UNIVERSITY OF SUSSEX

Edge Learning for Dynamic Networks

Naercio Magaia

SDI Seminar @ CMU

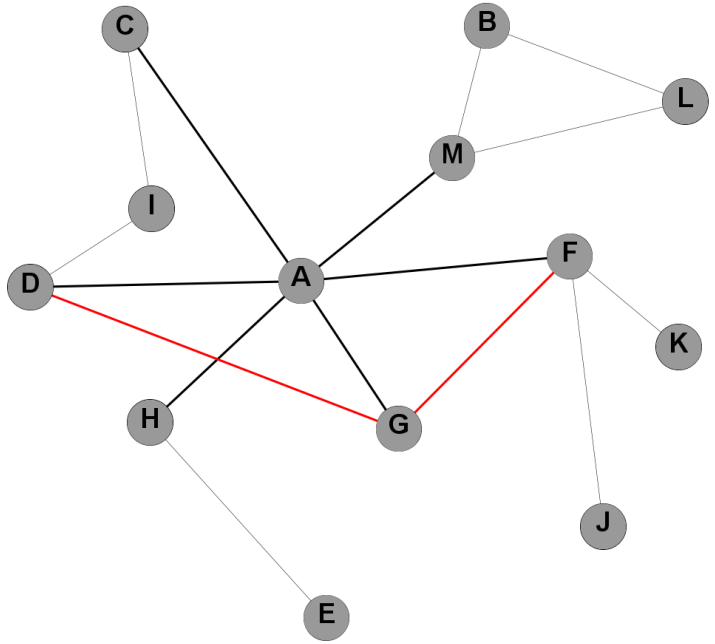
1 September 2022

Outline

1. Introduction
2. System Model
3. Building the Neighboring Graph
4. Edge Learning based Clustering
5. Misbehavior Model
6. The Smart Network Monitoring System
7. Performance Evaluation
8. Conclusions

Introduction: Dynamic network

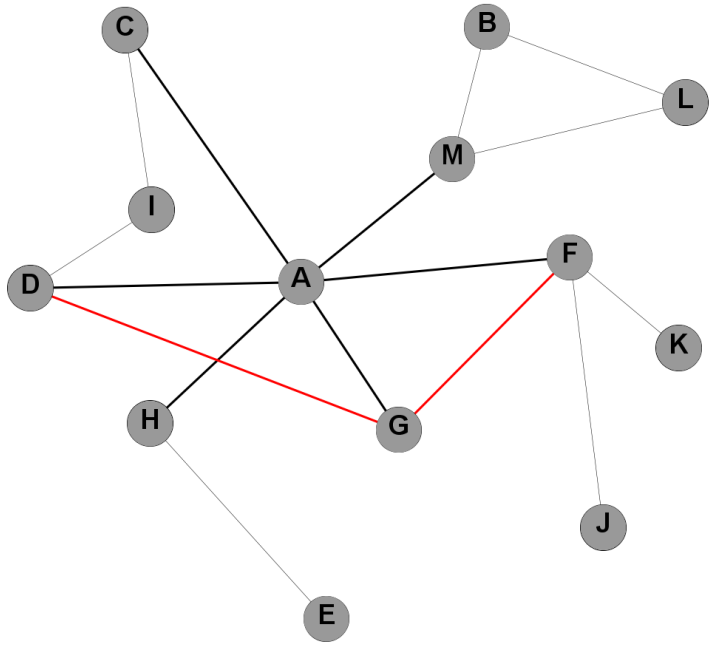
- Let
  - V denote a set of entities (or vertices, nodes)
  - E denote a set of relations (or edges, links) among these entities, and
  - an alphabet L incorporating possible properties that such relation might have (e.g., *terrestrial link, bandwidth of 8 MHz*);
- Specifically, $E \subseteq V \times V \times L$.
- It is assumed that entities' relations happen along a time span $\mathcal{T} \subseteq \mathbb{T}$ called the system's *lifetime*.



✓ N. Magaia et al., "Betweenness centrality in Delay Tolerant Networks: A survey", Ad Hot Networks, 2015. https://doi.org/10.1016/j.adhoc.2015.05.002

Introduction: Dynamic network

- The temporal domain $\mathbb{T}$ is commonly assumed to be:
  - (1) $\mathbb{N}$ for discrete-time systems,
  - (2) $\mathbb{R}^+$ for continuous-time systems.
- Thus, the dynamics of the system can be described by a temporal graph (or time-varying graph) $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$, where
  - $\rho: E \times T \to [0,1]$, named *presence* function.
  - $\zeta: E \times \mathcal{T} \to \mathbb{T}$, named latency function.



Introduction: Internet of Vehicles

What Is It?

- It is as a network of the future in which integration between devices, vehicles, and users will be unlimited and universal, overcoming the heterogeneity of systems, services, applications, and devices

Main Components?

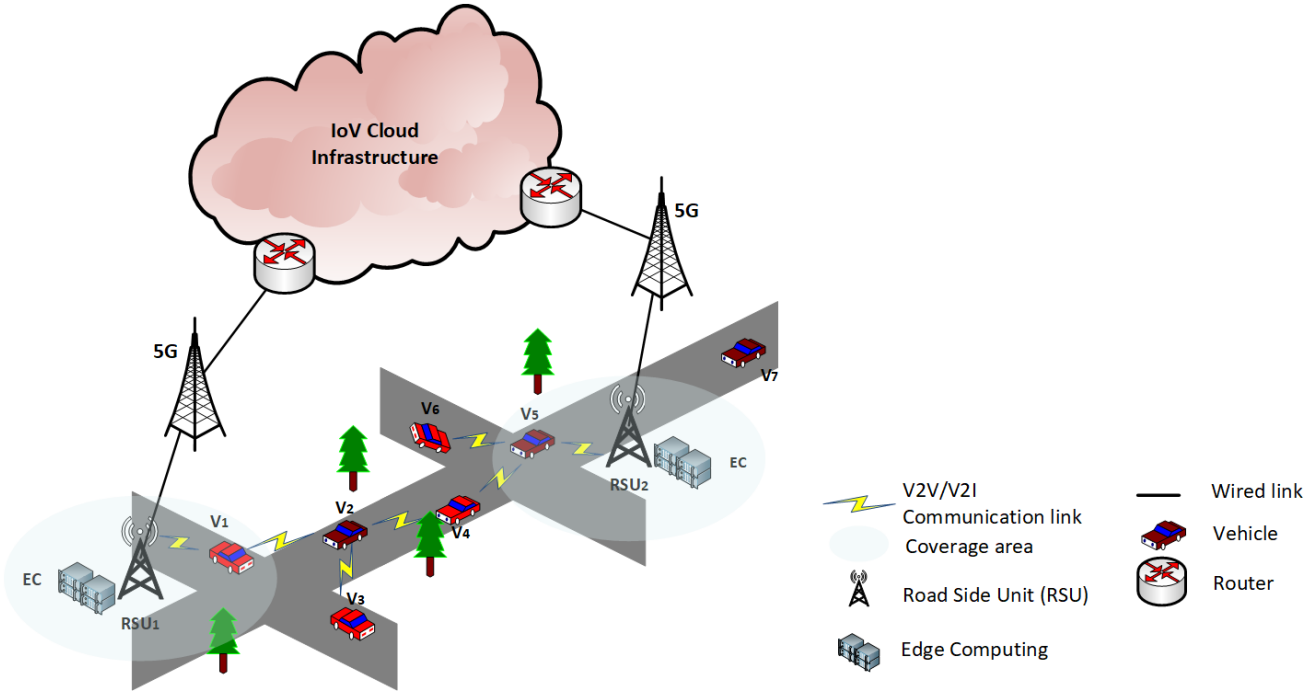
- Elements:
  - Users, IoV objects (i.e., Vehicles, pedestrians, RSUS), and trusted authorities (i.e., public networks.
- 3 Layers:
  - Vehicles, Edge, Cloud

Main Goals?

- Improve the user's experience and safety by coordinating human, vehicle and the environment.
- Safety, comfort and prompt delivery of its occupants with minimum impact on the environment.

Introduction: Edge Learning

- Edge computing, which are computations at the edge, in conjunction with Machine Learning have turned into a powerful tool for the local decision-making
- EL is based on the idea that storage and computational resources should be used at the edge of the network



Problem

For the IoV to work at its full potential, a huge amount of data has to be able to spread throughout the network

The dynamics of vehicular environments poses many challenges to realize efficient data dissemination.

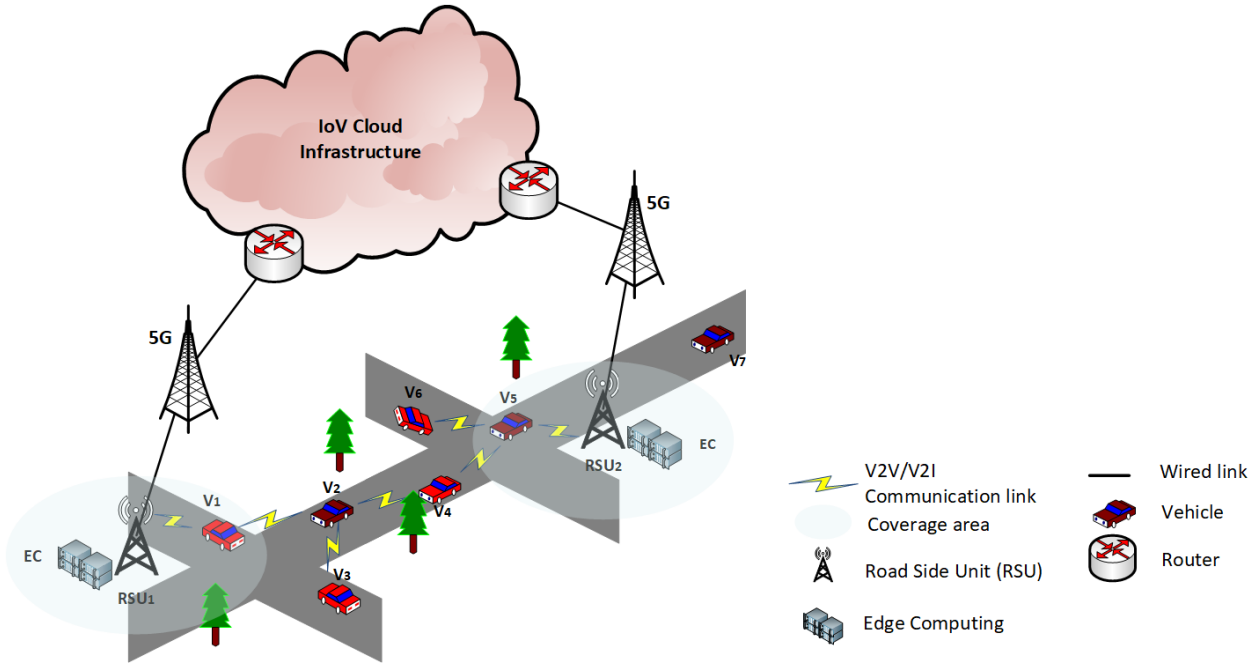
The dissemination of this data leads to numerous cybersecurity threats and vulnerabilities

Objectives

- In order to address the latter issue and **guarantee stable and reliable** communication between the nodes, we propose
  - A novel clustering algorithm leveraging EL and social relations among nodes, which are stable in dynamic networks
  - A Deep Learning (DL) based monitoring system at the Edge Layer to detect any anomaly and classify node behaviors in the network, helping mitigate the impact of misbehaving nodes.

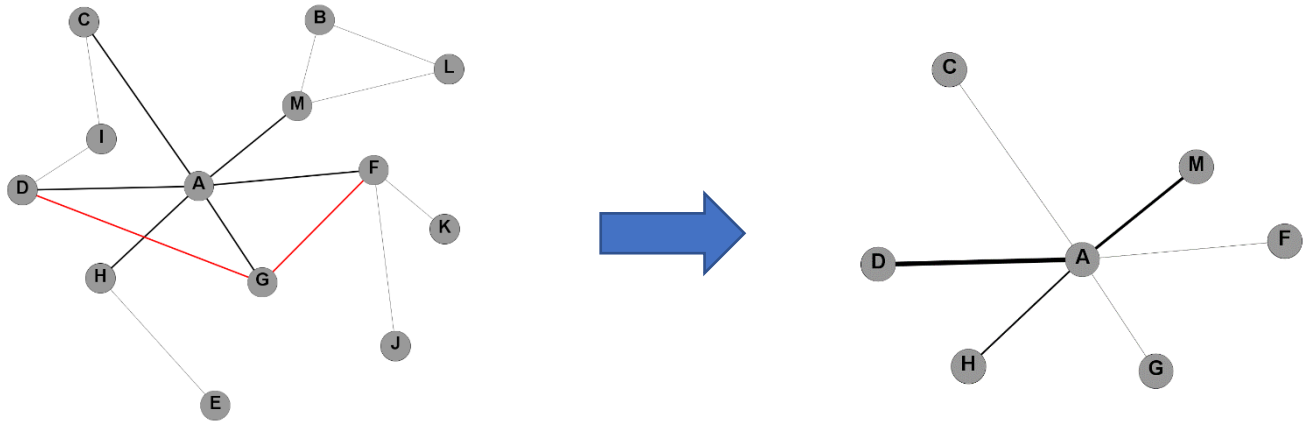
System Model

- Two groups of nodes
  - Mobile and stationary
- Vehicle layer
  - Cars and Buses
- Edge layer
  - RSUs
- Mobile nodes can offload data to the nearest RSU to be processed.



Building the Neighboring Graph (i)

- If $\zeta \to 0$, the neighboring graph can be modeled as a time-varying graph $\mathcal{G} = (V, E, \mathcal{T}, \rho)$
- Let $H = (V_H, E_H)$ be a subgraph of $G = (V, E)$, denoted $H \subset G$, iff $V_H \subset V$ and $E_H \subset E$.
  - H is a local subgraph (hereafter *neighboring graph*) with respect to a vertex $v \in V$, iff all vertices in the subgraph can be directly reached from v.



✓ N. Magaia et al., "Group'n Route: An Edge Learning-based Clustering and Efficient Routing Scheme Leveraging Social Strength for the Internet of Vehicles", IEEE T-ITS, 2022. https://doi.org/10.1109/TITS.2022.3171978

Building the Neighboring Graph (ii)

1. The time-varying average separation period between nodes i and j at timeslot τ_k (hereafter average separation period) is given by

$$\delta_{(i,j),\tau_k}(x) = \frac{\int_{\tau_k} x_{i,j}(t)dt}{n_{i,j}}$$

2. The normalized average separation period $\hat{\delta}_{\tau_k}$ at timeslot τ_k is given by

$$\hat{\delta}_{\tau_k} := \hat{\delta}_{(i,j),\tau_k}(x) = 1 - \frac{\delta_{(i,j),\tau_k}(x)}{|\tau_k|}$$

3. The normalized average separation period in the same timeslot τ_k over consecutive days is updated using an EWMA as follows

$$\Delta_{\tau_k}^{t} = \begin{cases} \hat{\delta}_{\tau_k}^{1}, & t = 1 \\ (1-\alpha)\cdot\Delta_{\tau_k}^{t-1} + \alpha\cdot\hat{\delta}_{\tau_k}^{t}, & t > 1 \end{cases}$$

Building the Neighboring Graph (iii)

4. The unbiased variance estimator $\hat{\sigma}$ at timeslot τ_k of day t is given by

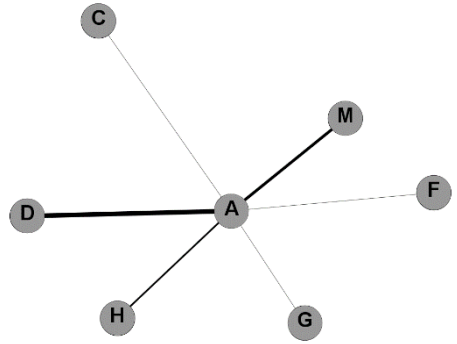
$$\hat{\sigma}_{\tau_k}^{\mathrm{t}} = \begin{cases} \left|\Delta_{\tau_k}^{1} - \hat{\delta}_{\tau_k}^{1}\right|, & \mathrm{t} = 1 \\ (1-\beta)\cdot \hat{\sigma}_{\tau_k}^{\mathrm{t}-1} + \beta \cdot \left|\Delta_{\tau_k}^{\mathrm{t}} - \hat{\delta}_{\tau_k}^{\mathrm{t}}\right|, & \mathrm{t} > 1 \end{cases}$$

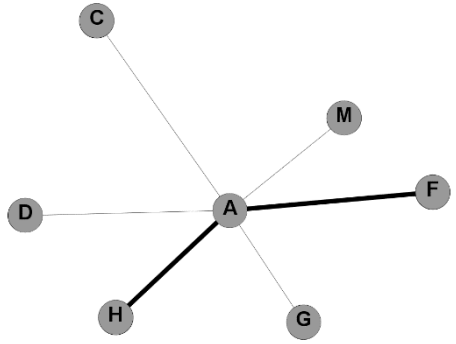
where $0<\beta<1$.

5. The time-varying weight $\rho_{i,j}$ over a daily time-period is given by

$$\rho_{i,j} = \frac{1}{|\tau^{\mathrm{t}}|}\sum_{k=1}^{|\tau^{\mathrm{t}}|} \Delta_{\tau_k}^{\mathrm{t}}$$

where $\left|\tau^{\mathrm{t}}\right|$ is the number of timeslots of day t





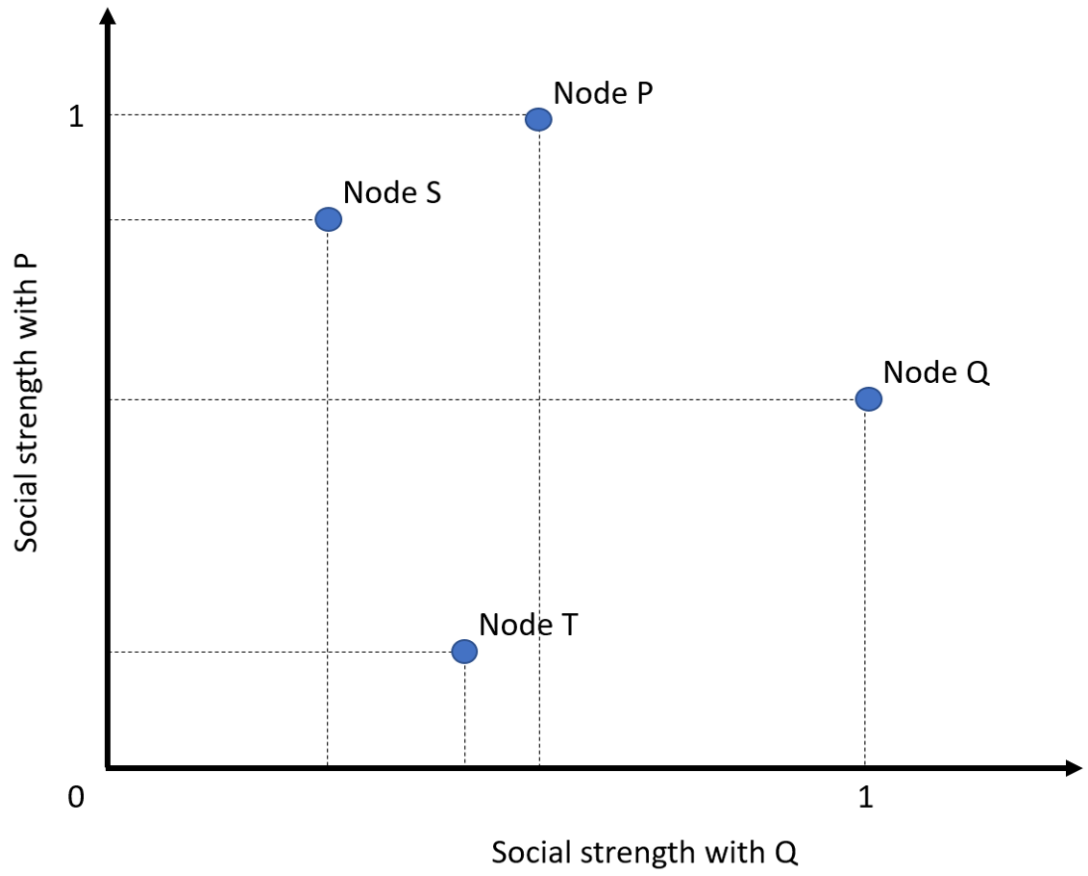
Edge Learning based Clustering (i)

- There are countless ways to cluster nodes in a network and clustering them based on their position is the most trivial method.
  - This leads to a **network stability problem**!
- Instead of node's position, a more stable metric should be considered, such as the **nodes' social relationships**.
  - Every time node i connects with node j, their social strength is increased.
  - This metric decreases if node i and node j do not connect during a given period.
- It is also assumed that every node updates its social relationship metric periodically (e.g., every hour)
- Therefore, nodes can be clustered according to the **similarity of their relationships**

✓ N. Magaia et al., "Group'n Route: An Edge Learning-based Clustering and Efficient Routing Scheme Leveraging Social Strength for the Internet of Vehicles", IEEE T-ITS, 2022. https://doi.org/10.1109/TITS.2022.3171978

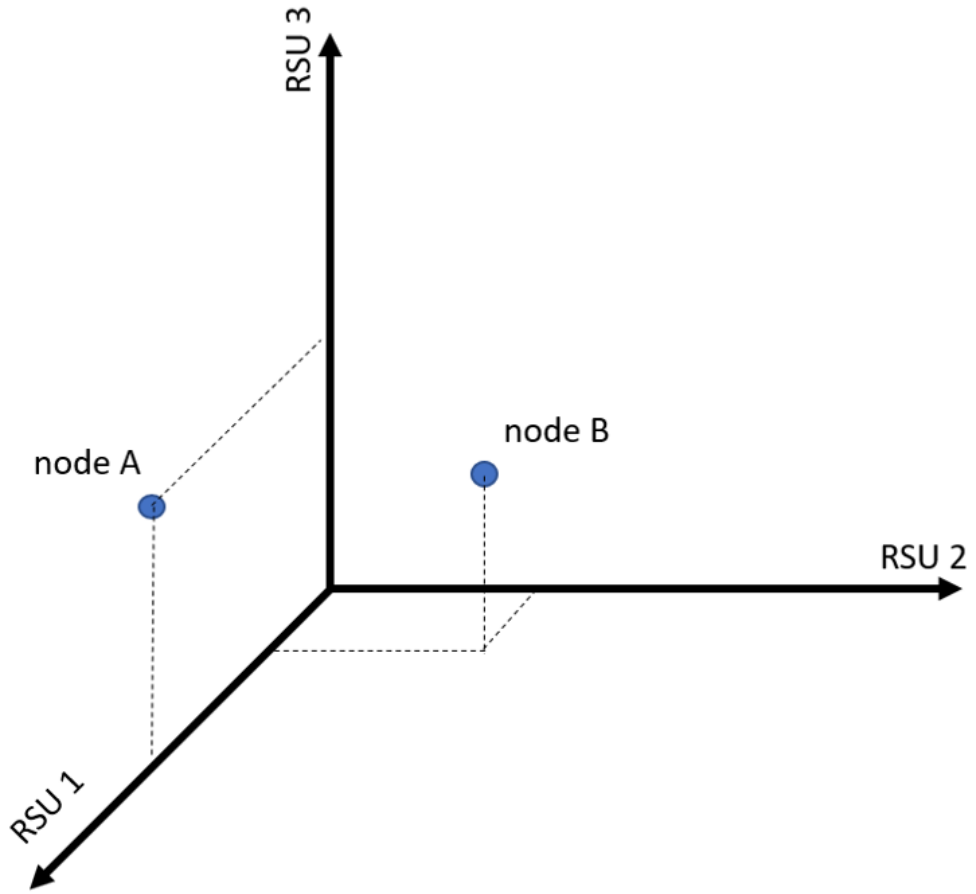
Edge Learning based Clustering (ii)

- Each node has a **data structure with a social strength** between it and each one of its contacts.
- Consider a network with nodes P, Q, S, and T
  - Node Q Node P
  - Q_P = 0,5 P_P = 1
  - Q_Q = 1 P_Q = 0,5
  - Q_S = 0,3 P_S = 0,7
  - … …
- These nodes can be represented in a Euclidean space with N dimensions
  - Each dimension represents the relations with a given node
  - The similarity between the nodes will be measured by **their distance** in this Euclidean space.



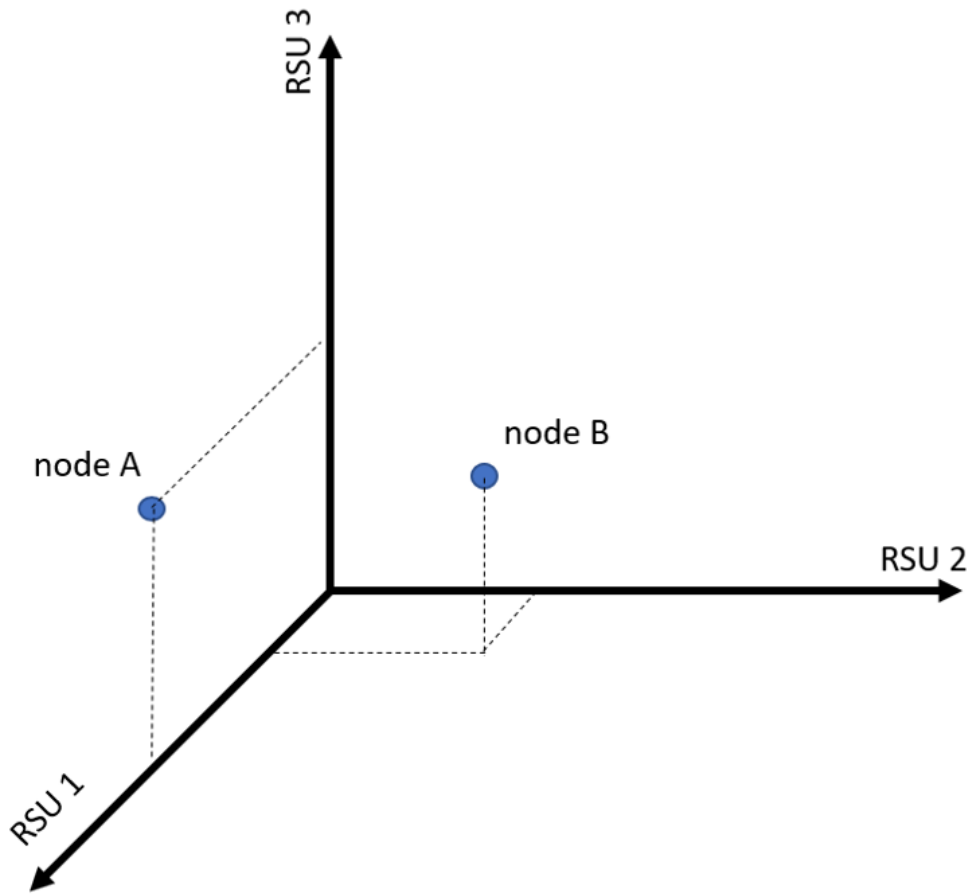
Edge Learning based Clustering (iii)

- Let
  - $\mathbf{y}_i$ be an embedding for node i
  - y_i^n be the n^{th} component of $\mathbf{y}_i$ for all $n = 1, \dots, N$
- The Euclidean distance is given by
  - $d_{i,j} = d(\mathbf{y}_i, \mathbf{y}_j) = \sqrt{\sum_{n=1}^{N}(y_i^n - y_j^n)^2}$
  - $d_{P,Q}$ is the similarity between nodes P and Q
- If RSUs store their social strengths
  - If the Edge merges all these data, it could be presented as a Euclidean space with N dimensions
    - where **N is the number of RSUs**, and each node would be represented by its social strength.
  - Node A has never connected with RSU 2, and node B has already connected with all RSUs.



Edge Learning based Clustering (iv)

- Based on this representation
  - which takes into consideration the algorithm complexity
  - for several mobile nodes that **is much higher than** the number of stationary ones, which is suitable for live computation
  - the K-Means algorithm is used to compute the clusters.
- This algorithm ran a predefined number of iterations each time it was called.
- The number of iterations is a balance between the resource consumption and the algorithm accuracy.
- The occupation of all clusters will be checked just after K-means finishes
  - if one or more of the clusters are empty, their centroid will be placed over a point of the largest cluster, and the algorithm will run a few more iterations.
  - There will then be no empty clusters, and their sizes will be balanced.



Edge Learning based Clustering (v)

- Once all the nodes are divided into clusters, the Edge will choose the Cluster Head (CH).
- The CH selection is based on
  - the **similarity** between each node and the centroid that represents the cluster
  - the **ego betweenness centrality** of each node in the network
- A logistic function is used in order to map the ego and the similarity values
  - This function is chosen, because only a small percentage of the nodes have an ego that is larger than 7.5, which should be the CH
- The main disadvantage of this approach is that when a node **does not connect with an RSU** for a significant time, it **gets closer and closer** to the origin of the Euclidean space
  - If this happens with a significant number of nodes, a cluster will contain **inactive nodes** that do not have any relations between them

Misbehavior Model

Sybil Attack

The attacker subverts the reputation system by creating a large number of pseudonymous identities and uses them to gain influence in the network.

Id Spoofing

A node that successfully identifies itself as another node or as having a different role in the network.

Not Working

Dropping or simply not sending the messages the node receives.

The Smart Monitoring System (i)

Intrusion Detection System

Misbehavior Detection

\+ may detect new attacks

\- may be trained by the attacker to lower its accuracy

Misbehavior Identification

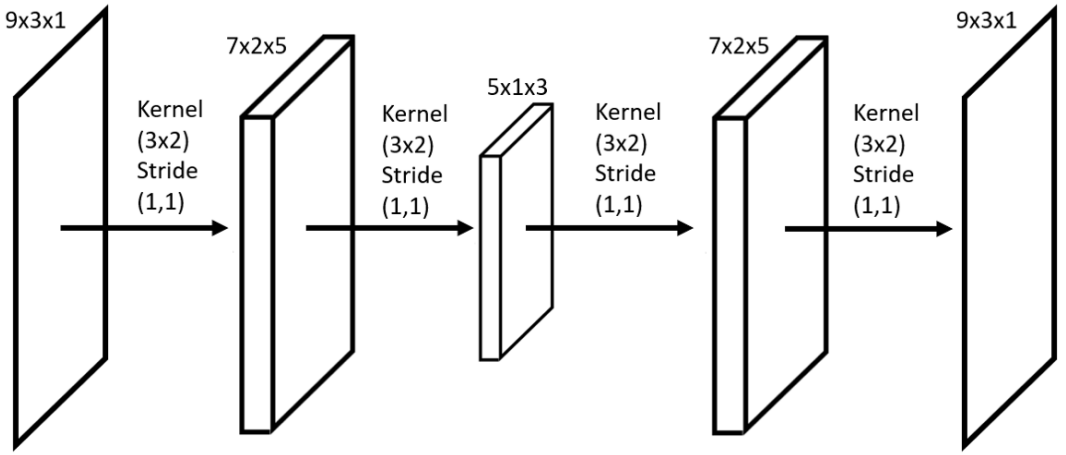
\+ high efficiency - low false positive rate

\- only detects known attacks

✓ N. Magaia et al., “An edge-based smart network monitoring system for the Internet of Vehicles”, IEEE ICC 2022.

The Smart Monitoring System (iii)

- Misbehavior Detection
  - The gathered data enters a **convolutional autoencoder** that uses the convolution property to simulate data dependency over time.
  - The performance of the algorithm is measured by the **mean squared error** between the input and the output data.
  - The autoencoder used is formed by convolutional layers.
  - All layers of the algorithm **have the same parameters**, a kernel of 3×2 and a stride of 1.

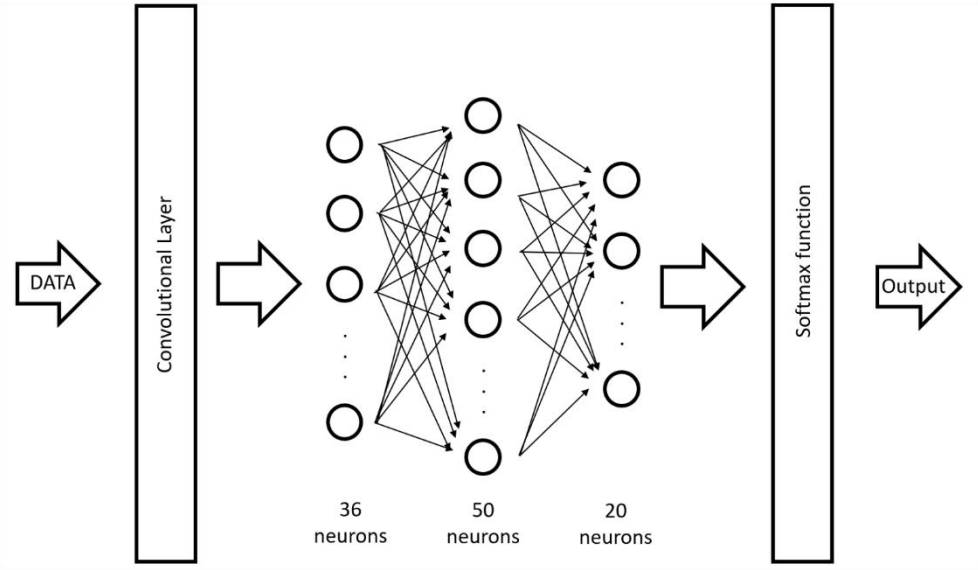


Output:

Reconstruction loss

The Smart Monitoring System (iv)

- Misbehavior Identification
  - This DL algorithm has the same input as the previous one but is now a **classification problem**
    - It has only **five different outputs**: Sybil attack, node not working, identity spoofing, normal behavior, and new unknown behavior.
  - It starts with a convolution layer to simulate the dependency of data over time
  - It consists of three dense layers and a **softmax function** to discern the labels.
  - The choice of the number of layers and the number of neurons **is a balance** between
    - the **flexibility** of the network and
    - the difficulty to train given that a deeper and denser network requires **more computation**



Output:

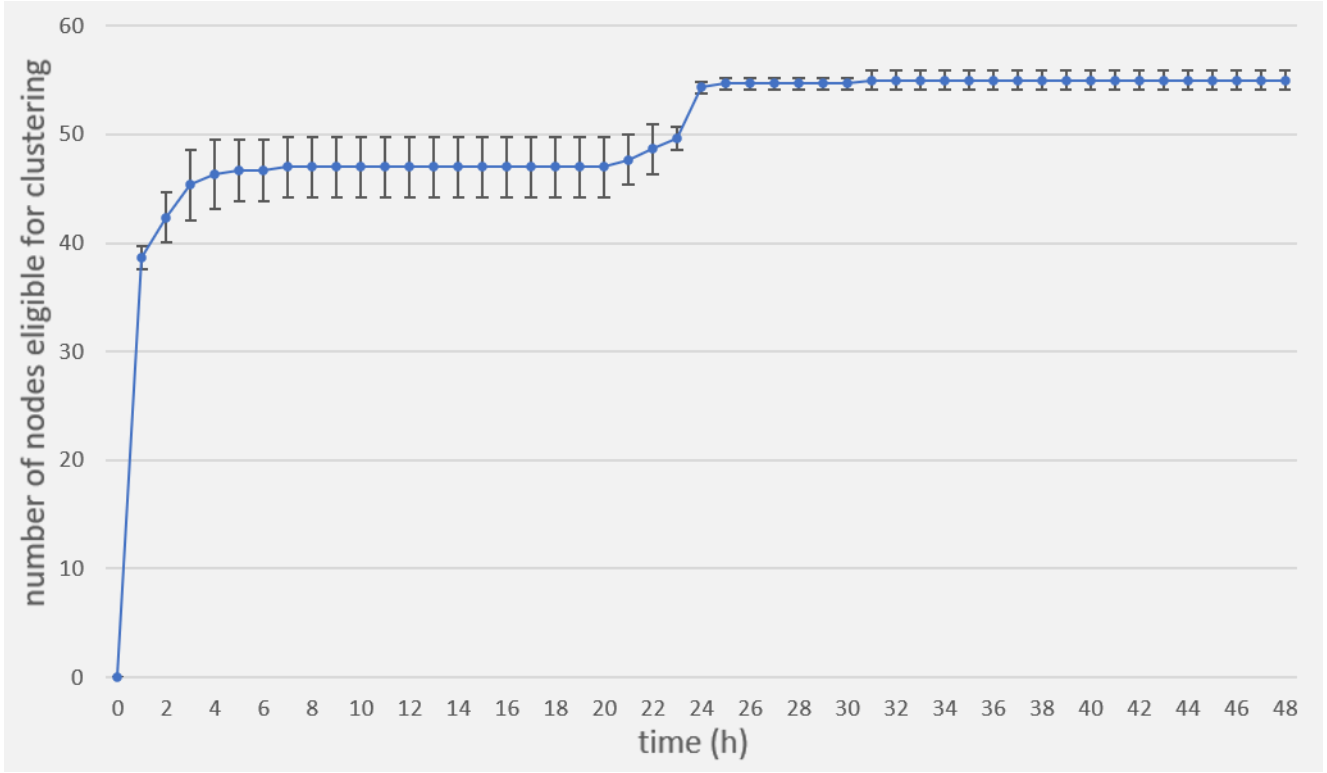
Behavior classification

Performance Evaluation: Simulation Model

- Simulator
  - The One Simulator
- Map
  - Helsinki City Center
- Types of nodes
  - Vehicles (i.e., Cars and Buses)
  - RSUs
- Movement Model
  - Workday Movement
- Communication Interfaces
  - V2V, V2I

Performance Evaluation: Results (i)

- We measured the number of nodes eligible for clustering and the number of cluster changes per hour
- Only the **nodes that already contacted an RSU** are known by the Edge and are eligible for clustering.
- During the first two days, the nodes start contact with RSUs
- The most significant changes occur in the morning and in the evening.

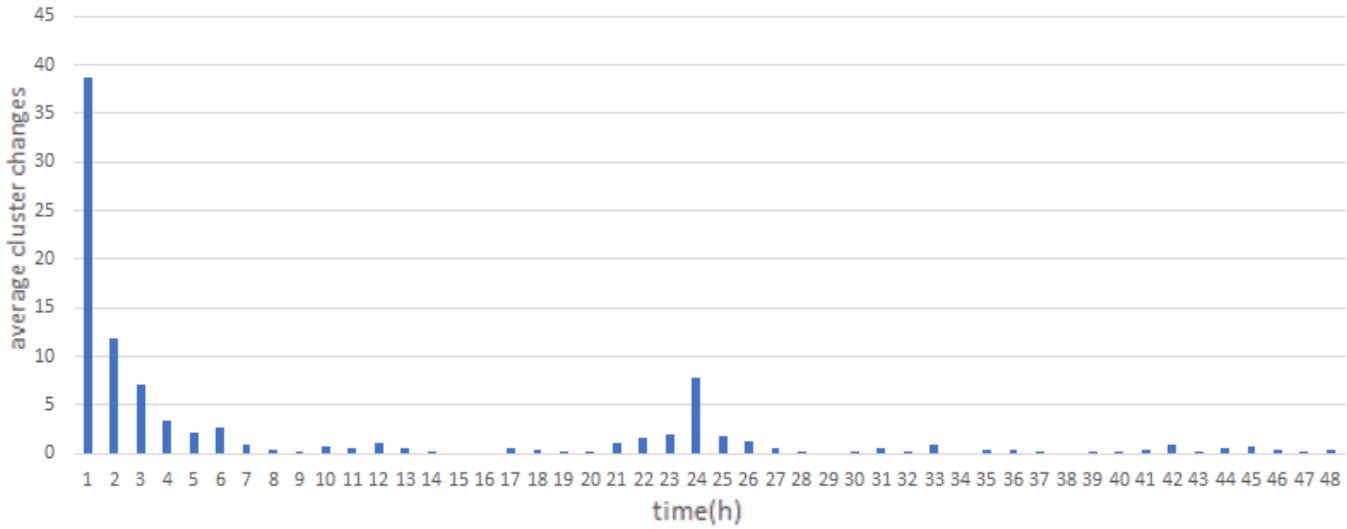


Performance Evaluation: Results (ii)

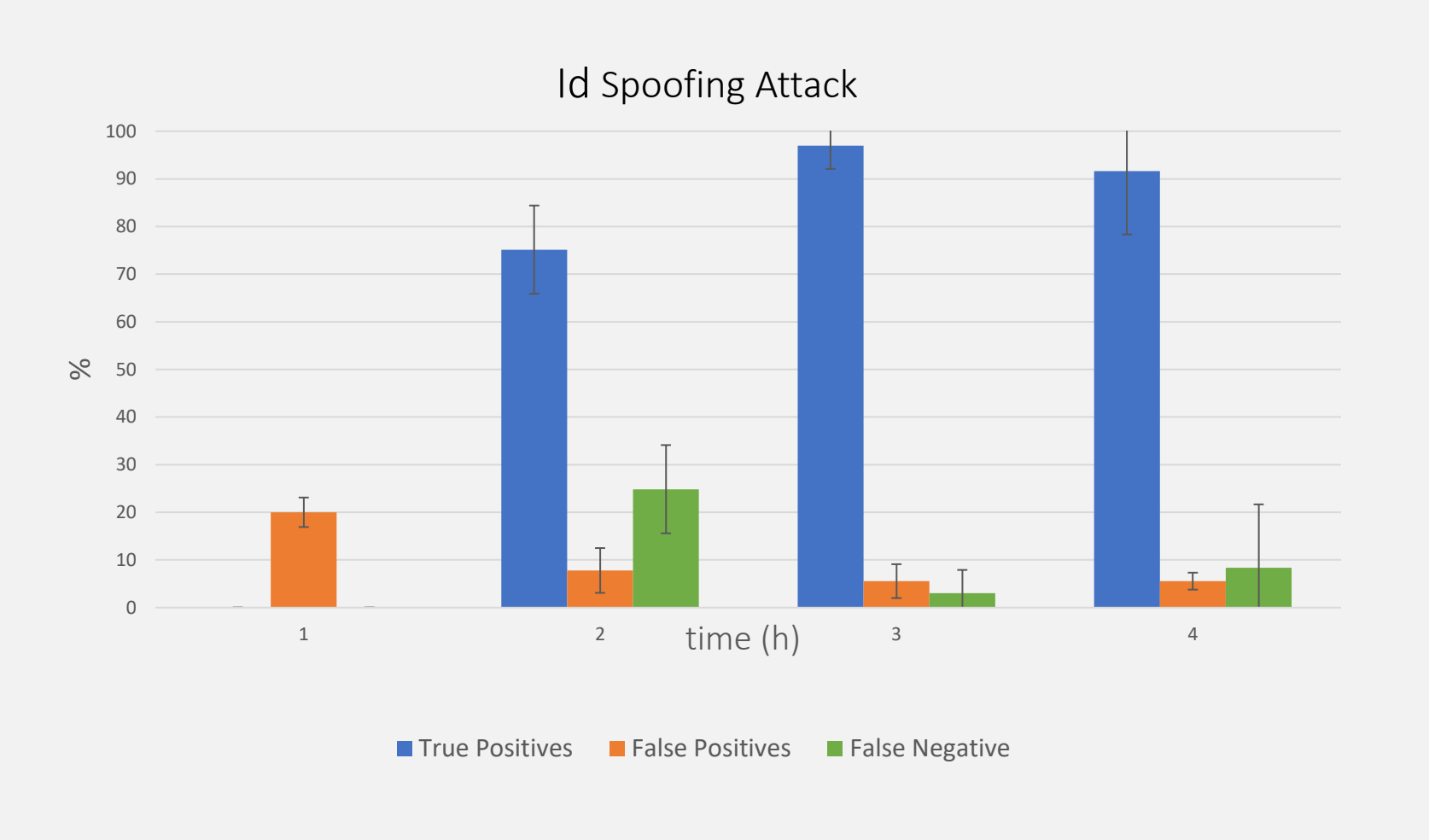
- Each color represents a different simulation configuration
- There are not any cluster changes counted in the first hour.
- During the second hour, there was a total of 65 cluster changes

Performance Evaluation: Results (iii)

- All the cluster changes are represented
  - nodes that connect for the first time with an RSU
  - the nodes that change from one cluster to another
- It takes an average of **six to seven hours** to reach a stable point.
- With the increase in the number of clusters, a decrease of the stability can be seen



Performance Evaluation: Results (v)



Performance Evaluation: Results (vi)

Main Scenario	Sybil	Id spoofing	Node not working
True positive (%)	68	95	52
False positive (%)	0.22	0.6	0.8
True negative (%)	98.78	99.4	99.2
False negative (%)	32	5	48

Conclusions

- An edge learning based clustering and edge-based smart monitoring system were proposed.
- Our proposed clustering, which is based on social relationships between the nodes, is an acceptable approach to the aforementioned problem
  - It provides a stable solution, which is due to the choice of the RSUs as anchors.
- The monitoring system is formed by two different deep learning algorithms
  - one to detect anomalies in the network, and
  - the other to identify these anomalies.
- This system is capable of identifying the behaviors studied and can be a great tool for any network manager.