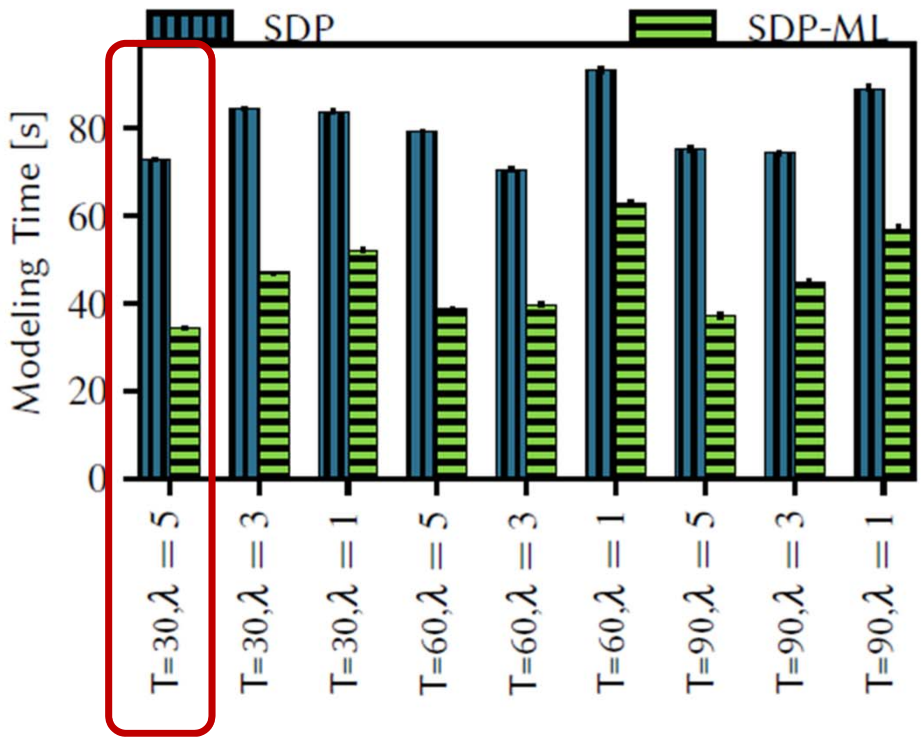
Case Study II: Predicting Acceptance Probabilities of VNE Requests



- Supervised learning: use data with accepted and rejected requests ! Offline training!
- Recurrent neural network (RNN) for classification
- Filter Infeasible and requests with unacceptable algorithm runtime (No Solution)

Can we speed-up optimal algorithms using admission control?





- Efficient Filtering of infeasible and unacceptable requests
- Efficient saving of model creation time
- Saving up to 50% computational resources

Conclusion

Machine Learning can be successfully used to **preprocess** models leaving existing algorithms or optimizers **untouched**

Boost your network algorithm with ML preprocessing – Tap your data!

- ***Neurovine**: Hopfield neural network to preprocess (subgraph extraction) VNE algorithms*
  - *tailored filtering*

- ***o'zapft is**: supervised learning to learn from previous solutions of network algorithms*
  - ***data-driven networking algorithms***

Important References

- Andreas Blenk, Patrick Kalmbach, Johannes Zerwas, Michael Jarschel, Stefan Schmid, Wolfgang Kellerer: ***NeuroViNE: A Neural Preprocessor for Your Virtual Network Embedding Algorithm*** IEEE INFOCOM 2018 (main conference), Honolulu, HI, USA, April 15-19, 2018.
- Blenk, Andreas; Kalmbach, Patrick; Schmid, Stefan; Kellerer, Wolfgang: ***o'zapft is: Tap Your Network Algorithm's Big Data!*** ACM SIGCOMM 2017 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (Big-DAMA), 2017.
- Patrick Kalmbach, Andreas Blenk, Markus Klügel, Wolfgang Kellerer: ***Stochastic Block Models for Analysis and Synthetic Generation of Communication Networks***2nd IFIP/IEEE International Workshop on Analytics for Network and Service Management (AnNet), 2017.
- Andreas Blenk: ***Towards Virtualization of Software-Defined Networks: Analysis, Modeling, and Optimization***. PhD Thesis, Technische Universität München, März 2018.
- Blenk, Andreas; Kalmbach, Patrick; van der Smagt, Patrick; Kellerer, Wolfgang: ***Boost Online Virtual Network Embedding: Using Neural Networks for Admission Control.*** 12th International Conference on Network and Service Management (CNSM), 2016