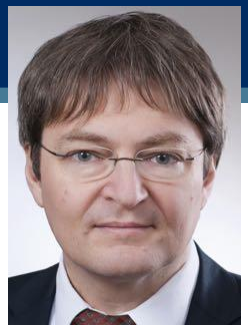
\Orchestrating a brighter world NEC

Artificial Intelligence in Network Operations and Management

Jürgen Quittek
NEC Laboratories Europe
quittek@neclab.eu

Short Bio



Dr. Jürgen Quittek is Managing Director of the NEC Laboratories Europe in Heidelberg, Germany. He received his degree in communications engineering from RWTH Aachen in 1989 and his Ph.D. from Hamburg University of Technology (TUHH) in 1996. After a postdoctoral year in Berkeley, California, he joined the NEC Laboratories in 1997. In 2000 he was a visiting professor at Freie Universität Berlin. He conducted research in the areas of neural networks, network management, data security, software-defined networking, energy-efficient communications, and 5G mobile networks, and he served as TCP chair and member of many conferences and workshops. As working group chair, rapporteur, and author he contributed to communication standards at ETSI, IETF, and ONF. His current research interests also include artificial intelligence and the internet of things.

Dealing with Complexity

We know
our networks
are complex

We could use some more help

Outline

Brief Overview and History of Machine Learning

Opportunities in Network Operations and Management

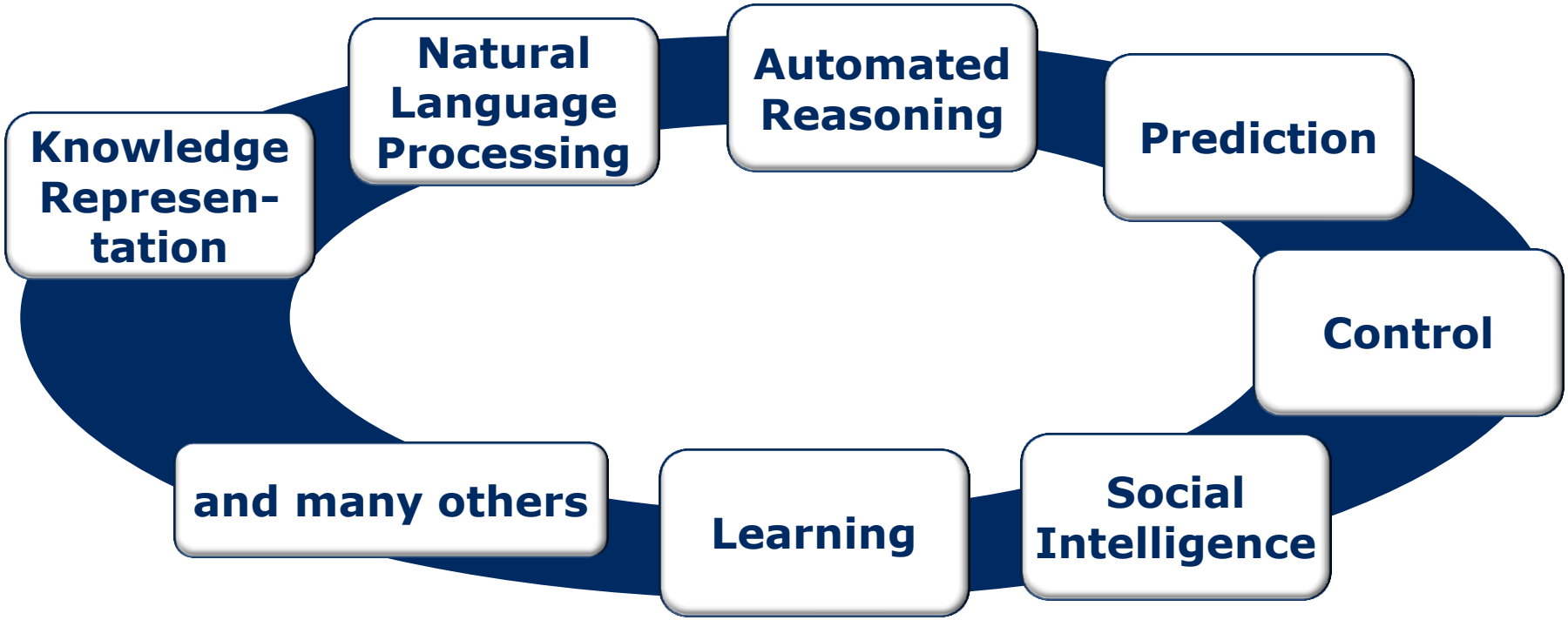
Examples

- Optimizing 5G network slicing
- User profiling with data analytics kit

Outlook

Artificial Intelligence (Machine Intelligence)

There is a huge variety of approaches to AI . . .



. . . with just a few big success stories

- (Deep) Machine Learning – biggest AI boom today
- Natural Language processing – Siri, Alexa, Cortina, etc.
- Profiling and Prediction – web ad placement

However, might be close to ubiquitous application of AI

The Power of AI today

- Siri, Alexa, Cortina, etc. show natural language processing
  - far from being perfect, but already high usability
  - Shortcoming rather in knowledge representation and social interaction than in NLP
- IBM DeepBlue, Google AlphaGo AlphaZero
  - Deep Blue: huge supercomputer run by a large team (1997)
  - AlphaGo:1200 CPUs, 180 GPUs, database of 30 million Go moves (2017/05)
    - Monte Carlo algorithms for tree search and (deep) learning from human teachers
  - AlpaGo Zero uses just 4 TPUs* and the basic Go rules (2017/10)
    - Beat AlphaGo 100:0 after three days of training **itself**
  - AlphaZero (2017/12)
    - Learned Chess, Go, **and** Shogi in a few hours and beat everything that existed before
- User preferences and behavior prediction in the web
  - The biggest AI market today: Placement of commercial advertisement
- Self-driving cars are emerging
- We will have robot doctors in some years from now
- Network operation and management will use AI as well

*Tensor Processing Units

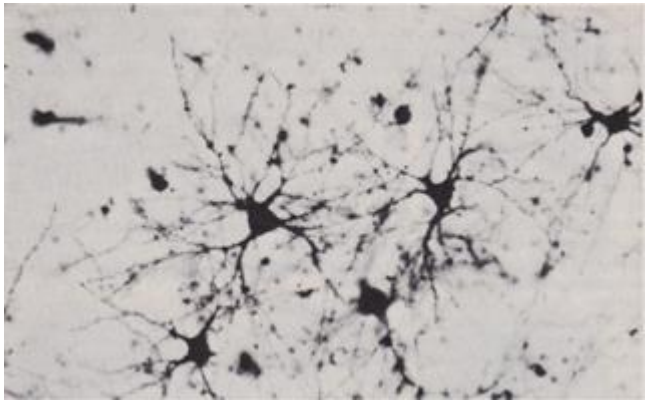
Brief Overview and History of Machine Learning

Problem: an unknown function

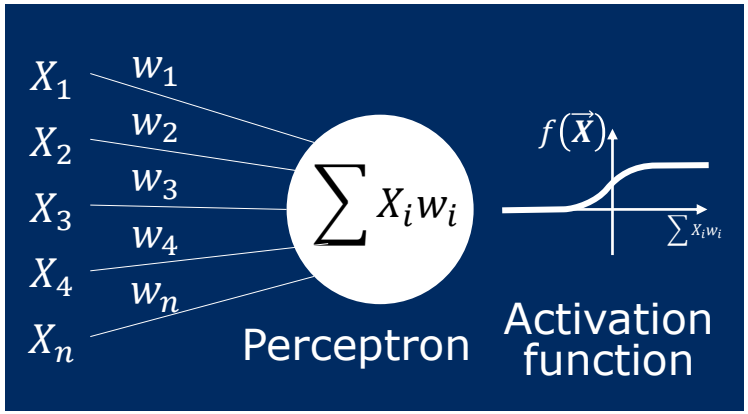
- Example image classification: is it a plane? is it a man? is it a 40G switch?
- Input is an image, output an indicator of a category or class of objects.
- How to get this function?
- Combinations of linear functions turned out to be too limited.
- Using polynomials turned out to be not feasible.

Idea: **neural networks** inspired by natural nervous systems

- Modeling neurons with non-linear perceptron (late 1950s to early 1970s)



Natural neurons



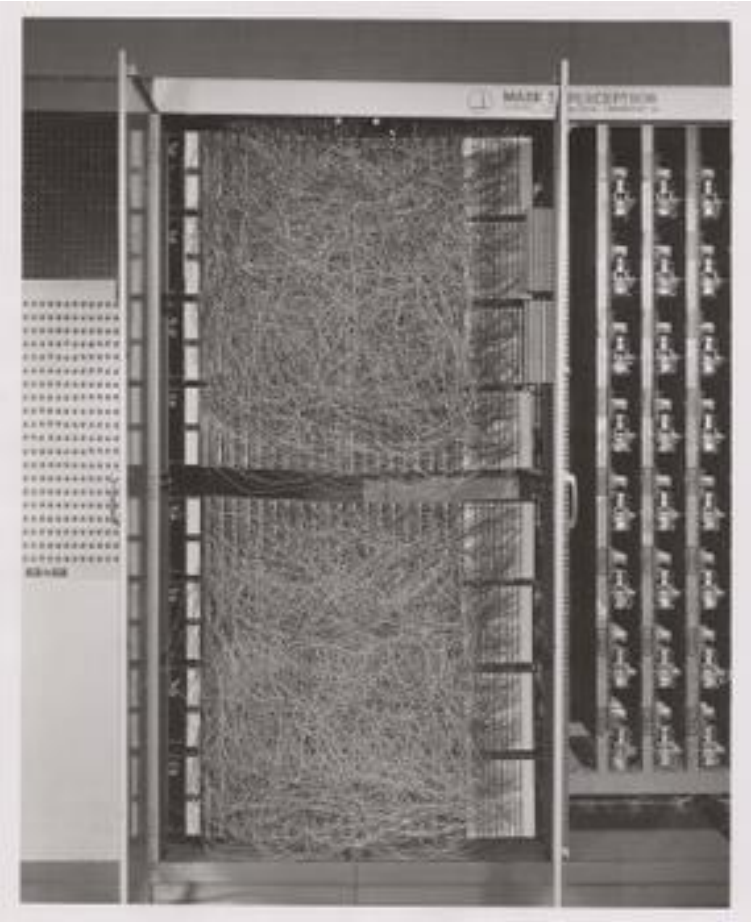
Artificial neurons

- Natural neurons receive and pass values in terms of firing rates
- The S-shaped activation function compresses the results and loses information, but the non-linearity is essential

Golden Age (1960s) and Long Winter of AI (1970s/80s)

- First boom od neural networks (golden age) ended around 1970 with disappointment
  - No significant achievements despite large investments in **image recognition**, **natural language processing**, **reasoning**, etc.
- For 20 years (long winter) there was very limited research funding and very little progress made

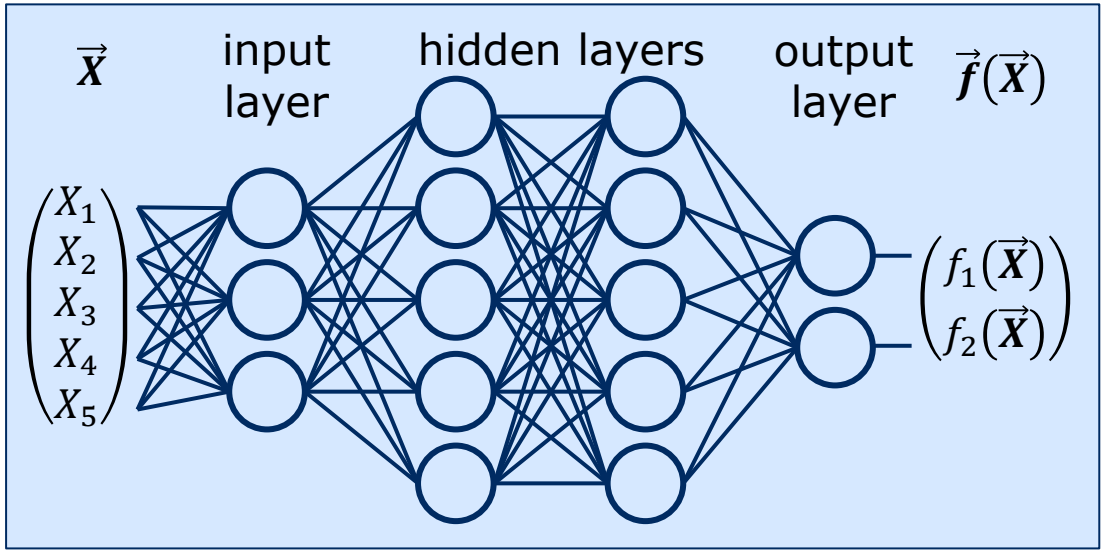
- Second boom in the late 1980s to early 1990s
  - PCs available
- New architectures
  - Hopfield net, multi-layer non-linear perceptron, self-organizing maps
- New algorithms
  - Backpropagation for training neural networks



Mark I Perceptron
Cornell, 1960

New Start with the Multi Layer Perceptron (~1990)

- Layered neural networks achieved first success stories in the 1990s
- Learning by backpropagation
  - Gradient based adjustment of perceptron weights to correct errors
- Competitive with other methods, but not really better



Multi Layer Perceptron (MLP)

- Size was too small to exploit advantages
  - Fully meshed connections between layers led to quadratic growth of computation
    - Attempts to 'thin out' connectivity were not successful
  - Number of layers and size of layers were limited by computing resources
  - Waiting for Moore's law to help
- Big breakthrough around 2010 by convolutional thinning and sufficient computing power for 'deep' neural networks
  - Deep networks starting to outperform other competing methods

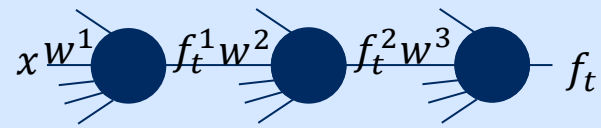
Learning: Unsupervised, Supervised, Reinforcement

Unsupervised: detecting hidden structure of "unlabeled" data

- A set of unlabeled examples inputs is presented to a learning system that detects structure in the data, e.g., by clustering

Supervised: Learning a function $f(\vec{X})$ that maps given inputs $\vec{X}$ to desired outputs y

- Training with 'labeled' data: each example input $\vec{X}(t)$ comes with a label $y(t)$ indicating the desired ('correct') output.
- The error E between $f_t\left(\vec{X}(t)\right)$ and $y(t)$) is used to adapt f and compute $f_{t+1}(\vec{X})$.
- After training use f for unlabeled data.

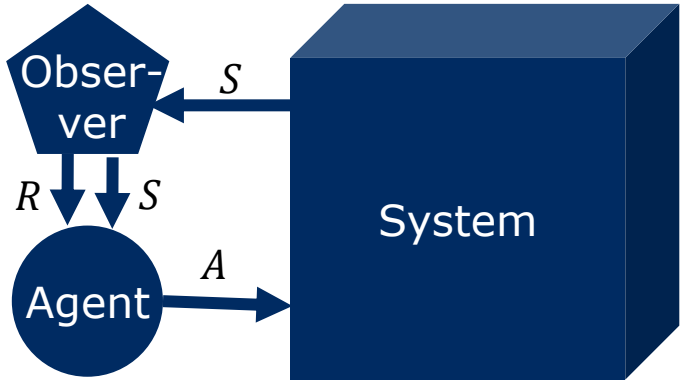


1. forward propagation: $x \rightarrow f_t(x)$.
2. backpropagation: $f_t(x), y(t) \rightarrow f_{t+1}(x)$ by a gradient descent $\partial E/\partial w$ for each weight w.

Example: Backpropagation

Reinforcement Learning

- Agent performs action A to change state S of a system.
- No indication of correct action is given, just a feedback value R called reward *after one or more time steps.*
- Learning system tries to optimize for expected cumulative future reward.
- After training agent can act without reward.

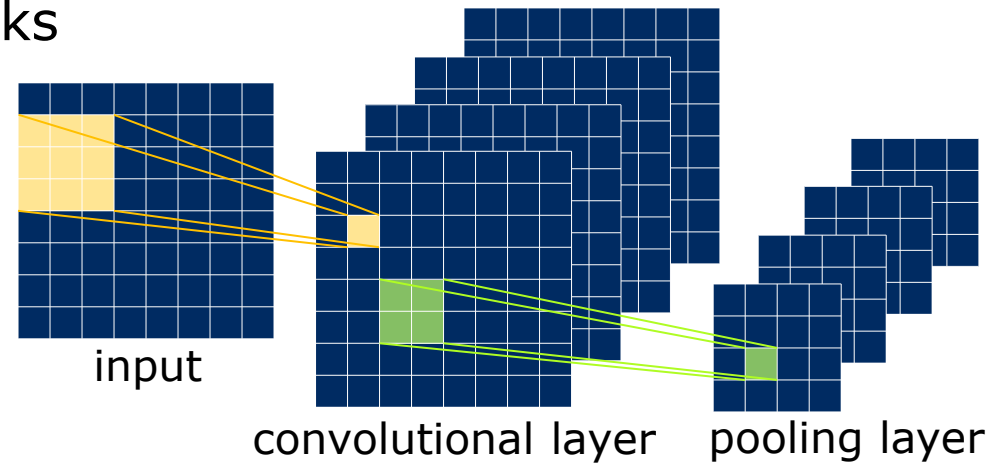


Reinforcement Learning Components

Convolutional Neural Networks (~2010)

Speeding up neural networks

- Convolutional layers (CL)
  - Every neuron has just a very limited number of inputs to the vicinity of a corresponding neuron in the previous layer
  - All neurons in a layer use the same set of weights
- Pooling layer (PL)
  - Neighboring neurons are merged (max, sum, etc.)



With these building blocks large networks can be composed

- Example:

- A fully connected layer (MLP) at the end connects all split components of layers

Neural networks with multiple convolutional layers are also called **deep neural networks**

Backpropagation is commonly used for **deep learning**

Runs very fast on modern GPUs

Today: Overcoming Limitations of $\vec{X}$

- Deep learning with convolutional networks needs a **single Euclidian input space** (vector space). This is not always given.

- For example, data collected in a communication network typically do not have an obvious embedding into a vector space.

- What do we have?
  - Text from log files
  - Graphs (connectivity, relationships)
  - Time series (can be vectorized with sliding window)

- The solution is **representation learning**
  - Embedding input data into Euclidian space with unsupervised learning

What Comes Next?

Knowledge Learning

- Improve AI knowledge with **multi-modal** data

Opening the AI Black box

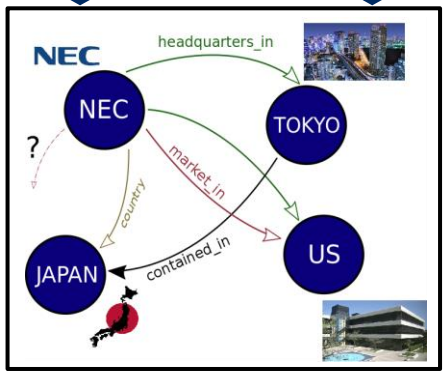
- **Find** reasons for AI answers

Technical Approach:

- Representation learning of multi-modal and multi-relational data
- Learn and perform logic reasoning on top of **knowledge graphs**

Data (image, text, …)

Domain knowledge



Knowledge Graph

AI system combining logic and deep learning

Queries

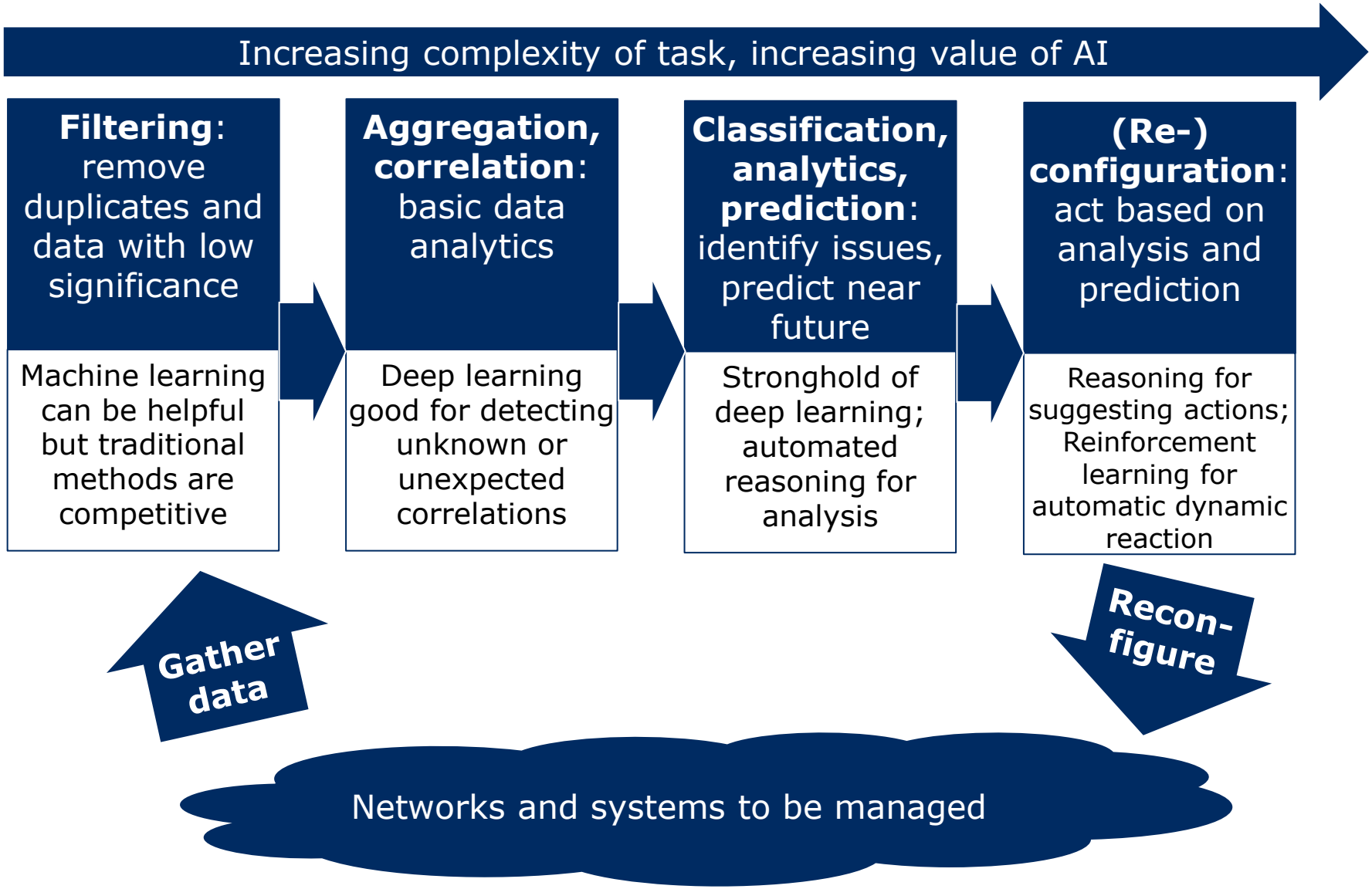
Is there a compliance risk?
Is there a security breach?

Answers

No. confidence: 0.89
Yes. confidence: 0.76

Explanations and Visualizations
Why did the system give this answer?

Opportunities in Network Operations and Management



Two Examples from NEC

5G network slice broker

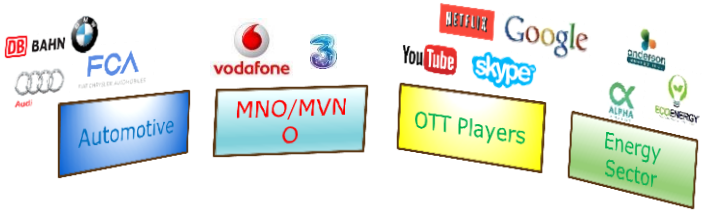
- Mapping per slice service requirements onto available resources
- Reinforcement learning just adds a small component to the overall solution
- Implemented on top of commercial components
- Published at Infocom 2017
  - V. Sciancalepore, K. Samdanis, X. Costa-Pérez, D. Bega, M. Gramaglia, A. Banchs: *Mobile traffic forecasting for maximizing 5G network slicing resource utilization*

Net2Vec telecom carrier analytics system

- AI engine for various analytics and prediction applications
- Uncovers the vast potential of AI in this area
- Components of NEC's network management solutions
- Example application: Development of user profiling application

5G Network Slice Broker

External infrastructure tenants are willing to pay for end-to-end self-contained virtual network (*network slices*)

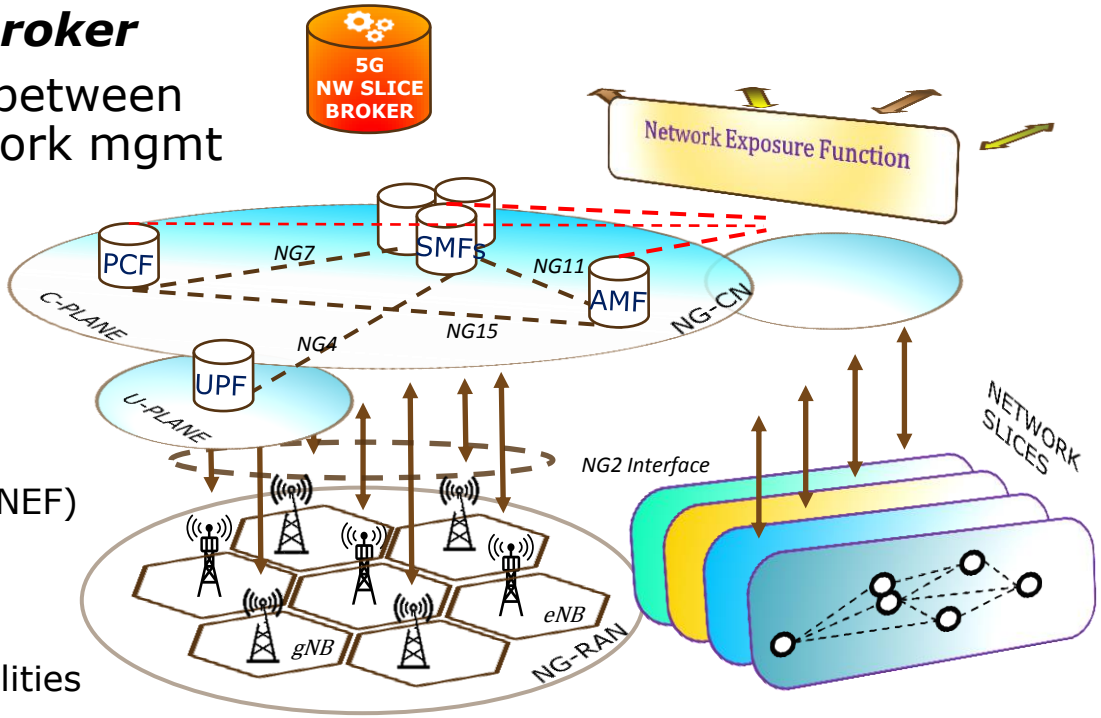


Problem:

How to map heterogeneous service requirements onto the network resource avaialbility?

Solution: ***5G Network Slice Broker***

A mediator should be interposed between external tenants and mobile network mgmt



3GPP Standard Requirements[2]:

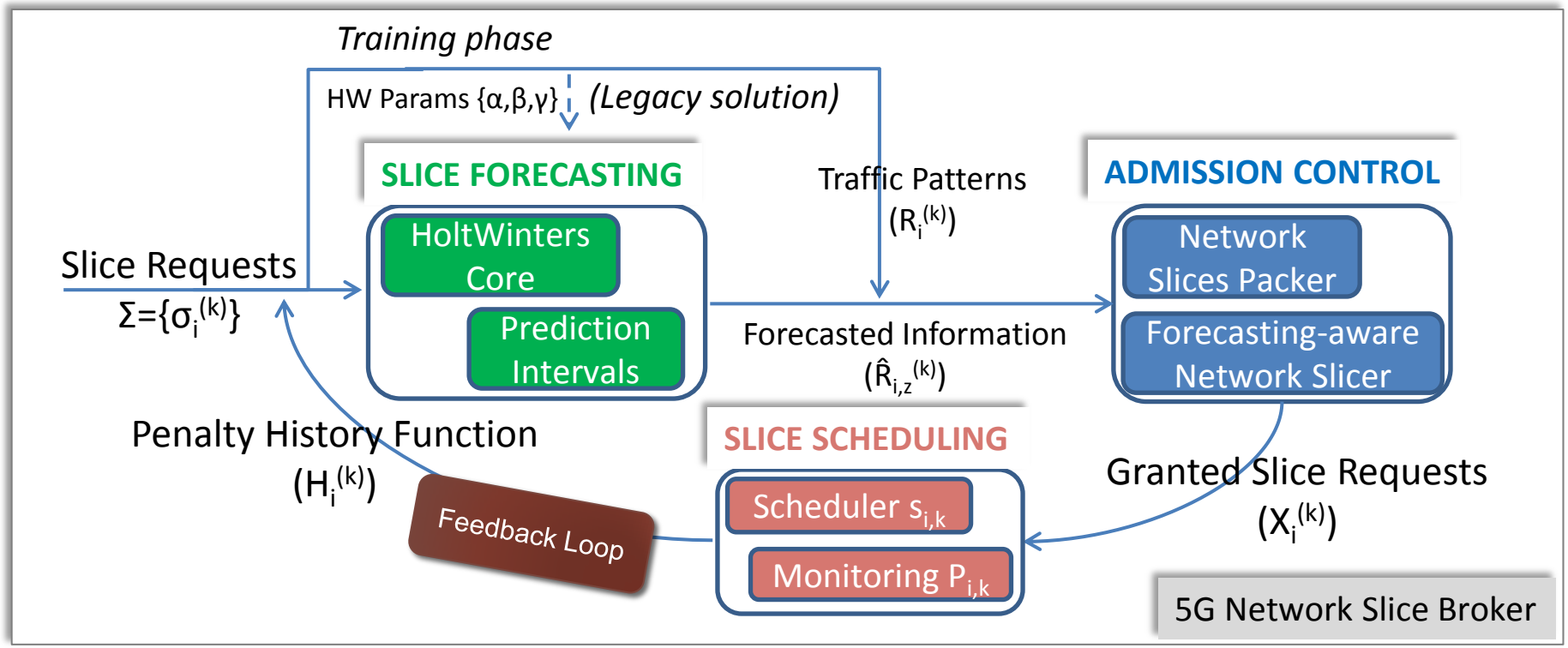
- Network Slice Templates (NSTs) are available for different services
- Each NST includes own SLAs
- Receive network slice requests from through a Network Exposure Function (NEF)
- Perform admission control based-on Slice Request NSTs
- Use NG2 interfaces to monitor KPIs and configure network slice on RAN facilities

Third Generation Partnership Project (3GPP), "Study on architecture for next generation system," 3GPP TS 23.799 v14.0.0 December 2016

5G Network Slice Broker Solution
Overbooking Mobile Networks Resources

5G Network Slice Broker features:

- Resource monitoring: e.g., resource blocks, MCSs
- Machine Learning operations for traffic forecasting: online reinf. learning
- Admission Control for network slice requests (based on forecasting info)
- Support for multiple classes of Network Slices SLAs
  - Heterogeneous QoS traffic requirements (data rate, latency, …)



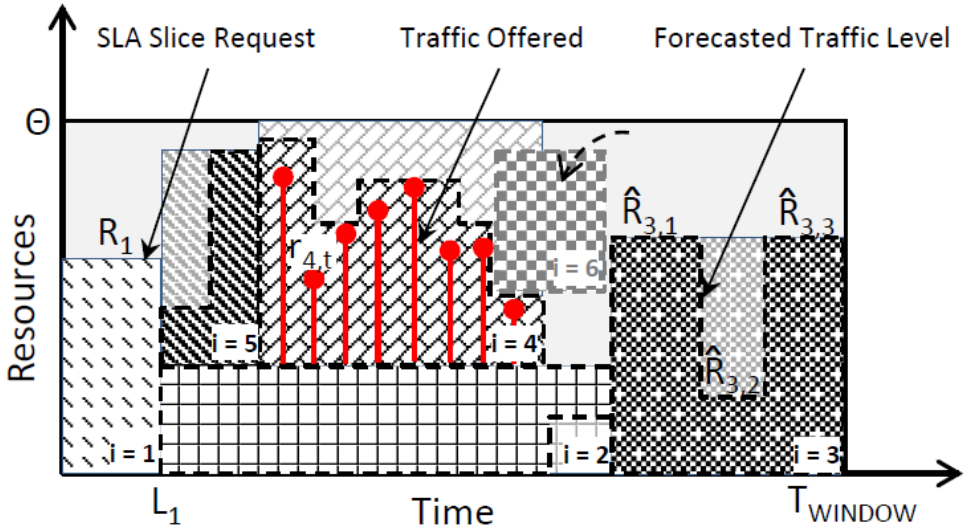
Math? Yes, thanks!

Admission Control:

- *Geometric Bin-Packing* problem:
  - Maximizing the overal system resource utilization
  - Optimization problem MILP → high complexity

Problem ADM-CONTROL:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in \mathcal{I}} c_i \cdot x_i \\ \text{subject to} \quad & \sum_{i \in \mathcal{I}} w_i \cdot x_i \leq W; \qquad \text{(relaxed)} \\ & \mathcal{S}(x_i) \cap \mathcal{S}(x_k) = \emptyset, \quad \forall i \neq k; \\ & \mathcal{S}(x_i) \subset \mathbb{S}, \qquad \forall i \in \mathcal{I}; \\ & x_i \in [0,1], \qquad \forall i \in \mathcal{I}; \end{aligned}$$



Slice Traffic Scheduling

- Minimizing the traffic scheduled per slice (while meeting the QoS constraints) in order to leave more room for other network slices.

Problem SLICER-SCHEDULING:

$$\begin{aligned} \text{minimize} \quad & s_{i,j}^{(k)} \\ \text{subject to} \quad & \left(\sum_{j=zk+\bar{t}}^{zk+\bar{t}+T^{(k)}} s_{i,j}^{(k)} \right) \geq r_{i,z}^{(k)} x_i^{(k)}, \forall z \in \left[0, \left\lceil \frac{L_i}{T^{(k)}} \right\rceil - 1\right]; \\ & \sum_{i \in \mathcal{N}} s_{i,j}^{(k)} \leq \Theta + P_{i,j}^{(k)}, \quad \forall j \in \mathcal{L}; \\ & s_{i,j}^{(k)} \in \mathbb{R}_+, \qquad \forall i \in \mathcal{N}, j \in \mathcal{L}, k \in \mathcal{K}; \end{aligned}$$

Mixed traffic classes with different QoS requirements:

- Mission Critical (guaranteed bit rate);
- Interactive Gaming;
- TCP-Based (FTP, e-mails, p2p).

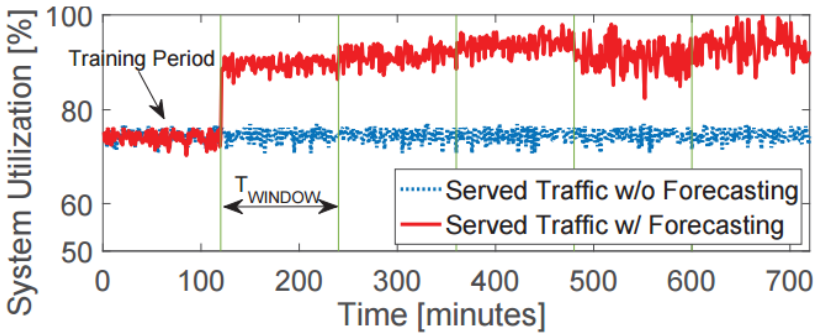
Monitoring and Feedback

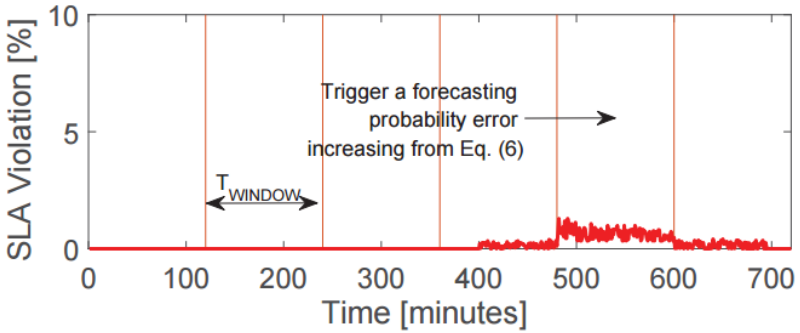
- Holt-Winters technique to predict traffic pattern for pair {tenant, traffic class}.
- Online Reinforcement Learning to dynamically adjust the forecasting interval accuracy:

$$\chi_i^{(k)} \; : \; h_i^{(k)} \Omega_\chi \sqrt{Var(e_{i,z}^{(k)})} = \hat{d}_i^{(k)}$$

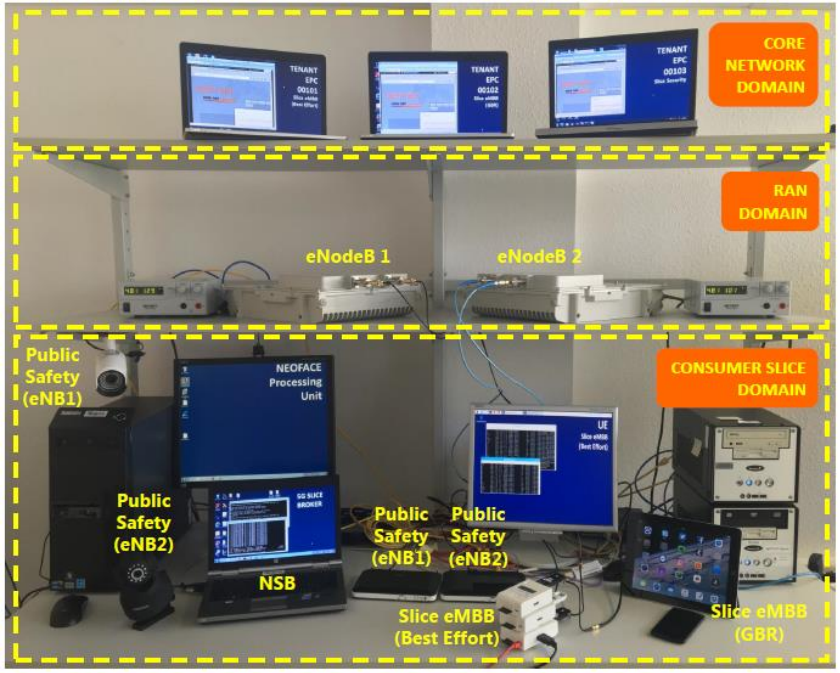
5G Network Slice Broker Evaluation

- Utilization of 5G networks can be improved by exploiting statistical multiplexing (overbooking)





- Machine learning can be used to cover components for which no analytical solution is available.
- Continuous learning can make the system adaptive to environmental changes.
- Prototype implemented of top of commecial components



Lessons Learned from Network Slice Broker

- Machine Learning can increase value of network management and control systems
  - Filling gaps in available set of functions
  - Improving quality and usability of available functions
- Existing functions are still strong
  - It is too early to replace all established optimization and control functions by machine learning
- However, established functions needed many years of development
  - Machine learning function develop much quicker
- Still, every solution needs good engineering.

Net2vec: AI Platform for Global Carrier Analytics

Equipment logs
HTTP(S) requests
User data

Network operators

BLOCKMON

Sense: high performance modular platform for data acquisition and transformation

Learn: advanced profiling of users, their interests, and equipment activity

Embedding
Deep learning

NEC the WISE

Net2vec

Actuate: deliver value with the learned knowledge

Manage NW and IT resources

Protect customers from threats

Game changing tarif plans

Monetize network data via ads

Net2Vec Network Analytics Motivation

What can we do with the operators data?

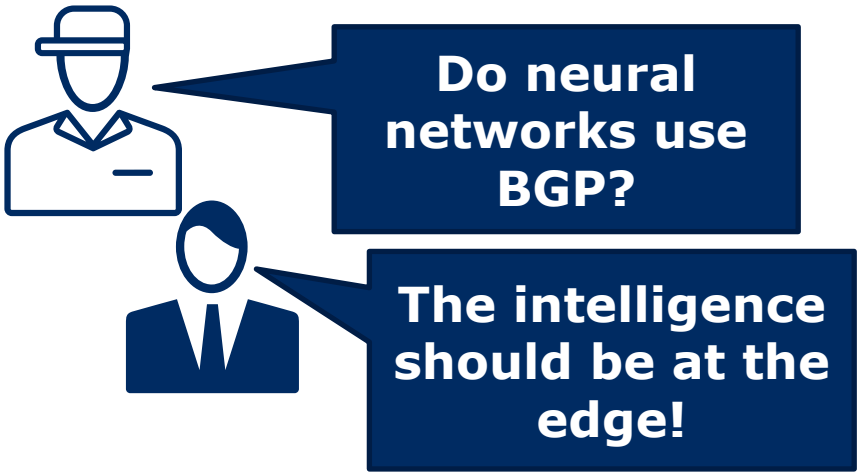
- **Improve the network**
  - Network planning.
  - Anomaly detection.
  - Root cause analysis of problems.
- **Improve the user experience**
  - Tariff recommendation.
  - Churn prediction.
  - Protection against malware/phishing.
- **Increase the revenue**
  - User mobility analysis for city planning.
  - Marketing reports.
  - Participate in the online advertising ecosystem.

How?

- Using Artificial Intelligence

Challenges

- Operators data is stored in multiple data silos.
- Network data is too fast.
- Network people don’t know machine learning, machine learning people don’t know about network.

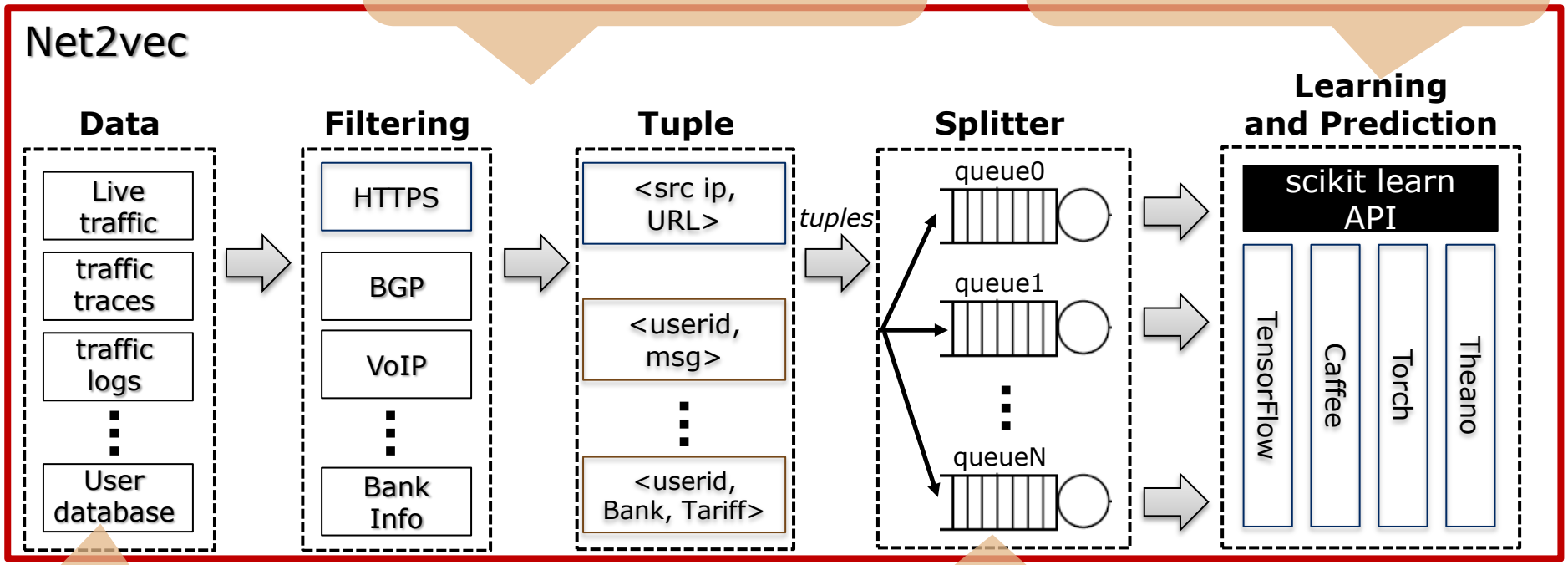


- Processes to obtain data are really slow.

Net2Vec Design

Local pre-filtering can be applied to minimize the data transferred

Different algorithms are applied for different use cases



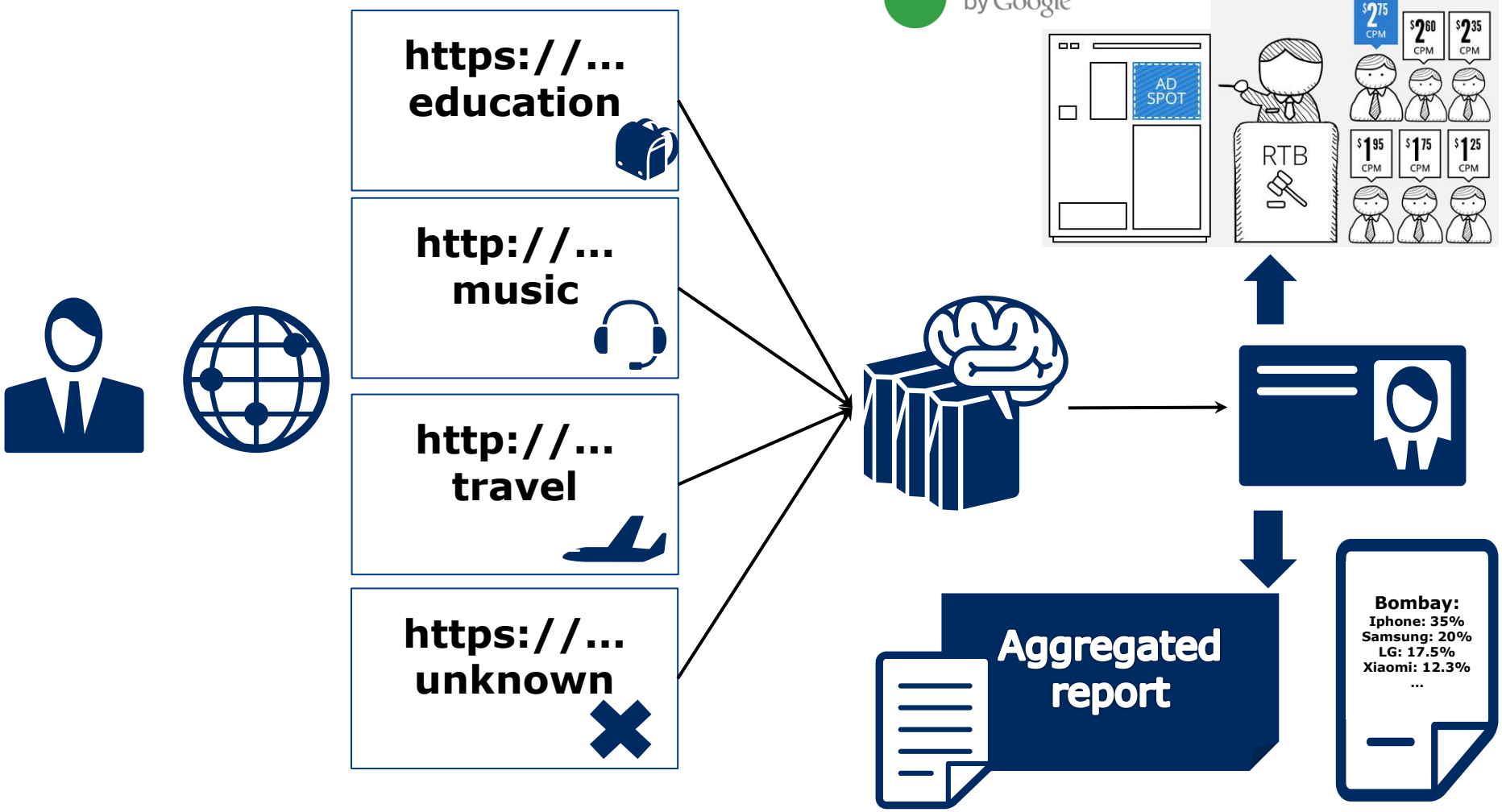
Net2Vec accept multiple sources of data

The system is parallelizable ensuring a great scalability

NEC the WISE

User Profiling Idea

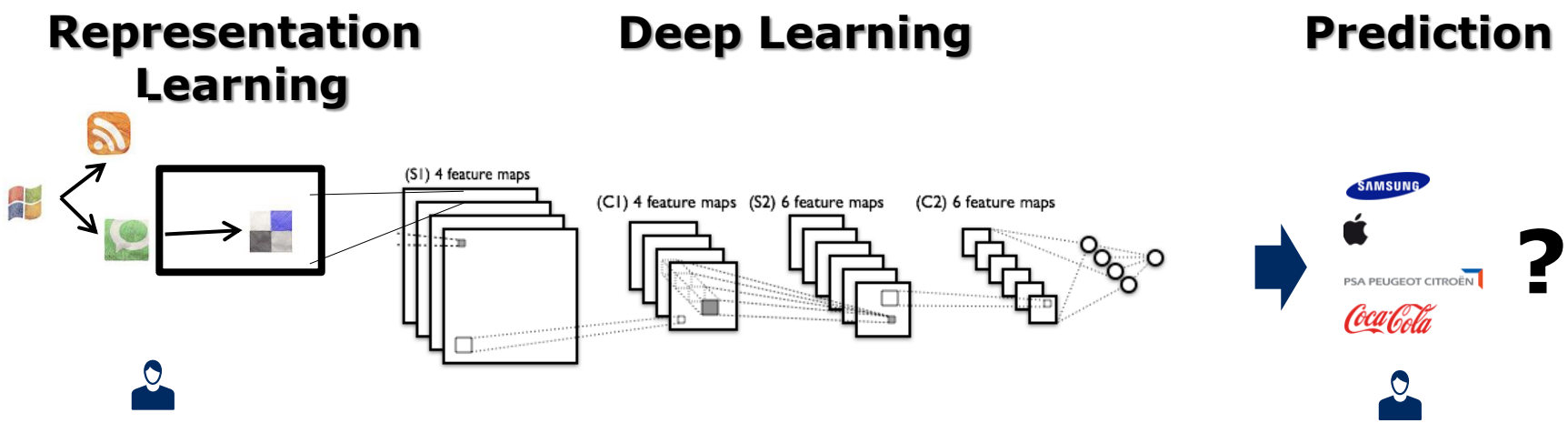
- Given the websites visited by the user we can generate a profile about the user interests.



High Level Design of Network User Profiling

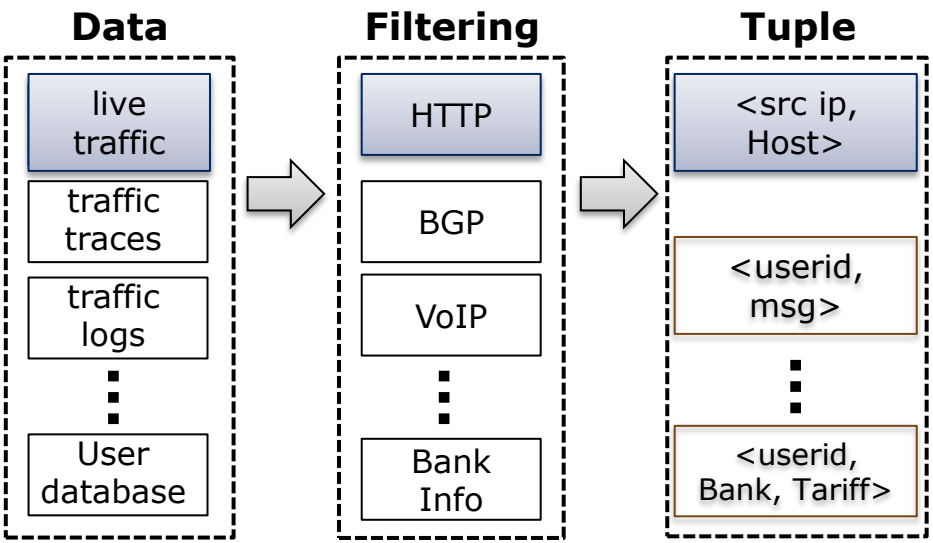
Input: User's HTTP(S) requests, domain labels, interest categories

Output: Predicted interest categories per user



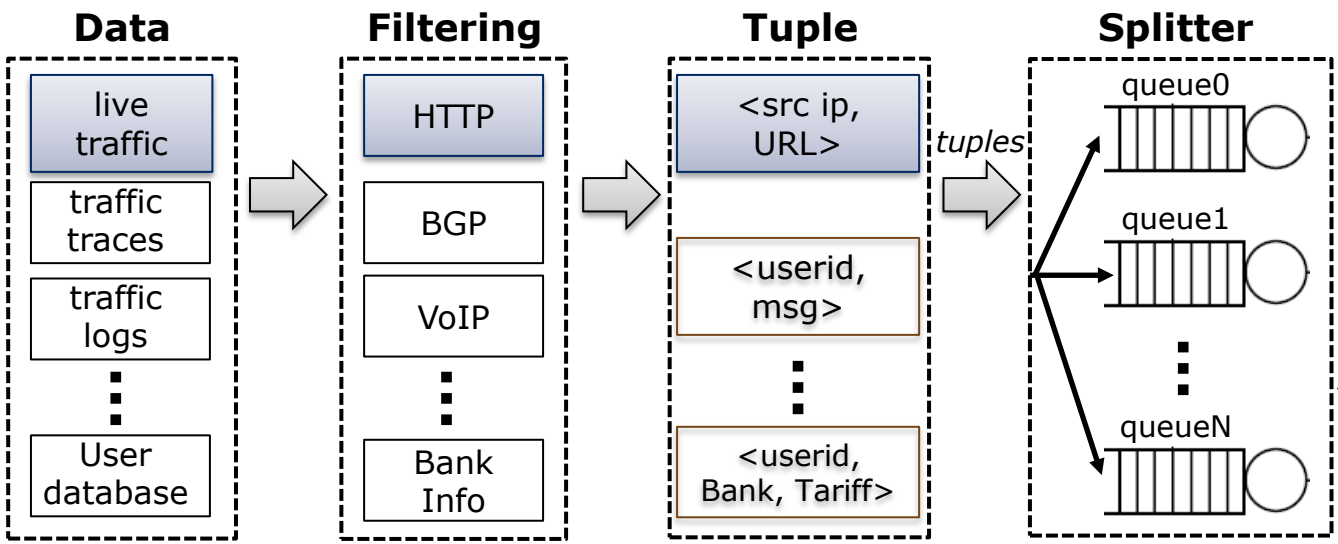
Capturing Data on the Wire

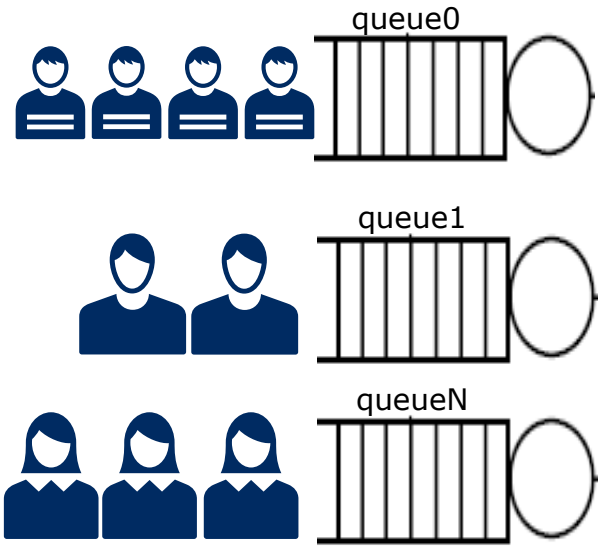
- We only need and identifier of the user and the host visited.



- Fast data capturing based on Netmap
- Listening to six 10Gb/s network ports per probe.
- For HTTP(S) parsing a single CPU core is sufficient for any packet size

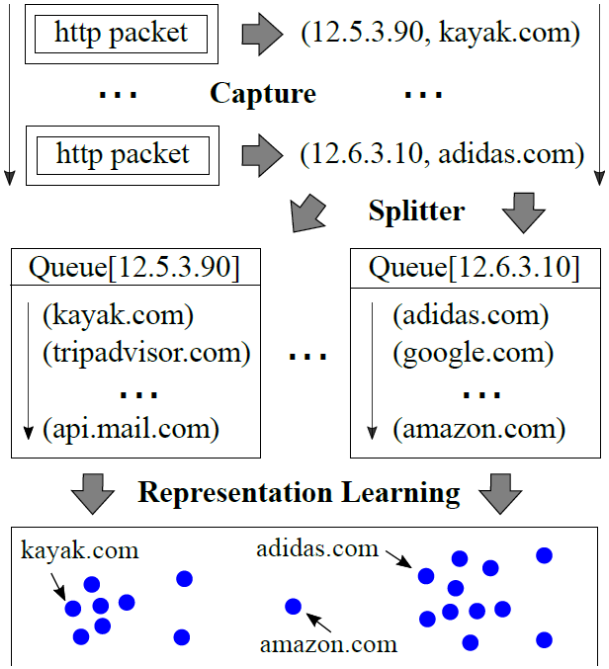
Splitter for Parallelization and Scalability of Profiling

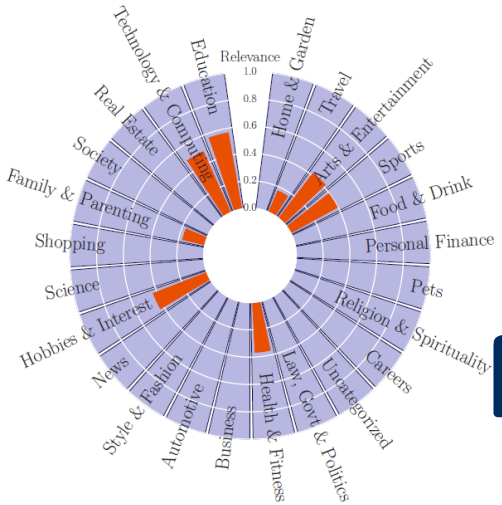




- The splitter distribute users in queues and periodically sends batches of data to deep learning engines.

First Try: Too Many Unknown (Unlabeled) Domains

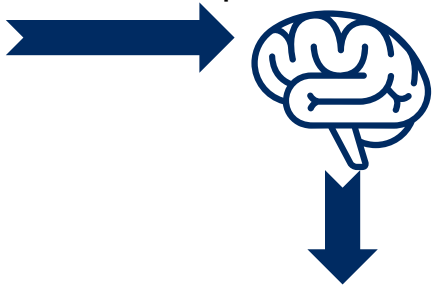




- Most of the domains are not labeled:
  - Login required
  - API calls of mobile apps
  - Unknown websites
- Solve it by machine learning:

kayak.com – Travel
api.booking.com –
api.tripadvisor.com –
travel.cnn.com – Travel
api.espn.com –
12cdn.akamai.com –
api.nba.com –
…

Model trained the previous day



kayak.com – Travel (p=1)
travel.cnn.com – Travel (p=1)
skyscanner.com – Travel (p=0.7)
espn.com – Sports (p=0.4)
hotels.com – Travel (p=0,35)
acm.com – Science (p=0,01)

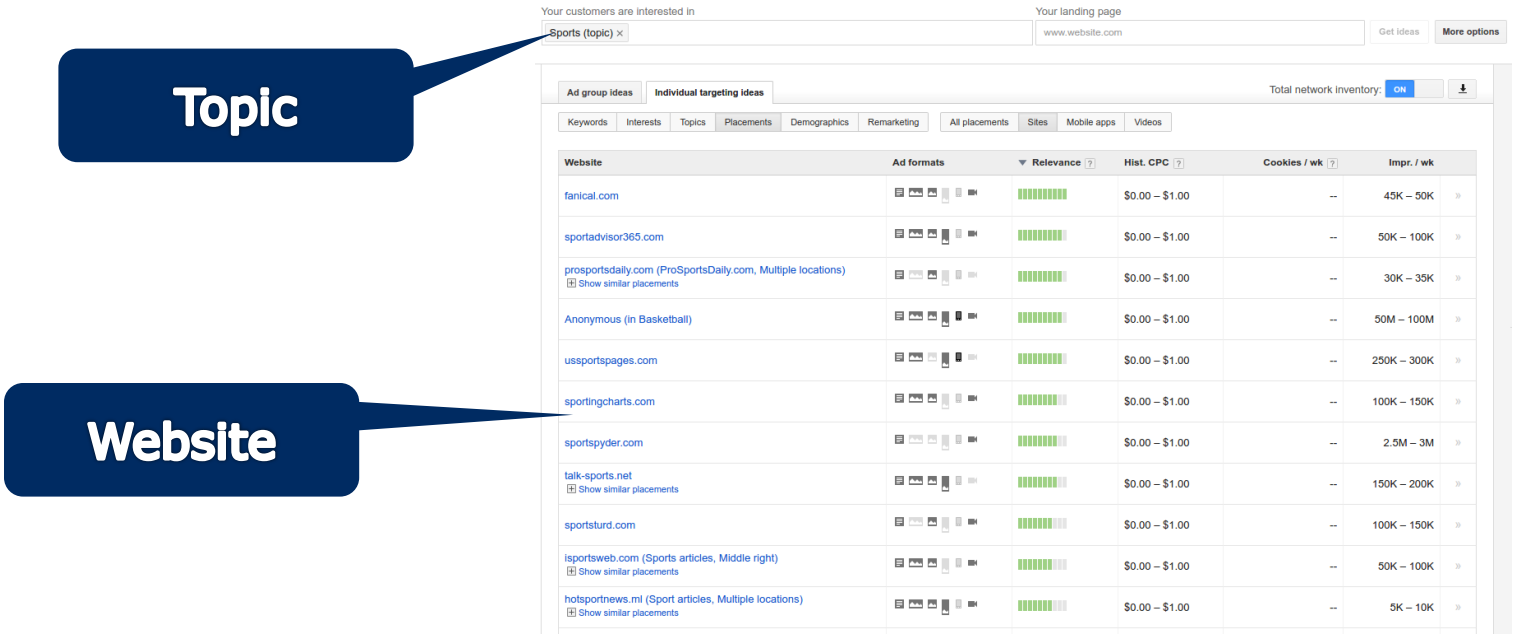
The Solution: Do It Yourself with AI

- Why is it difficult to get domains labeled with categories?
  - Manual labeling is too expensive because the number of webpages is huge
  - It is difficult to obtain good ground truth
  - Encrypted Internet traffic is growing
  - Mobile traffic (Mobile Apps) is growing

- The data analyst approach:
  Start your own host database of labels for hosts visited by users.
- Step 1: Label "readable" domains based on online analysis of their content.
- Step 2: Label domains from which it is not possible to download the content (private content, API calls, etc.) with knowledge from step 1 by learning representations for all domains in an input space.

Step 1: Obtaining Ground Truth

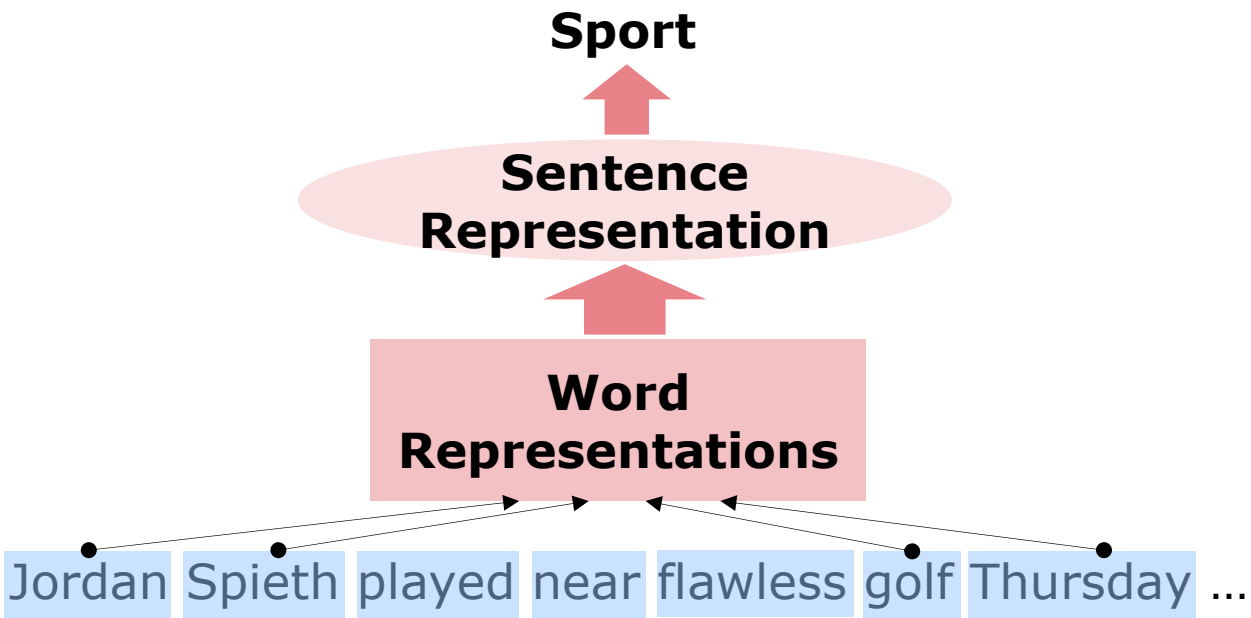
- We create our gold standard asking AdWords the most typical webs for each category defined by IAB.
  - We use Google (the main advertising actor) as a starting point -> ☺



- We retrieve more than 100 websites related with each one of the categories defined by the Internet Advertising Bureau (IAB), the standard in online advertising. -> ☺
- Different use cases require different categorizations:
  - Working now in a security use-case.
  - Direct product (i.e., running shoes instead of sports) related categories using common-crawl data.

Step 1: Categorizing Readable Webpages

- Similar to Facebook FastText, we used a deep learning algorithm able to categorize webpages based on their content.



- For each unknown webpage we download the content in the main page, as well as the content in all the links we find on it.
- Learning categories is language-specific. Our categorizer reads English and Spanish.

Step 2: Inferring Labels for Other Websites

- It is impossible to obtain the category for all the websites in the world (even when our database is growing.)
  - The users can visit websites that we have not seen before.
  - Mobile APPs use APIs that cannot be used externally.
  - Some websites require to login to can download the content.
  - …

- Inspiration by AI systems for word suggestions in text editors (e.g. WhatsApp):

  Treat sequences of host visits as sentences and give domains that often occur close to each other the same category

Step 2: Learning Domain Sequences

Tuple formation:

- Given a window size of 2 and the sentence:

The rain in Spain stays mainly in the plain

- Tuples are collected by picking words from the key word within the range of the window:

The rain in Spain stays mainly in the plain

The rain in Spain stays mainly in the plain

The rain in Spain stays mainly in the plain

The rain in Spain stays mainly in the plain

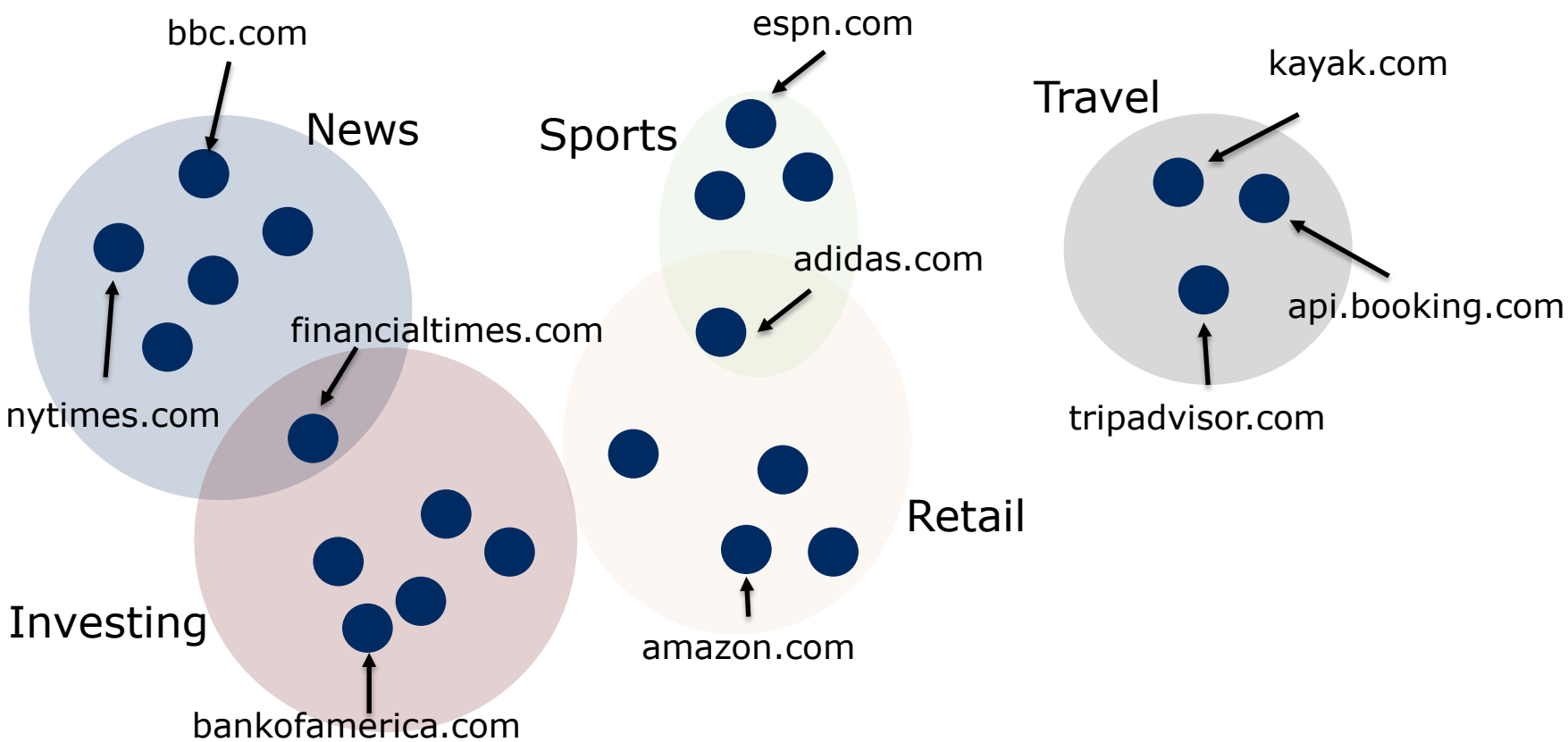
Training Tuples
(The, rain)
(The, in)
(rain, The)
(rain, in)
(rain, Spain)
(In, The)
(In, rain)
…

- We can do the same with host sequences:

bbc.com akamai.com api.weather.com espn.com

Step 2: Learning Data Representations

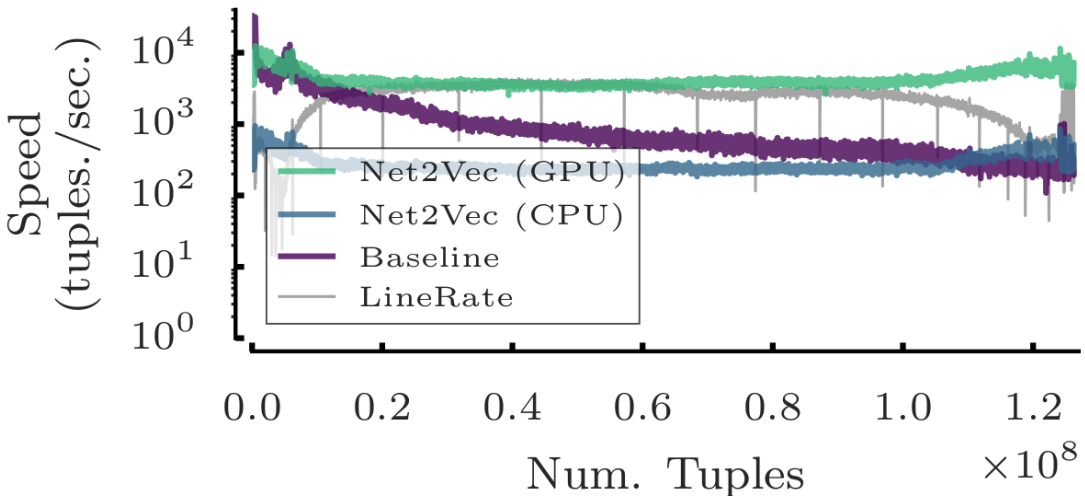
- Intuitively, similar domains get located in similar positions in the representation space.
- Imagine a simplistic 2D representation space:



- This representation allows us to categorize domain even if we don't know their categories: api.booking.com -> Travel

Evaluation of User Profiling with Net2Vec

- We used 2 real (anonymized) datasets to test our system.
  - Proxy logs of an Asian mobile network in Asia.
  - Network traces of a European fixed line operator.
- We trained our model using the network data of the day X, and executed Net2Vec on the data of the day X+1.
- We labeled the top 200K hosts most visited in day X.
  - Labeling only 50K would have been sufficient already
- For fast profiling kept the browsing history of users in memory and calculating the profile with the domains categorized.
- Result: You can profile 60Gbps HTTP traffic with a single GPU
  - Or multiple CPU.
  - The system is highly parallelizable because of the user-based splitter



Lessons Learned from Net2Vec

- With a modular toolkit like Net2Vec including library of AI functions you can quickly build powerful network analytics tools.

- The potential application area is huge:
  - Improve network
  - Improve user experience
  - Generate additional revenue

- Hardware cost is negligible!

- Still, every application needs individual good engineering.

Conclusion

- Recently, artificial intelligence, particularly machine learning, has become powerful enough to have impact
  - sufficient computing power
  - deep learning
  - representation learning
- This evolution is just starting, expected impact is huge
- AI will also enter network operations and management
- **The time to study it is now!**
- There are still many low hanging fruits.
- Analytics frameworks can help a lot.
- Still, every application needs individual good engineering.

Thank you for your attention!

Questions are highly welcome

Orchestrating a brighter world

NEC