Teaching Machines to Understand Urban Networks

Maria Coelho and Mark A. Austin
Department of Civil and Environmental Engineering,
University of Maryland, College Park, MD 20742, USA
E-mail: memc30@hotmail.com; austin@isr.umd.edu

***Abstract*—Next-generation urban systems will be enabled by technological (cyber) advances deeply embedded within the physical domain. The volume and variety of collected data in years to come is only going to grow and diversify, making the task of urban system design and management much more difficult than in the past. We believe these challenges can be addressed by teaching machines to understand urban networks. This paper explores opportunities for using recently developed graph embedding procedures to encode the structure and associated network attributes as low-dimensional vectors. These embeddings can be later used to advance various learning tasks. We exercise the proposed approach on a problem involving identification of leaks in an urban water distribution system. The Dynamic Attributed Network Embedding (DANE) framework is used to generate low-dimensional vectors for a water distribution network, whose pressure attributes are simulated with EPANET. The embeddings are then fed to a Random Forest algorithm trained to identify water leaks.**

Keywords-Systems Engineering; Machine Learning; Graph Embeddings.

I. INTRODUCTION

This paper is concerned with integrating recently developed graph embedding procedures with machine learning tasks that can enhance decision making in urban settings.

A. Problem Statement

Modern societal-scale infrastructures are going through an interesting time where the digital wave (e.g., the Internet, smart mobile devices, cloud computing) has opened up new avenues for enhancing the development of urban systems (e.g., transportation, electric power, wastewater facilities and water supply networks, among others) whose operations and interactions have superior levels of performance, extended functionality and good economics. While end-users applaud the benefits that these digital technologies afford, model-based systems engineers are faced with a multitude of new design challenges that can be traced to the presence of heterogeneous content (multiple disciplines), network structures that are spatial, multi-layer, interwoven and dynamic, and behaviors that are distributed and concurrent. In the past, engineers have kept these difficulties under control by designing subsystems that operate as independently as possible from each other. Today, however, it is acknowledged that subsystem independence and inferior levels of situational awareness imply sub-optimal functionality and performance. Communication and information exchange establishes common knowledge among decision makers which, in turn, enhances their ability to make decisions appropriate to their understanding, or situational awareness, of the system state, its goals and objectives. Overcoming these barriers makes future challenges in urban system design and management a lot more difficult than they used to be.

B. Scope and Objectives

Our work is motivated by the premise that next-generation cities are transitioning to an information-age fabric, where highly efficient sensing and communication technologies are deeply embedded within the physical urban domain. Present-day trends indicate that the flow and variety of urban data is only going to grow and diversify, making the task of system design, analysis and integration of multi-disciplinary concerns much more difficult than in the past.

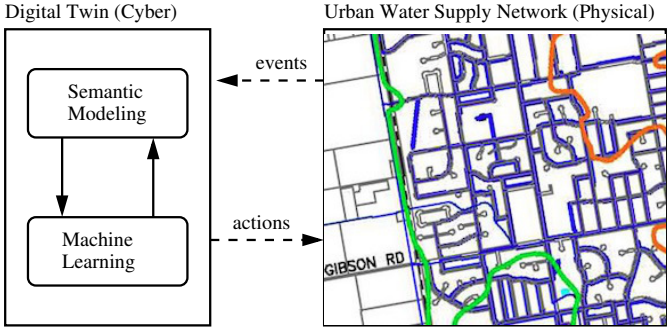

Figure 1. High-level representation for an urban water supply network digital twin (cyber) working alongside a physical urban water supply network.

As illustrated in Figures 1 and 2, we believe that these challenges can be addressed by combining Machine Learning (ML) formalisms and semantic model representations of urban systems that work side-by-side in collecting data, identifying events, and managing city operations in real-time. To this end, Figure 3 shows a preliminary classification of the strengths/weaknesses of AI/ML. The proposed approach builds upon our recent work in semantic modeling for (multi-domain) system of systems [1] [2] and exploration of a combined semantic and ML approach to the monitoring of energy consumption in buildings [3].

This paper explores opportunities for using recently developed graph embedding procedures to encode the structure and associated network attributes as low-dimensional vectors. These embeddings can be later used to advance various learning tasks. We exercise the proposed approach on a problem involving identification of leaks in an urban water distribution

 ISBN: 978-1-61208-771-9

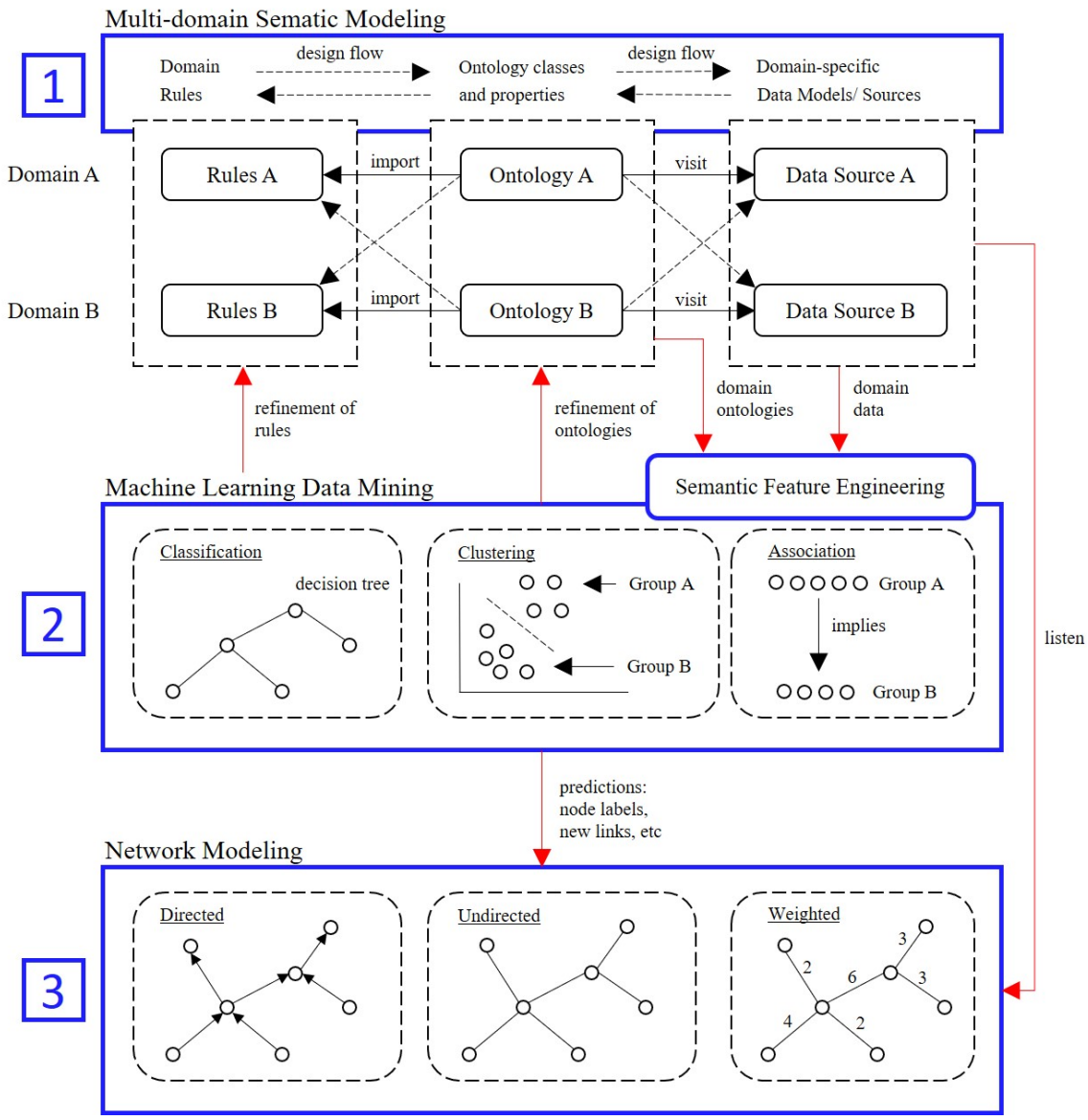

Figure 2. Digital twin architecture.

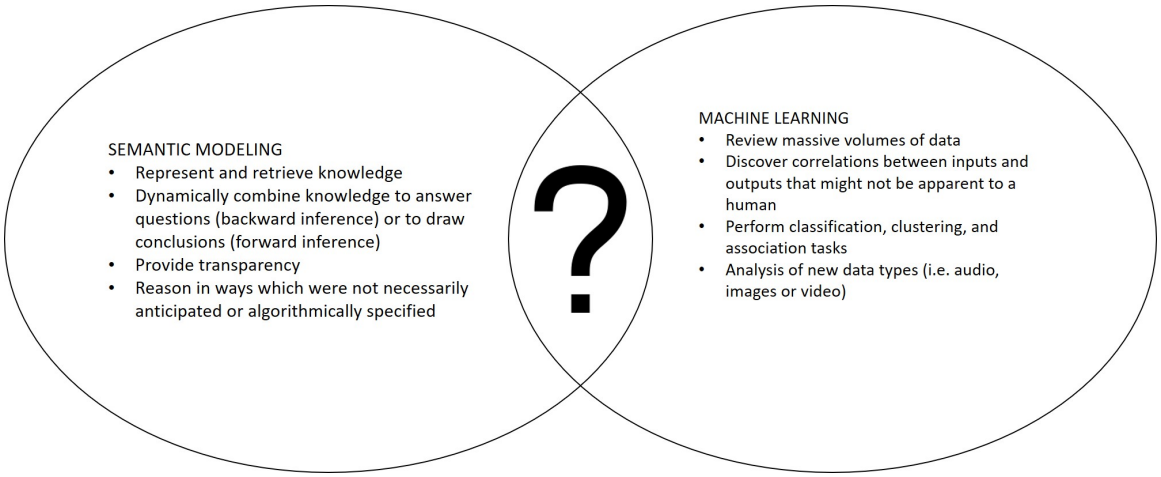

Figure 3. Venn diagram of semantic modeling capabilities versus machine learning capabilities.

 ISBN: 978-1-61208-771-9

system. The remainder of this paper proceeds as follows: Related work in ML algorithms for graphs is covered in Section II. Our work in progress is described in Section III. Conclusions and directions for future work are located in Section IV.

II. Related Work

This section covers the relationship of graph embedding procedures to our related work in integration of semantic modeling and ML approaches for the design of "city digital twins".

A. Architectural Template for Combined AI/ML

The proposed architectural template for a combined multi-domain semantic modeling and ML approach is shown in Figure 2. It is an extension of our work in 2018 [4]. Box 1 covers a framework for multi-domain semantic modeling, where concurrent development of ontologies, rules and data models placed on an equal footing. Box 2 shows ML for three classes of problems – classification, clustering and association – found in the data mining domain. Traditionally, ML approaches rely on user-defined heuristics to extract features encoding information about a graph (e.g., degree statistics or kernel functions). However, recent years have seen a surge in approaches that automatically learn to encode graph structure and attributes into low-dimensional embeddings, using techniques based on deep learning and nonlinear dimensionality reduction. Box 3 is the starting point for our investigation and the focus of this work-in-progress paper.

B. Graph Embeddings for Urban Networks

A prerequisite to network data mining is to find an effective representation of networks. Established network representations, such as adjacency matrices, suffer from data sparsity and high-dimensionality, and a lack of support for capturing semantics. During the most recent decade, however, there has been a strong surge of interest in learning to encode continuous and low-dimensional representations of networks as graph embeddings. Graph embedding provides an effective and efficient way to solve the graph analytics problem, by learning a continuous vector space for the graph, assigning each node (and/or edge) in the graph to a specific position in the vector space. This process provides users a deeper understanding of what is behind the data, and thus can benefit a lot of useful applications such as node classification, node clustering, node recommendation, link prediction, and so forth [5].

Embedding urban graphs into a low-dimentional space is not a trivial task. A key challenge in the design of graph embeddings for urban networks stems from the observation that the information to be preserved is strongly affected by the underlying characteristics of the graph. Urban networks may be homogeneous, heterogeneous, and carry auxiliary information modeled as attributes. Graph edges may be undirected, directed and/or weighted. In a comprehensive survey of graph embedding problems, techniques and applications, Hongyun and co-workers [5] propose two taxonomies of graph embedding which correspond to what challenges exist in different graph embedding problem settings and how the existing work address these challenges in their solutions.

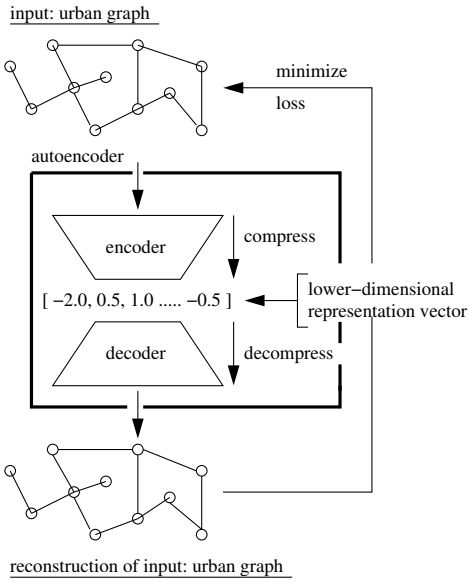

Figure 4. Traditional encoder-decoder approach.

One of the challenges described is capturing the diversity of connectivity patterns in the graph. When embedding a graph with topology information only, the connections between nodes are the target to be preserved. However, for a graph with edge weight or direction, the connectivity pattern provides graph property from other perspectives, and thus should also be considered during the embedding. Different types of objects (e.g., nodes, edges) are embedded into the same space in heterogeneous graph embedding. Therefore, another challenge is conserving global consistency and addressing imbalance between objects of different types. Some urban graphs (e.g., urban water supply networks) contain auxiliary information of a node/edge/whole-graph in addition to the structural relations of nodes (i.e., labels, attributes, node features, information propagation, knowledge base). The auxiliary information helps to define node similarity in addition to graph structural information. The challenges of embedding graph with auxiliary information is how to combine these two information sources to define the node similarity to be preserved.

In addition to the graph embedding input considerations, output format also pose challenges. Different types of embedding facilitate different applications. Output can be categorized into node embedding, edge embedding, hybrid embedding and whole-graph embedding. The challenge is determining suitable embedding output to meet the needs of a specific application or task. The task may be node classification, node clustering, node recommendation/retrieval/ranking, link prediction, triple classification, graph visualization, etc.

C. Autoencoders

Autoencoders are neural networks that are trained to reconstruct their original input. Figure 4 shows a high-level architecture for an autoencoder designed to work with graphs. First, an encoder takes a graph as its input and systematically compresses it into a low-dimensional (embedding) vector. The decoder then takes the vector representation and

 ISBN: 978-1-61208-771-9

attempts to generate a reconstruction of the original (graph) input. Encoder-decoder pairs are designed to minimize the loss of information between the input graph and the output (i.e., reconstructed) graph, and then use the embeddings for downstream ML tasks. These frameworks may be deterministic or probabilistic [6].

III. Work in Progress

In this section we exercise a graph embedding procedure that can encode both structure and network attributes on a problem involving the identification of leaks in an urban water distribution system.

Topic 1. Use Case

This use case aims to explore ML techniques for the detection and localization of leakages in very simple water distribution systems (WDSs). See Figure 5. Figure 6 is a flowchart of the process for detecting the location of the leak and taking actions to restore the system. We start by extracting a graph representation of the WDS and determining the initial hydraulic parameters of the system. The following topics describe: (1) the generation of hydraulic data, in particular node pressure, by the hydraulic simulation software EPANET [7]; (2) the preservation of the network topology and node pressure information in the encoding of node embeddings by the DANE framework [8]; (3) the training and testing of a Random Forest algorithm [9] with the node embeddings to infer leak location; and (4) the resulting performance obtained using this proposed framework.

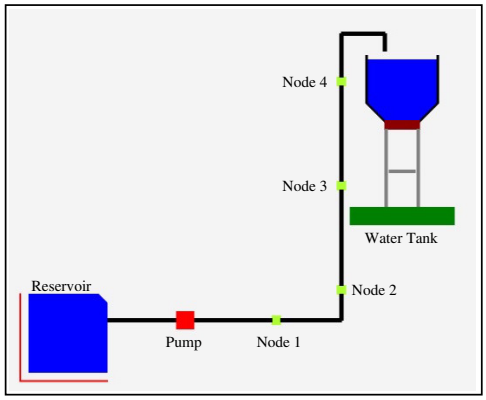

Figure 5. Elevation view of urban water distribution network and junction (node) numbers used by EPANET simulation.

Topic 2. Data Generation

ML algorithms for automatic water leakage detection requires training data. The data should involve hydraulics parameters at different locations in the WDS, pertaining to previous leaks that occurred in the past. However, for security reasons WDS data, which includes geographical layout of pipes, tanks, and demands are kept confidential by the water utility companies and are not readily available in public domain. Alternatively, the training sets can be generated by simulation of the pipe network under consideration. The simulation tool EPANET [7] can be used to achieve this goal

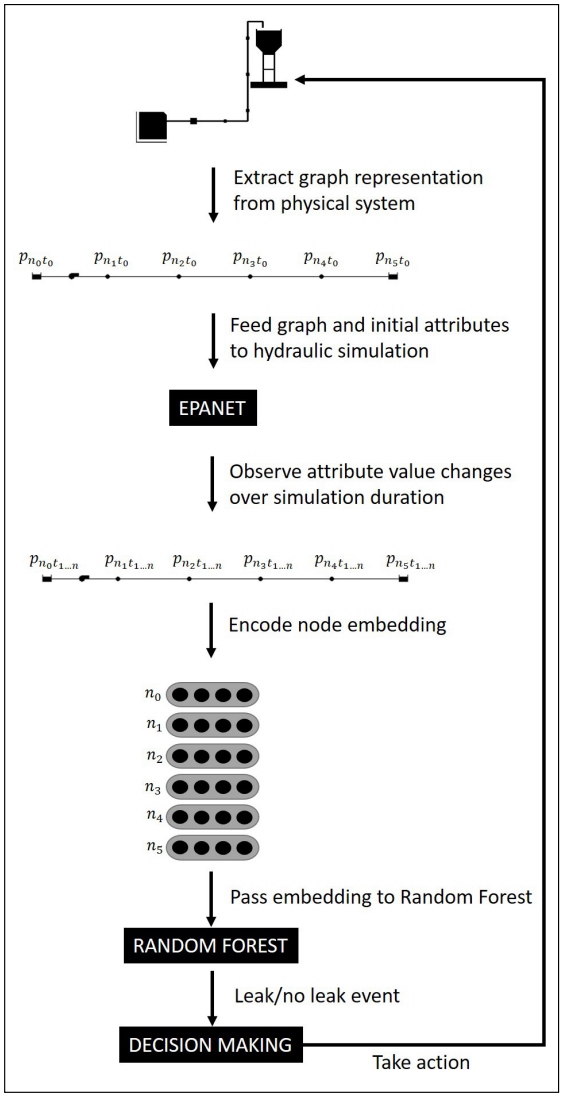

Figure 6. Process flowchart for training and executing machine.

[10]. EPANET is a computerized simulation model produced by the Environmental Protection Agency of the USA that predicts the dynamic hydraulic and water quality behavior within a drinking water distribution system operating over an extended period of time. Pipe networks consist of pipes, nodes (junctions), pumps, valves, and storage tanks or reservoirs. EPANET tracks the flow of water in each pipe, the pressure at each node, the height of the water in each tank, the type of chemical concentration throughout the network during a simulation period, the age of the water, and source tracing. A user can edit various characteristics of a network element and perform simulation to observe its effect on the overall system.

One of the main features of EPANET is that its hydraulic calculation engine is demand-driven. The water output data at each node is defined as the base demand. Although the software does not have direct tool to induce leakage in the system, it is still possible to model leaks as an additional demand, independent of the pressure in a consumption node. The demand can be increased at different times during the

 ISBN: 978-1-61208-771-9

simulation. The virtual WDS layout used to perform the simulations is shown in Figure 5. with location of the 4 junctions, 4 links, pump station, water source, and tank. In this work, we assume sensor nodes are deployed in each junction of the network; however, in our resource limited world, placing as many sensors as there are junction nodes to monitor all of the nodes in real time is extremely infeasible. An opportunity for future work would be to investigate how many sensor nodes are needed and where to place them in the network. Placement of sensor node would have direct impact on the efficacy of locating the leakage in the WDS.

In order to generate the large number of cases required for the ML training sets, the implementation of EPANET can be automated by developing a program which calls EPANET many times with varying leak locations. In this work, we will use the EPANET-Python Toolkit to perform this task. The toolkit is an open-source software, originally developed by the Flood Resilience Group (a multidisciplinary research group affiliated to UNESCO-IHE and Delft University of Technology), that operates within the Python environment, for providing a programming interface for the latest version of EPANET. It allows the user to access EPANET within python scripts. The toolkit is useful for developing specialized applications, such as water distribution network models that require running many network analyses. For simplicity, we will limit the parameter of interest to node pressure, although we recognize other parameters such as flow may be helpful in the indication of a leak as well. We obtain the pressure data by making some underlying assumptions: (1) The data obtained through simulation does not involve any noise in it (i.e., the sensors are ideal), (2) At-most one leakage can occur in the WDS in a simulation run, and (3) Water leakage is assumed to occur at the junction nodes only.

Topic 3. Node Embedding

With the data obtained from the hydraulic simulation through EPANET-Python Toolkit, graph embedding can be performed. In this use case we are interested in obtaining a low-dimensional node vector representation for each node in the network. The learned embeddings could advance various learning tasks, particularly leak detection by node classification. WDSs' networks are associated with a rich set of node attributes, and their attribute values are naturally changing, with the emerging of new content patterns and the fading of old content patterns. In addition, it has been widely studied and received that there exists a strong correlation among the attributes of linked nodes [11]. These node correlations and changing characteristics motivate us to seek an effective embedding representation to capture network structure and attribute evolving patterns, which is of fundamental importance for learning in a dynamic environment. In 2018, Li et al. proposed a novel DANE framework that first provides an offline method for a consensus embedding and then, in order to capture the evolving nature of attributed networks, leverages matrix perturbation theory to maintain the freshness of the end embedding results in an online manner [8]. Applying DANE to the pressure data outputted from EPANET simulation, yields a six dimensional node embedding vectors for each node. How to determine the optimal number of embedding dimensions is still an open research problem, thus we chose a set up for which the best results were reported.

Topic 4. Node Classification

With the node embeddings obtained from DANE, leakage detection can be performed. Leakage detection in this work pertains to finding the corresponding junction where the leakage has occurred, therefore the target function assigns a value of 1 to the node where leakage has occurred, and a value of 0 to the remaining nodes. The input and output data are prepared, and passed to a Random Forest classification algorithm. Random forest is considered a highly accurate and robust method because of the number of decision trees participating in the process. It does not suffer from the overfitting problem often encountered in other ML methods, since it takes the average of all the predictions, and cancel out the biases. Random forests can also handle missing values, by using median values to replace continuous variables, or computing the proximity-weighted average of missing values. It also provides the relative feature importance, which helps in selecting the most contributing features for the classifier [9].

The training set needs to capture as much of the expected variation in the target and environment as possible, therefore we generate training set from a simulation where all of the nodes are leaking for half of the simulation duration, and for the other half of the simulation duration none of the nodes are leaking. Since the simulation was set to last 24 hours, with pressure readings at every hour, the training set contains 24 cases for each node. Figure 7 shows the plots for each node in this scenario, where the first of the embedding dimensions is plotted against time. Note that at the time step where the leak occurs, the embedding value for that dimension changes; therefore, the problem can be framed as anomaly detection. In order to test the trained machine, we generate a test set from a simulation where none of the nodes are leaking for half of the simulation duration, and for the other half of the simulation duration only one of the nodes is leaking. Similar to the training set, the test set also contains 24 cases for each node. Figure 8 shows the first of the embedding dimensions plotted against time. Note that the embedding values change slightly compared to the previous scenario where all the nodes where leaking; therefore, the goal of the ML process is not only to detect the anomalies, but also identify which anomalies are actual leaks and which ones are just a propagation of the leak effects. Also note that we keep the leak duration constant through all simulations, since the initial objective of this work is not to identify when the leak occurs but where it occurs. However, we do aknowledge that the time domain is relevant and future work will need to address variations in not only space but time as well.

Topic 5. Preliminary Results

By training the Random Forest algorithm with both leak and non leak data for each node, we were able to test whether the algorithm is able to detect a leak in the system. The test was performed by feeding the algorithm data for a scenario where initially none of the nodes was leaking, and later introducing the leak only at node number 3, as shown in Figure 8. Classification problems are perhaps the most common type of ML problem and as such there are a myriad of metrics that can be used to evaluate predictions for these problems. Classification accuracy is the most common evaluation metric for classification problems, and it is the ratio of number of correct predictions to the total number of input samples. We

 ISBN: 978-1-61208-771-9

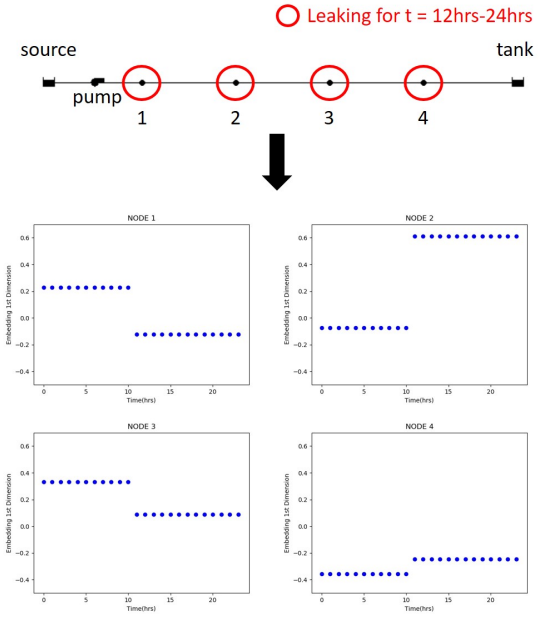

Figure 7. Node embeddings (1st dimension) obtained for train data set plotted against time.

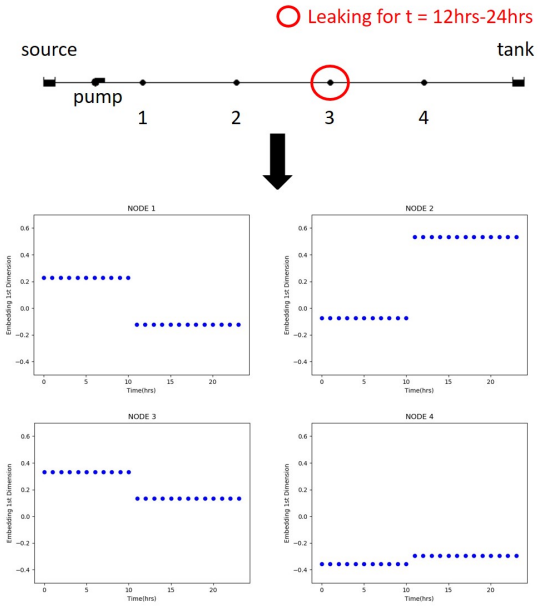

Figure 8. Node embeddings (1st dimension) obtained for test data set plotted against time.

found that the Random Forest algorithm used to train and test has a classification accuracy of 100 percent. We recognize such a high performance may be due to the simplicity of our network and the presence of only one leak when testing. Future work will investigate the influence of size and complexity of the network to the performance of the leak detection.

IV. Conclusions and Future Work

The long term objective of this research is to understand how ML and semantic-modeling can work hand-in-hand to enhance the collection of data, identification of events, and management of city operations. By exploring potential applications of ML to the identification of leaks in urban networks, this work work-in-progress paper takes a tiny step towards realization of the goal. Note that no validation set was used in this work because the simplicity of the network did not provide enough dimensionality to partition the cases into separate training, validation and testing sets without losing significant modeling or testing capability. In addition, we have used only one basic scenario for training and one for testing. Looking forward, our investigation will explore other possible simulations where different leak combinations and larger network sizes will be used. Future work will also explore the accuracy of the learned model when facing dynamic topologies, where edges are removed or created. We also aim to understand what types of graphs (e.g., undirected, directed, weighted) are easy for the ML to learn. Lastly, to the best of our knowledge, the DANE framework does not incorporate a decoder; therefore, extensions of the DANE framework to incorporate this capibility will be needed.

References

[1] M. Coelho, M. A. Austin, and M. R. Blackburn, “Distributed System Behavior Modeling of Urban Systems with Ontologies, Rules and Many-to-Many Association Relationships,” The Twelth International Conference on Systems (ICONS 2017), April 23-27 2017, pp. 10–15.

[2] ——, The Data-Ontology-Rule Footing: A Building Block for Knowledge-Based Development and Event-Driven Execution of Multi-Domain Systems. Proceedings of the 16th Annual Conference on Systems Engineering Research, Systems Engineering in Context, Chapter 21, Springer, 2019, pp. 255–266.

[3] P. Delgoshaei, M. Heidarinejad, and M. A. Austin, “Combined Ontology-Driven and Machine Learning Approach to Monitoring of Building Energy Consumption,” in 2018 Building Performance Modeling Conference and SimBuild, Chicago, IL, September 26-28 2018, pp. 667–674.

[4] M. A. Austin, P. Delgoshaei, M. Coelho, and M. Heidarinejad, “Architecting Smart City Digital Twins: A Combined Semantic Model and Machine Learning Approach,” Journal of Management in Engineering (Special Issue on Smart City Digital Twins), ASCE, 2019, (In Press).

[5] H. Cai, V. W. Zheng, and K. C. Chang, “A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications,” IEEE Transactions on Knowledge and Data Processing, vol. 30, no. 9, 2018, pp. 1616–1637.

[6] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation Learning on Graphs: Methods and Applications,” CoRR, vol. abs/1709.05584, 2017. [Online]. Available: http://arxiv.org/abs/1709.05584

[7] L. Rossman, EPAnet 2 Users Manual, January 2000, vol. 38.

[8] J. Li et al., “Attributed Network Embedding for Learning in a Dynamic Environment,” in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ser. CIKM ’17. New York, NY, USA: ACM, 2017, pp. 387–396. [Online]. Available: http://doi.acm.org/10.1145/3132847.3132919

[9] L. Breiman, “Random Forests,” in Machine Learning, vol. 45, no. 1. Norwell, MA, USA: Kluwer Academic Publishers, October 2001, pp. 5–32.

[10] J. Mashford, D. Silva, S. Burn, and D. Marney, “Leak Detection in Simulated Water Pipe Networks using SVM,” Applied Artificial Intelligence, vol. 26, May 2012, pp. 429–444.

[11] J. J. Pfeiffer, S. Moreno, T. La Fond, J. Neville, and B. Gallagher, “Attributed Graph Models: Modeling Network Structure with Correlated Attributes,” in Proceedings of the 23rd International Conference on World Wide Web, ser. WWW ’14. New York, NY, USA: ACM, 2014, pp. 831–842. [Online]. Available: http://doi.acm.org/10.1145/2566486.2567993

 ISBN: 978-1-61208-771-9