CS3907/CS6444
Introduction to Big Data and Analytics
Term Paper

Title: **An Overview of Machine Learning-Based Predictions Techniques Using Dynamic Graphs**

Submitted by: **Harshita Chadha (GWID – G40737617)**

"I affirm that this is my own work, I attributed where I used the work of others, I did not facilitate academic dishonesty for myself or others, and I used only authorized resources for this assignment, per the GW Code of Academic Integrity. If I failed to comply with this statement, I understand consequences will follow my actions. Consequences may range from receiving a zero on this assignment to expulsion from the university and may include a transcript notation."

An Overview of Machine Learning-Based Predictions Techniques Using Dynamic Graphs

1. Introduction

Graphs and graph representation learning have become an increasingly popular field of research in the past few years. The number of publications on these topics, as can be seen in Figure 1, has seen a tremendous surge in the past few years - a 1522.54% increase since the year 2000[1-3]. The widespread interest in this field can be attributed to the utility of graphs and their ability to accurately model real-life scenarios. Not only are graphs useful tools in the field of data analysis where they help visualize and analyze the nature of interactions between represented actors but they have also increasingly become integral to the field of machine learning and predictive analysis[4,5]. Traditionally, machine learning models have been trained on datasets that constitute instances inherently assumed to be independent of each other. However, as the complexity of applications increases, so does that of the subjects of predictive analysis, and thus more complex representations are warranted.

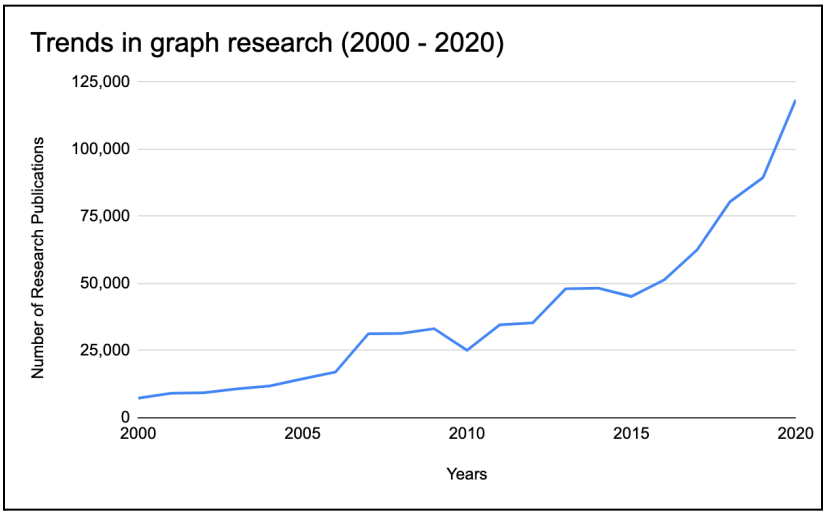

Figure 1: Increasing trends in graph representation learning publications in the last two decades

To substantiate the above point, the example of a simple content-based recommender system can be considered. In such a system the training examples typically consist of the past behavior of the customer and each training example or customer data in the training set is assumed to be independent of another. However, if user data is modeled as a graph and the underlying

assumption that each data point is independent of the other is ignored, then the interactions and similarities that exist between each user may be captured more effectively. The consequent increase in information available leads to more personalized recommendations and enhanced user experience. In fact, state-of-the-art recommendation systems such as those employed by big corporations like Netflix and Amazon make use of network graph models to personalize user experience[6,7]. Thus, it can be concluded that graphical data as an input to machine learning algorithms helps make more accurate predictions for complex real-life situations simply because more information is captured.

A majority of the current machine learning techniques for graphs work under the assumption that the underlying structure used to make predictions is static in nature but this is rarely true of real-life interactions [8,9]. For instance, in medical applications of machine learning, most of the captured patient data is time-variant and in fact, it is these timed variations that most often carry characterizing information. This is where dynamic graphs come into the picture. In addition to capturing the spatial information about the scenario they model, dynamic graphs also have a time component that helps capture temporal fluctuations of information. In terms of applications of dynamic graphs in predictive analysis, one way to handle this is to apply static graph machine learning techniques to time-variant graphs, however, the results obtained are more often than not sub-optimal. This warrants the need to develop dedicated frameworks capable of working with dynamic graphs. The field of the application of machine learning algorithms to dynamic graphs, however, is fairly new and most of the work focuses on extending static techniques by treating dynamic graphs as discrete time-stamped snapshots - a practice that is restrictive in nature. Only very recently techniques have started to emerge that treat dynamic graphs as continuous time entities that constantly evolve[10].

The objective of this term paper is to study some of the most recent machine-learning techniques that have been applied to dynamic graphs to obtain useful predictions. The paper is divided into five main sections. The present section serves as an introduction to the problem statement, this is followed by section two where a discussion of the importance of graphs to big data and analytics is done. In section three, an overview of dynamic graphs, major modeling techniques, and their areas of application has been provided. In section four, we present the latest machine learning

frameworks that have been proposed for dynamic graphs and carry out a detailed discussion about graph representation learning algorithms and graph embedding strategies. Section number five serves to present the major conclusions drawn from the study undertaken and also discusses the future line of work that can be carried out in this domain.

2. Background

Graphs are a way to model real-life situations more accurately and represent them in the form of ordered pairs of interactions with several parameters specified to indicate the nature of such relationships. Traditionally, a graph can be represented as G(V, E) where the symbol G is used to refer to the created graph, V stands for the vertices which are often real-life entities modeled by the graph. E is usually a set containing ordered pairs that symbolize the interaction of the vertices with each other[11].

Graph analytics is a subdomain of big data analytics that deals with graph-based algorithms that are used for the identification of relationships and their strength in a given network. The main idea in graph analytics is to both observe the relationships between the different vertices as well as capture the structural characteristic of the graph as a whole. In the case of dynamic graphs, a third objective which is to study the evolution of these structures and relationships over time is included. Further, graph analytics can also be used to optimize access paths between various vertices[12]. There are many applications in this field such as social network analysis, security anomaly detection, recommendation systems, etc. The algorithms that can be used in graph analytics include clustering or grouping together of similar objects based on labels attached to graph vertices, partitioning which can help in the identification of weak spots in graph connectivity, searching or finding a particular element in a graph, the shortest path to find optimal layouts, connected components to find strongly connected regions useful in social network analysis, page rank - popularity measure of web pages can be transferred to social network analysis also[13]. This information is also represented in Table 1 below -

Table 1: Classes of Graph Algorithms and some application areas

Class of Algorithms	Graph relevant description	Applications Areas
Clustering	These algorithms help in the identification of similar actors in a graph using specified node attributes.	In recommendation systems to personalize user experience.
Partitioning	This class of algorithms helps in the identification of weak link spots in the context of graphs.	In the configuration of network layouts and fault localization in networks.
Search	These algorithms help find optimal ways to localize a given actor in graphs.	Traversal of graphs to find relevant nodes.
Shortest Paths	These algorithms help in finding optimal paths.	To configure the layout of roads and bridges, optimal route configuration, etc.
Connected Components	These algorithms help in the identification of strongly connected regions of a graph.	In social media analysis to find the most influential neighborhoods.
Page Rank	These algorithms help in the numerical estimation of the importance of nodes in graphs.	To find influential actors in social media analysis for marketing and business applications.

In historical applications[14] of big data, the focus was to study past behavior and make disjoint predictions based on that. But in recent years, there has been a shift to model experiences instead. Graphs help add context to data or in other words, mathematically capture some essence of experience which can help provide insights into the modeling task and lead to better prediction and/or inferences. Figure 2 below illustrates the market share of the big data and analytics industry by verticals.

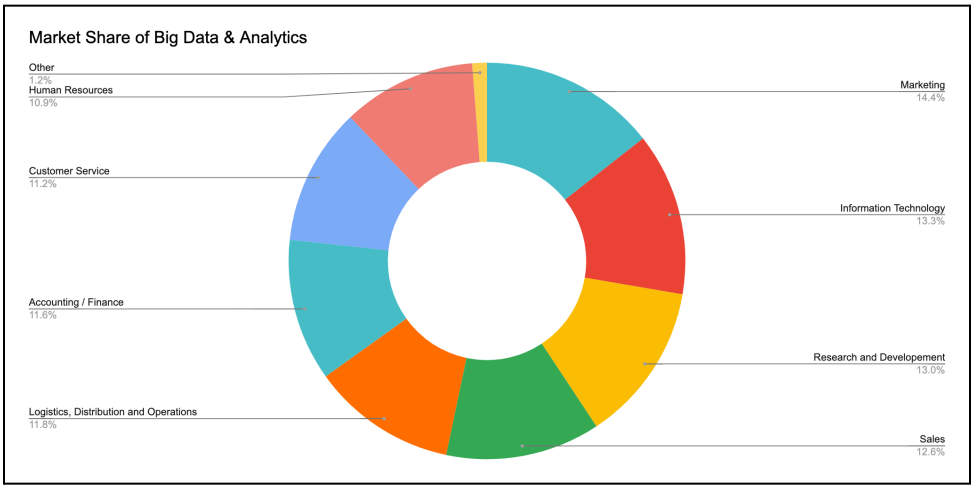

Figure 2: Plot of the usage of Big Data and Analytics in Various Industries [15]

As can be seen from the pie plot, the major applications of big data and analytics are in industries that are driven by user experiences and as stated above, user experiences can be modeled better by capturing interactions and context and graphical representations are able to achieve this. To put things into perspective, the example of social media analysis[16] can be considered where the interactions of users can be modeled as a graph and simple metrics such as a node's in-degree, out-degree or the connectivity of a subgraph can help identify influential groups and actors in the social media ecosystem being modeled. Not only does this help understand the flow of information and gain insight into opinion formation, but more often than not such information is also important from the marketing perspective.

Therefore in conclusion, it can be said that because the usage of big data is moving more towards end-user applications, and because graphs model such ecosystems better, a shift towards graph analytics and graph-based machine learning research has been on the rise in the last couple of years.

3. Brief Overview of Dynamic Graphs

When graphs become temporal in nature or capture the evolution of the relationships they represent over time, they are called dynamic graphs. These are mathematically represented as a series of static graph snapshots over time $G = (G1, G2, G3...Gt)$ where each $G_t = (V_t, E_t)$ and t is representing the time component[11].

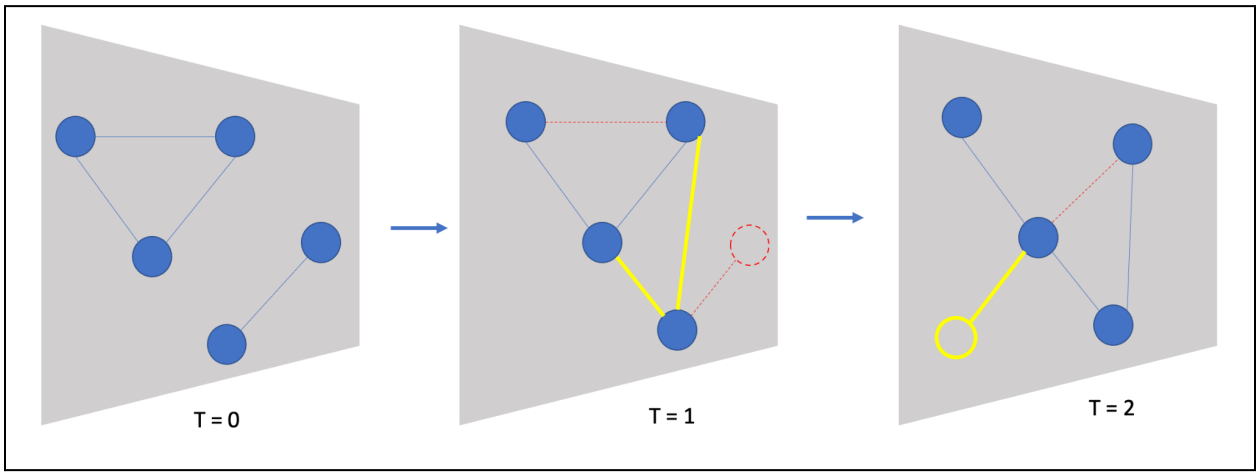

Figure 3: Dynamic graph event evolution represented as a set of timed static snapshots [17]

Thus, dynamic graphs are essentially a set of static graphs spread over time [11]. As their name suggests, in a dynamic graph environment, new nodes can appear, new relationships or edges may appear and similarly, both can also disappear. A representation of these instances in the case of a social network of evolving friendships is shown in Figure 3.

3.1 Types of Dynamic Graphs

Dynamic graphs capture a wide variety of information about a given situation including attributes, interactions, lifespans, etc. and different applications warrant different types of information at varied granularities[18]. The two major categories of dynamic graphs are discrete vs continuous dynamic graphs. In discrete graphs, data is captured at fixed and regular time intervals. In continuous time graphs, on the other hand, a static graph representing the initial state at time t_0 followed by a sequence of temporal observations/events as they occur is captured. In this case, each observation is a tuple of the form (event type, event, timestamp) where the event type can be a node or edge addition, node or edge deletion, node feature update, etc., the event represents the actual event that happened, and timestamp is the time at which the event occurred. A taxonomy of dynamic graph types is illustrated in the figure below -

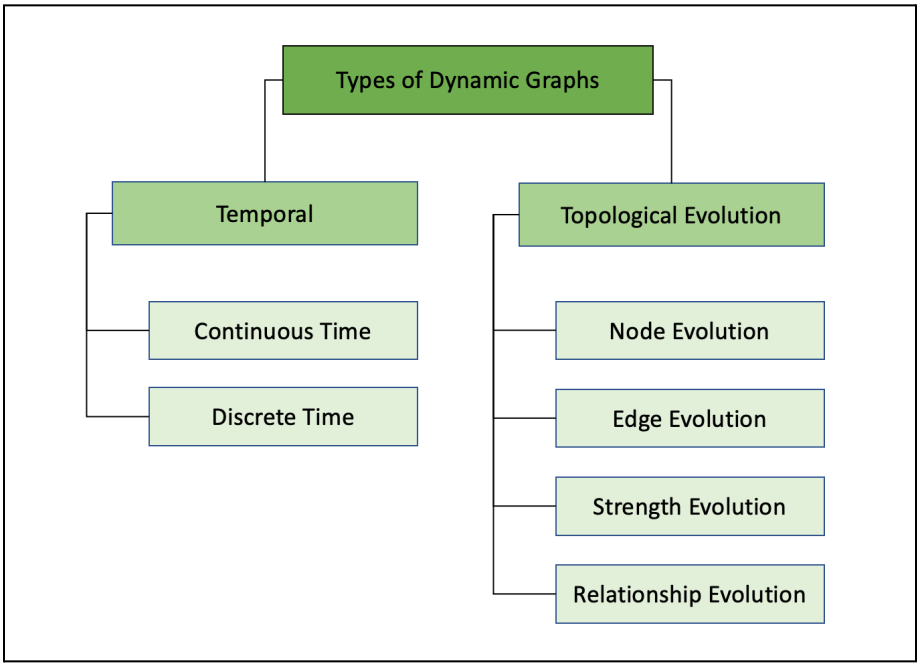

Figure 4: Visualization of the types of dynamic graph representations

Secondly, based on the nature of evolution, dynamic graphs can further be classified into five sub-categories - node addition and deletion evolution, feature updates - the changes in nodular attributes, edge addition, and deletion - the modification of relationships over time, edge weight updates - to capture the evolving strength of relationships between the nodes, and relationship updates - the nature of the connections evolving over time[19].

3.2 Dynamic Graph Modeling Techniques

Broadly speaking, dynamic graphs can be represented mathematically or modeled as either a stream of timed events or in the form of ordered pairs of evolutions. The most common way of representation is in the form of ordered pair of static snapshots where each dynamic graph is represented as a set of static graphs with time stamps. Another way to model dynamic graphs is through a difference-based technique where two main components are stored - the first is a static graph at instance t=0 and the second is a set of timed topological evolutions.

This method exploits the sparsity of topological evolutions in time. In other words, it can be said that for any real-time event that a dynamic graph models, it is important to capture the modifications or irregularities that arise since it is these “blips” that contain the most valuable information. Furthermore, since the number of modifications or irregularities is not very big, in the case of a large dataset, it makes sense to store the huge invariant chunk of data once and subsequently to only store the changes which are most often stored as lists of adjacency matrices. In addition, dynamic graphs may also be stored in the form of continuous time models where the new additions and departures themselves are time-stamped and the link streams are charged with node interaction representation[19].

3.3 Dynamic Graph Visualisations

As demonstrated in section 3.1, dynamic graphs can be visualized as time series plots which are time-dependent snapshots of static graphs. However, there exist other more intuitive ways that can be used to visualize such graphs. Since dynamic graphs can offer insights into problem domains even through simple visualization, it is important to offer a brief overview of the various techniques currently employed to do so.

One of the ways in which this can be done is through animation. In such a case, the nodes and their interconnections are shown to be moving objects and most commonly the time evolution is represented using modifications in shapes, colors, and sizes of the components in a dynamic graph. Another way is to use heatmaps that model the varying intensity of interactions between dynamic graph actors as a series of evolving colors in a two-dimensional plot. Streamgraphs are another specialized way of dynamic graph representation where stacked area plots with nodes and their consequent edges are shown as color-coded bands. Further, the most intuitive way of visualizing dynamic graphs is the use of online interactive visualization tools where plots can be zoomed into, they can be panned, and graph characteristics at different granularity levels and time instances can be observed[20].

3.3 Dynamic graph application areas

The applicability of dynamic graphs to model modern-day phenomenon is quite vast. For instance, one of the most researched subdomains of machine learning - computer vision can also benefit from the application of dynamic graphs. Here, dynamic graphs can be used in tasks such as activity recognition and object tracking as they can help in the modeling of time-dependent relations amongst the actors in video sequences. Dynamic graphs can also be used for pose estimation tasks where these graphs can help model the temporal relationship between the orientation of body parts in real-time[21-23]. Another important application in computer vision can be for real-time event tracking systems that can be of great importance in monitoring crowds and traffic accidents. Video summarization is also another subdomain of application in the larger computer vision field. Further, dynamic graphs are also useful in natural language processing tasks, an application that might not be intuitive at first glance. These graphs, however, can help in document analysis - by modeling the correlation of words and phrases, in sentiment analysis to study the evolution of emotion over time, and in dialogue systems and text generation tasks[24, 25].

Apart from applicability in subdomains of machine learning, dynamic graphs also have many uses in real-world task-specific applications. For instance, in the field of traffic monitoring and mitigation dynamic graphs when coupled with various machine learning technologies can help in traffic flow prediction, congestion detection, route optimization, public transport scheduling

incident management, etc [26, 27]. In the healthcare industry, these graphs can help in the epidemiological modeling of disease spread, patient monitoring systems, healthcare network analysis, public health surveillance, drug discovery, etc [28, 29].

Currently, dynamic graphs are most commonly used in recommendation systems to provide personalized suggestions, analyze trends, make context-aware recommendations, and make cross-domain suggestions. Further, they are also most commonly used in social network analysis where these graphs help in the detection of communities, influence analysis, anomaly detection, diffusion analysis, etc [30, 31].

In general, dynamic graphs can be applied to any domain that can benefit from insights provided by the following general prediction problems - (1) node classifications that help to find classes for nodes and group similar occurrences together (2) graph classification to look at the overall graph and classifying similar interactions together (2) link prediction to predict the likelihood of new relationship formation (3) time prediction to predict when an event happened or when it will happen. In addition to the above-mentioned prediction problems that are transductive in nature i.e., which predictions are made about entities that are present and observed when training is taking place, dynamic graphs and many of the proposed machine learning algorithms can handle inductive problems as well - these are those problems where predictions about unseen entities or new graphs have to be made.

4. Dynamic Graphs and Machine Learning

Traditional machine learning techniques assume that data samples are independent in nature and are designed to take in attributes about a particular entity and make predictions based on these attributes. Consequently, inputs to most of these models are flattened vectors or tensors. Most machine models are thus not equipped to handle the complex information that is modeled by a graph. In order to make graphs compatible with machine learning tasks, initially graph kernels were used (for static graphs). These kernel methods are mostly used in classification analysis wherein the inner product of two graph inputs is usually computed to measure their similarity metric. These similarity measures can then be plugged into machine learning algorithms such as support vector machines or SVMs for probabilistic predictions.

These kernel methods have, however, recently lost popularity and another methodology employing graph neural networks are being used. Graph neural networks are, as their name suggests, a graph-in-graph-out neural network architecture designed to take in as their input graphs and produce optimizable statistical mappings of these graphs called embeddings. These embeddings are simplified lower-dimensional representations of graphs that capture all the spatial relationships and maintain the overall graph connectivity information. Graph embedding can be thought of as being analogous to feature engineering tasks. Just like in the case of feature engineering, in graph embedding, the input is transformed into a more usable representation that can be suitably utilized by the machine learning algorithm to make useful predictions.

However, as stated earlier, in most complex applications it is integral to capture the nature of the interactions between these entities over time to be able to make better predictions and this is where dynamic graphs come into the picture and the complexity of the embedding task increase manifold. Traditionally for static graphs, graph neural networks as described above are used to make predictions, and a large chunk of research is now focused on the transferability of GNNs onto dynamic graph space but in addition to this, novel embedding mechanisms and even embedding free prediction architectures for dynamic graphs are also being developed. In the section that follows, we present an overview of the most prominent techniques being applied to dynamic graphs to obtain useful ML-based predictions from them.

4.1 Prominent Machine Learning on Dynamic Graphs Architectures

In general machine learning techniques that are applicable to dynamic graphs can be broadly classified into two groups - first dynamical algorithms and second graph representation learning algorithms. In dynamical algorithms, the first step is to produce a first solution for a given static graph - this is called preprocessing - and then for any subsequent updates in the graph, the solution space is also suitably updated. This is in direct contrast to representational learning-based techniques where the idea is to embed the given input dynamic graph into vectors and then perform machine learning computations on these embeddings - the process of embedding production can be machine learning based or not. Section 4.2 that follows discussed dynamic graph embedding techniques in detail.

In the case of dynamical algorithms for node clustering tasks, Dhanjal et al [32] in their work present a spectral clustering algorithm for nodes that constitute a dynamic graph. They primarily employ the single value decomposition technique of rank k on a laplacian matrix of the given dynamic graph a process that as its result produces a novel node eigenspace for each of the nodes in the graph. These nodes represented in the new eigenspace are then grouped together or clustered. To incorporate a dynamic graphs temporal component, the framework updates the singular value decomposition of the graph laplacian to account for the changes and consequently, the cluster labels for nodes are also subject to changes and updated. Gorke et al [33] utilize the concept of a partial minimum cut tree to achieve graph clustering. Here, given an initial weight, an artificial node is inserted into the given graph and is connected to all other network nodes at a time instance with the given weight value. Using this, a minimum cut tree of the graph is obtained and the artificial node is removed from consideration thereby facilitating clustering by utilization of connected components of the minimum cut tree. In their algorithm, a partial minimum cut tree is first dynamically updated as is the case with dynamic graphs, and this update is used to update the clustering.

For classification tasks, the first algorithm for node classification in dynamic graphs was proposed by Aggrawal et al [34] and was titled DYCOS. This algorithm used random -walk and made use of both the node linked and the content of nodes for the classification task. Yao et al [35] proposed a support vector machine classification algorithm for dynamic graphs where the label update only occurs in case of a new node addition or an edge addition in the dynamic graph. At the core of their methodology, the support vectors learned from the previous set of nodes and edges that are then used to predict the class labels for all future nodes, and the model is updated after each batch addition (new node and edge addition). Further, Xu et al [36] proposed AdaNN that is an aggregation-based framework that computes and endeavors to learn node attributes by aggregating both node and neighbor attribute information. The random walk method is used to gather structural information about the node's neighborhood.

In the case of graph representation learning-based methods, the architectures, unlike in the case of dynamical algorithms can be divided into two major parts - the encoder and the decoder. The

encoder mainly works to produce embeddings of the dynamical graph and the decoder is usually a classic prediction model such as a multi-layer perceptron model (MLP) that uses the embeddings to make suitable predictions. The architectures described henceforth are all representation learning based and employ the same basic aforementioned methodology.

Temporal Graph Network or TGN as proposed by Rossi et al [11, 37] is a general-purpose encoder architecture that has four main components. The first is the memory which is analogous to hidden states in Recurrent Neural Networks or RNN and is charged with the task of representing the past interactions of a node. The second component is a message function that captures node interactions and encodes them as a vector for each of these nodes in memory. The memory updater is the third component that embeds the new messages computed into the memory. A major contribution of their work is that they tackle the staleness problem - the case where many of the nodes are not frequently changing state/interacting. They do this by using a module called embedding that performs aggregation on the neighbors of the inactive nodes and on the memories of the most likely active neighbors and this helps them compute an up-to-date embedding for the nodes. To train the TGN, another module namely the raw message store is introduced that helps prevent leakage during training. When the performance of their model was compared with other baseline techniques, the success observed was attributed to the existence of the memory component that reduces computation and complexity three folds and to the embedding module that tackles staleness.

Ma et al [38] propose a dynamic graph neural network framework that mainly constitutes two parts - the update and propagation components. It extends the idea of static graph neural networks and endeavors to learn graph embeddings progressively and apply the proposed technique on dynamic graphs to capture the evolution of phenomena modeled therein. The framework is designed to automatically update information present in the attributes of the nodes by gathering information about the edge interactions and the time intervals between the edge propagations and encoding this into the embeddings. They evaluated the performance of their model with tasks such as link prediction, node classification, etc and these experiments echoed the superiority of performance.

Gunarathna et al [39] proposed GTA-RL (Graph Temporal Attention with Reinforcement Learning) wherein they focus their attention on dynamic graphs that capture combinatorial problems such as that in the case of a ride-sharing scenario. They propose a novel encoder-decoder architecture that uses a multi-dimensional attention methodology to embed both the temporal and the spatial features of a graph in parallel and then employs a fusion mechanism to learn the interdependencies between the consequent embeddings. These are then arranged in a series and fed into a temporal pointing decoder that pays attention to the spatiotemporal representation for the decision-making stage The GTA-RL model undergoes training by virtue of a reinforcement learning-based policy-gradient algorithm. To address situations where the dynamic changes are not already known, they partially embed their encoder inside of the decoder thereby making GTA-RL highly relevant to real-time decision-making applications.

An architecture entitled NEDGNN or the Novel Neighborhood Extended Dynamic Graph Neural Network was proposed by Yu et al [40]. Their architecture focuses on capturing the temporal evolution of the graphical data as a whole and localizing the modifications in the neighborhoods therein over time. The uniqueness of their architecture lies in the introduction of a novel module entitled the temporal attention propagation module. This part of their architecture is responsible for spreading information messages to the n-hop neighbors of a node and it utilizes a self-attention mechanism in order to achieve that. Further, in order to enhance their architecture's time-space performance, a FIFO (First-In-First-Out) message box module is used.

An Evolving Graph Convolutional Network Design for Dynamic Graphs or EvolveGCN for short was proposed by Pareja et al [41]. Their architecture is in direct contrast to most mainstream models where the main goal is to use a graph neural network or GNN to extract embeddings from an input dynamic graph and then use a pipelined Recurrent Neural Network or RNN to make predictions based on the information contained in the embeddings. Instead, EvolveGCN uses the Recurrent Neural Network not to make predictions from embeddings but to actually propagate time-based evolutions as modifications back into the graph neural network where these messages are used to evolve the network itself. This methodology helps their architecture to skillfully capture the changes in a dynamic graph in the network parameters itself. In order to update the weight matrix of their proposed methodology, they offer two versions of

the EvolveGCN architecture. The first version entitled H- is implemented using gated recurrent units (GRU). The second version entitled the O- version is based on a standard LSTM (Long-Short-Term-Memory) model. In their article, they suggest that the GRU-based weight update version should be used when for the prediction task, node attributes are of importance. In contrast, the O- version of the architecture should be used when the end goal of the application is to make predictions based on the structure of the graph as a whole. They also test their proposed architecture on tasks such as node classification, edge classification, link prediction, etc, and their model was shown to outperform most baseline contemporaries.

Incremental learning is a branch of machine learning that utilizes ever-changing input data to continuously enhance the model's knowledge by taking into account new developments. Kim et al [42] propose a novel incremental learning-based framework which they title DyGRAIN. Their model addresses one of the most crucial challenges of incremental learning with dynamic graphs which is that the receptive fields are also dynamic in nature. Their proposed framework DyGRAIN is capable of suitably identifying the most important changes that are taking place in the graph receptive fields. The two main components of their architecture that help them achieve this are the influence propagation area and the knowledge distillation with the truth component. The influence propagation area identifies the most influenced nodes by new inputs and the knowledge distillation with the truth component identifies all those current nodes that become insensitive to previously learned information by computing the loss discrepancy.

Text and image generation, even video generation in recent times, using static data are problems that have received widespread attention in academia. Furthermore, some of the RNN and Generative Adversarial Network (GAN) based approaches have also found tremendous success in such prediction tasks. Zhang et al's [43] work focuses on a similar but relatively unexplored area of machine learning applications on dynamic graphs which is that of dynamic graph generation. They propose their model D2G2 or Disentangled Deep Generative Model for Interpretable Dynamic Graph Generation. Instead of reinventing the wheel, their model focuses on exiting a pre-existing static graph approach to dynamic graphs by overcoming three salient challenges - (1) lack of decoders for joint edge and node generations, (2) difficulty of characterization of static and dynamic graph components, and (3) difficulty in the joint modeling

of the dynamics of nodes and edges. Their method works with attributed temporal graph generation wherein the generative model is formulated as a Bayesian network tasked with characterizing the independence among the components of the graph. Further, based on the conditional independence that exists between the variables, they provide two separate inference models - factorized inference model and the full inference model. Lastly, based on recurrent and graph deconvolutions, novel node-edge encoders are proposed. As a part of their research, they also undertake performance comparison studies that further solidify the utility of their technique.

Trivedi et al [44] propose DyREP which is a novel modeling framework that is shown to outperform existing baselines for link prediction problems and time prediction problems on dynamic graphs. Their newly proposed framework is a representation modeling-based system that takes in dynamic graphs represented as association or communication events and dynamically updates the node representations in accordance with such events. Since their model focuses on association and communication events, their solution utilizes the concept of a two-time scale temporal point process methodology. This temporal point process model is parameterized with the help of a representation network that is inductive in nature which facilitates the modeling of the node representations. Finally, they design a novel component for their pipeline for mapping a dynamic graph into lower dimensional embeddings called the temporal attention mechanism that helps capture attentive neighborhood information that governs node representations over time.

Zhu et al [45] propose an application-specific novel architecture for Alzheimer's disease diagnosis using graph convolutional networks as one of the components of their architecture. The architecture is composed of three modules - the first is an interpretable feature learning module that selected only the most informative features at any stage and attempts to eliminate redundant features by attempting to capture the correlation of nodes. The second module deals with dynamic graph learning and it deals with the task of generating graph embeddings that capture relevant neighborhood relationships. The last is the graph convolutional network module that outputs the diagnosis for a given input based on learnings it has gained while training on and studying the embedded characteristics of the fed dynamic graph structure.

Zhang et al [46] propose a framework that focuses on leveraging the unique characteristics of dynamic graphs to create a more intuitive and personalized recommendation system. Their model is entitled DGSR or the dynamic graph neural network and is used to obtain sequential recommendations. They argue that most of the currently existing methods in the field of user preference modeling and consequent recommendation-making focus only on the users' interest and their own pattern/sequence of choices. Most of the current methods do not pay heed to collaborative relations that may exist between different users' item choice sequences. To address this challenge, they set out to use dynamic graph structures and model a user's own preference sequence and the dynamic collaborative signals between different user sequences into one cohesive model. Their framework utilizes a subgraph sampling strategy that helps extract only relevant user sequences and this information is encoded using the dynamic graph recommendation network which constitutes an attention module as well as an RNN module to learn both short-term and long-term preferences from user sequences. They design a dynamic graph recommendation network that helps model user interactions as a dynamic graph and thus consequently convert the recommendation task to a link prediction problem between two nodes.

Peng et al [47] propose a reinforcement learning-based graph convolutional network approach tuned specifically to traffic flow problems. They tackle the problem of data integrity in the traffic prediction task by proposing the use of reinforcement learning. In particular, they use a graph convolutional policy network on traffic flow probability graphs which are the baseline inputs for the model, and operate under the assumption that this data is defective. Using the reinforcement learning technique, dynamic graph sequences are generated that suitably capture the characteristics of the given traffic flow graph and utilize complete historic information to produce more complete and accurate graphs. Following this, graph convolution is performed on these results - a process that is intended to capture the spatial information from the dynamic graphs - and an LSTM model is used to capture both the temporal information and the dynamic nature of the problem.

Peng et al [48] also previously proposed another framework called Dynamic GRCNN and focused their application to predict urban traffic passenger flows. Here, they utilize historic passenger flow data and use it to construct dynamic graphs that model relationships among the

traffic transportation hubs. The node attributes, here, capture information regarding the inflow and outflow of passengers. This dynamic graph is then fed into the Dynamioc GRCNN model to output feature embeddings and the final prediction is extracted using a three-layer simple MLP (Multi-Layer Perceptron) model to predict each station's inflow and outflow as a node attribute prediction task.

Goyal et al [49] propose their embedding model dyngraph2vec which uses deep learning architecture made up of dense recurrent layers that is capable of learning the time-based changes that occur in the network. Their goal was to understand how the ever-changing nature of the graph affects the performance of prediction tasks by capturing the underlying evolution dynamic. Their method focuses on learning the structural patterns in each network by virtue of non-linear layers and it uses recurrent layers to understand temporal dependencies.

La Gatta et al [50], in their article, apply the dynamic graph problem to the task of modeling COVID-19 evolution. Traditionally, mathematical models such, as the SIR model, are used to represent the spread of diseases. However, such models fail to incorporate preventive measures taken by actors and also do not capture the introduction of new dynamic states into the model to distinguish those who have and have not taken precautions. To overcome this challenge, the authors have modeled the diffusion of the epidemic as a dynamic graph where the vertex corresponds to places and edges represent the spread and movement of infection. Further, they apply graph convolutional networks and LSTM to capture the spatiotemporal characteristics of these graphs and accurately model real-time pandemic spread trends. Instead of traditional epidemiological models that tend to model the diffusion of disease in one particular area as a whole, their novel technique helps to observe the flow of infections amongst different areas effectively.

Table 2: Summary of the surveyed Machine learning models

Authors	Framework Title	Application Ares	Machine Learning Approach Used	Class of Technique
Dhanjal et al [32]	Incremental Approximate Spectral Clustering (IASC)	Node Clustering on HIV epidemic, citation network and the purchase history graph of an e-commerce website.	Spectral clustering	DYNAMICAL APPROACH
Gorke et al [33]	Cut clustering	Node clustering on network of e-mail communications.	minimum-cut tree clustering	
Aggrawal et al [34]	DYCOS- DY namic Classification algorithm with cOntent and Structure.	Classification on CORA dataset - scientific publications classified into one of seven classes.	classification	
Yao et al [35]	IncSVM - e incremental SVM , WinSVM - window-based incremental SVM	Classification on IMDB Network, DBLP Network (database containing millions of publications in computer science).	SVM classification	
Xu et al [36]	AdaNN - adaptive neural network for transductive node classification	Node classification on Brain dataset (here, nodes represent tidy cubes of brain tissue and edges indicate the connectivity to find similar areas of activity), DBLP-53 (a co-author network dataset where the nodes represent the authors), Reddit posts, Epinions data (extracted from a product review site where users construct trust graphs to seek advice from others).	Encoding for Node classification	
Rossi et al [11, 37]	Temporal Graph Network or TGN	Future link prediction and dynamic node classification on Wikipedia, Reddit posts, and Twitter datasets.	General purpose encoding of dynamic graph - can be coupled with any neural network/ ML model architecture.	GRAPH REPRESENTATION LEARNING
Ma et al [38]	DGNN, a new Dynamic Graph Neural Network model	Link prediction and node classification on an online community of students from the University of California, Irvine, 2016 Democratic National Committee email leak, and product review platform Epinions.	General purpose encoding of dynamic graph - can be coupled with any neural network/ ML model architecture.	
Gunarathna et al [39]	GTA-RL (Graph Temporal Attention with Reinforcement Learning)	Combinatorial problems such as that in the case of a ride sharing scenario	a novel encoder decoder architecture	
Yu et al [40]	NEDGNN or the Novel Neighborhood Ectended Dynamic Graph Neural Network	-	General purpose encoding of dynamic graph - evolution of the entire graph topology	
Pareja et al [41]	Evolving Graph Convolutional Network Design for Dynamic Graphs or EvolveGCN	Node classification, edge classification, and link prediction on Stochastic Block Model, bitcoin, reddit posts, and UC irvine student data.	General purpose encoding of dynamic graph	
Kim et al [42]	DyGRAIN - An Incremental Learning Framework for Dynamic Graphs	Node classifcation task on Open Graph Benchmark (OGB) Arixv dataset, Reddit posts dataset and PubMed dataset.	General purpose encoding of dynamic graph using incremental learning	
Zhang et al [43]	D2G2 or Disentangled Deep Generative Model for Interpretable Dynamic Graph Generation	Graph generation using datasets on protein folding, user authentication, and metro farecard records from the D.C. metro system.	Interpretable dynamic graph generation using VAE and GAN.	
Trivedi et al [44]	DyREP: Learning Reprisentations Over Dynamic Graphs	Dynamic link prediction and time prediction tasks on social evolution dataset and Github dataset.	Graph encoding using representation learning techniques.	
Zhu et al [45]	Interpretable Dynamic Graph Convolutional Networks (IDGCN)	Alzheimer disease diagnosis using graph convolutional networks.	Three module framework that includes feature learning module, dynamic graph learning module, and graph convolutional network.	
Zhang et al [46]	DGSR or the dynamic graph neural network	Recommendation based on Amazon-CDs, Amazon-Games, and AmazonBeauty datasets.	RNN for sequential recommendations	
Peng et al [47]	A traffic flow probability graph model	Traffic flow spatio-temporal prediction model using dynamic graph generation using realtime NYC bike data.	Reinforcement learning, Graph convolution, nad LSTM based dynamic graph generation	
Peng et al [48]	novel dynamic graph recurrent convolutional neural network, namely DynamicGRCNN	Urban traffic passenger flows prediction.	Convolutional neural network based dynamic graph generation at varied granularity levels	
Goyal et al [49]	dyngraph2vec	Link prediction on various real life datasets.	General purpose encoding of dynamic graph	
La Gatta et al [50]	-	Modelling of COVID-19 evolution.	Pandemic diffusion prediction using Graph Convolutional Network and LSTM	

In Table 2 above, a summary of the presented frameworks has been presented -

4.2 Dynamic Graph Embeddings

From the various models presented in the section above, it can be concluded that generally speaking, for most graph representation learning-based approaches, the proposed architecture is of the general form illustrated in figure 5 below.

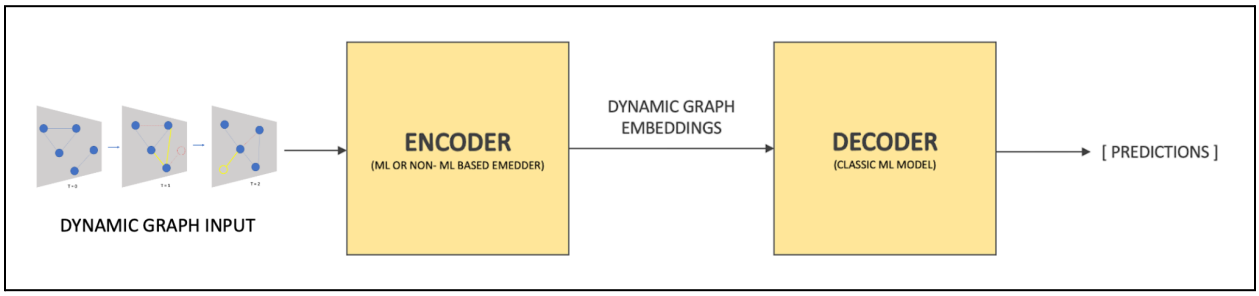

Figure 5: General Architecture of ML Models that Work with Dynamic Graphs

For most of the surveyed models, the encoder is usually the portion of the architecture that produces graph embeddings and the decoder constitutes mainstream predictive algorithms that can be trained to work on the produced embeddings. The nature of the decoder model used and the embeddings produced depend on the application at hand. Furthermore, it can be observed that the main task in most of the dynamic graph prediction models presented above is to devise efficient embeddings for graph information. Thus in conclusion, in most recent works on machine learning model applicability to dynamic graphs, much emphasis is placed on the generation of graph embeddings. Figure 6 below illustrates a general taxonomy of the various dynamic graph embedding techniques that are currently in use. A detailed overview of some of these available embedding techniques and the associated algorithms is followed.

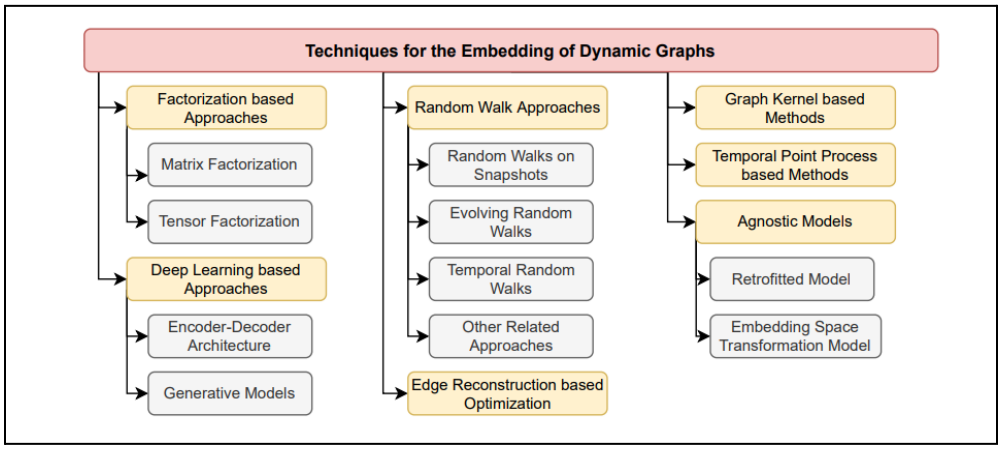

Figure 6: taxonomy of the various dynamic graph embedding techniques [19]

Graphs are non-euclidean structures that contain three levels of information within them - this is information at the node level, at the edge level, and that at the graph level. While graphs are a very good way to visualize and represent real-life scenarios computationally, there have to be suitable mathematical representations to be able to perform computations on them. Traditional representations of graphs like the adjacency matrix might look like a valid solution but as the complexity of real-life situations grows (consider a case where you have 1 million rows for each time instant) the adjacency matrix approach is no longer practical. The most common way to handle this problem is to use a technique called graph embedding. Graph embeddings help to represent an input graph as a vector thereby modifying it to become a more suitable input to machine learning models. This mapping to vectors is done in a manner that captures the salient topological features of the graph.

At the node level, dedicated node embeddings can be produced that result in vectors that capture information about the graph structure. Similarly, in edge embeddings, each edge of a graph is mapped to an edge vector. For example, in a social network, you can have a multi-edge graph where nodes could be connected by edges based on age ranges, gender, friendships, etc. An edge embedding in such a case would capture the aforementioned attributes. In the same graph, node embeddings could capture information about the personal characteristics of the actors involved in the social network. Embeddings on the graph level are not as common and they consist of generating an embedding vector representing each graph. Think of a large graph with multiple subgraphs, each of the corresponding subgraphs would have an embedding vector representing the graph structure. Classification problems are a common application where graph embedding can be useful [19].

One of the major techniques that can be used to produce dynamic graph embeddings is factorization-based methods where the idea is to compute similarity measures amongst the time-stamped snapshots and then map these to a lower dimensional space. The similarity matrices that are computed can be represented in two main ways - as a series of similarity matrices called the matrix factorization approach or as a three-dimensional tensor where the first two dimensions store the metric value and the third represents the time dimension called the tensor factorization approach. Some of the techniques that use matrix-based factorization to

embed dynamic graphs include Zhu et al's [51] temporal latent space model with the gaol to predict the evolution of links based on previous snapshots of latent space. Further, Ferreira et al [52] propose a temporal ideological space model while studying and analyzing the behavior of dynamic political networks. Their method is baes on temporal vertex embeddings and is capable of revealing patterns - both in terms of the dynamically changing opinions characterized by actors and the overall network structure. Both the above approaches involve the joint optimization of the loss function and the concept of temporal smoothing in order to obtain the embeddings. Tensor factorization approach is used in the CP [53] tensor rank decomposition technique and in the TuckER model to produce graph embeddings [54].

Neural network architecture, as has been seen in section 4.1 have shown remarkable performance in producing dynamic graph embeddings in recent years. Architectures that employ recurrent neural networks such as LSTMs and GRUs have been proposed and have the capability to produce embeddings that encapsulate temporal correlation of graph components as well as the more complex underlying correlations amongst graph attributes. Furthermore, based on the specific application at hand, attention components can be added to the neural networks to focus the embeddings to capture a particular kind of correlation well. Recently, convolutional neural networks as iterative propagation procedures have also been employed to produce dynamic graph embeddings [55, 56]. Generative models, such as variational autoencoders (VAEs) and generative adversarial networks (GANs) can be employed to gradually learn these data distributions in dynamic graphs. These networks can learn an approximate representation of the network distribution by capturing the data distribution and using a recognition network (or a discriminator network) to calculate the likelihood that a sample came from the data distribution. This helps them learn low-dimensional latent representations of the training data.

A third technique to obtain graph embeddings is the random walk method which makes it possible to extract higher-order relationships without adjacency matrices using a class of algorithms for graph embedding functions that generate a context for each node. Numerous random walks of a defined length L are used to create the node sequence matrix, which is then factorized using a neural network architecture like the Skip-Gram [57] to create low-dimensional vector representations for each node while preserving their closeness in the new embedded

space. Random walks must produce time-dependent contexts on dynamic graphs. Walks on snapshots, evolving random walks, and temporal random walks are the three methods in this category that can be used to incorporate the temporal aspect. To create contexts, one can potentially apply additional node sequence sampling techniques such as neighborhood aggregation.

Finally, another major category of dynamic graph embedding technique is the temporal point process-based approaches. Here, it is assumed that interactions between nodes are a function of the network topology, node attributes, and historic interactions and that if an event affects a given node then as a consequence all the nodes that are susceptible to its influence shall also be affected. Other embedding techniques include agnostic-based methods, temporal smoothing with optimized edge reconstructions, etc.

5. Conclusions and Future Scope

The application of machine learning algorithms to dynamic graphs is a new and upcoming field in the graph machine learning domain. Much work is being done in this field due to the significant importance of such applications to real-world problems. Through the use of dynamical approaches and graph representation learning, the most recent research is well on its way to tackling the unique problems associated with the usage of such graphs such as the efficient tackling of the temporal component.

In this term paper, the analysis of the usage of machine learning algorithms that perform predictive analysis of dynamic graphs was performed. Initially, a survey of the relevance of dynamic graphs to the big data and analytics industry was conducted and the reason for the failure of traditional machine learning methods or static graph frameworks in the case of their dynamic counterparts was highlighted. Further, a brief overview of dynamic graphs, the different ways in which they can be modeled and visualized, and their importance to numerous critical areas of applications was presented. The currently used and most recent machine learning frameworks being used to make dynamic graph-based predictions were surveyed and it was discovered that these classes of algorithms are broadly of two kinds - dynamical or graph representation learning based. The exploration of the field also helped conclude that a majority

of the algorithms being developed focus their attention on the production of dynamic graph embeddings and hence a detailed overview of various techniques that produce embeddings was also presented.

However, it must be noted that the field of machine learning with dynamic graphs is still relatively new, and significant work needs to be done to tackle the numerous flaws the currently proposed frameworks suffer from. Some of the challenges that need to be addressed in this field include -

- Lack of Sophisticated modeling - The current algorithms used to produce dynamic graph embeddings are limited in their ability to capture the complex spatiotemporal dependencies in dynamic graphs.
- Lack of scalable solutions - Real-world scenarios when modeled using dynamic graphs can generate structures of the order of hundreds of millions of nodes and relevant frameworks to tackle such large-scale dynamic graphs need to be developed.
- Lack of a standardized approach to evaluate performance - while many new machine learning frameworks for dynamic graphs have been proposed recently, there is a lack of research comparing and contrasting the most recent methods. Furthermore, there is no standard approach to evaluating a dynamic graph-based model's performance leading to a wide array of promising but incomparable performance metrics.

In conclusion, it can be stated that dynamic graphs and the application of machine learning algorithms to these complex structures is a rapidly growing field due to the numerous advantages it offers. While working with dynamic graph structures is complicated due to the challenges of handling the complex information they carry, the development of suitable frameworks can lead to more refined and personalized predictions which are of great importance in today's user experience-based industry. Therefore, while much work is required to refine methodologies that can help unlock the potential of dynamic graph applications, it is clear with the present trends that dynamic graphs will play a big role in the field of predictive analysis in the coming time.

References

[1] Alfredo Cuzzocrea and Il-Yeol Song. 2014. Big Graph Analytics: The State of the Art and Future Research Agenda. In Proceedings of the 17th International Workshop on Data Warehousing and OLAP (DOLAP '14). Association for Computing Machinery, New York, NY, USA, 99–101. https://doi.org/10.1145/2666158.2668454

[2] J. A. Miller, L. Ramaswamy, K. J. Kochut and A. Fard, "Research Directions for Big Data Graph Analytics," 2015 IEEE International Congress on Big Data, New York, NY, USA, 2015, pp. 785-794, doi: 10.1109/BigDataCongress.2015.132.

[3] Meticulous Market Research Pvt. Ltd. (2022, January 19). Graph Analytics Market Worth $2.03 Billion by 2027 - Market Size, Share, Forecasts, & Trends Analysis Report with COVID-19 Impact by Meticulous Research®. *GlobeNewswire News Room*. https://www.globenewswire.com/en/news-release/2022/01/19/2369446/0/en/Graph-Analytics-Market-Worth-2-03-Billion-by-2027-Market-Size-Share-Forecasts-Trends-Analysis-Report-with-COVID-19-Impact-by-Meticulous-Research.html

[4] Stankovic, L., Mandic, D. P., Dakovic, M., Brajovic, M., Scalzo, B., Li, S., & Constantinides, A. G. (2020). Data Analytics on Graphs Part III: Machine Learning on Graphs, from Graph Topology to Applications. *Foundations and Trends in Machine Learning*, *13*(4), 332–530. https://doi.org/10.1561/2200000078-3

[5] Yi, H., You, Z., Huang, D., & Kwoh, C. K. (2021). Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, *23*(1). https://doi.org/10.1093/bib/bbab340

[6] *Tracked: How Does Netflix Know What You Like? : Networks Course blog for INFO 2040/CS 2850/Econ 2040/SOC 2090*. (2021, September 15). https://blogs.cornell.edu/info2040/2021/09/15/tracked-how-does-netflix-know-what-you-like/#:~:text=In%20this%20manner%2C%20our%20viewing,made%20in%20the%20recommendations%20graph

[7] Dong, X. L. (2020). Building product graphs automatically - Amazon Science. *Amazon Science*. https://www.amazon.science/blog/building-product-graphs-automatically

[8] Hood, J., Graber, C., & Brase, G. L. (2020). Comparing the Efficacy of Static and Dynamic Graph Types in Communicating Complex Statistical Relationships. *Frontiers in Psychology*, *10*. https://doi.org/10.3389/fpsyg.2019.02986

[9] Stanford University. (2021). *Stanford University CS230 Deep Learning Section 5 (Week 5)*. www.cs230.stanford.edu. https://cs230.stanford.edu/section/5/

[10] Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. M. (2020). Temporal Graph Networks for Deep Learning on Dynamic Graphs. *arXiv (Cornell University)*. https://arxiv.org/pdf/2006.10637

[11] Teja, R. (2023, January 26). What are Dynamic Graphs and Why They are Interesting? *Medium*. https://towardsdatascience.com/what-are-dynamic-graphs-and-why-they-are-interesting-180b9fab9229

[12] Yangzihao Wang, Yuechao Pan, Andrew Davidson, Yuduo Wu, Carl Yang, Leyuan Wang, Muhammad Osama, Chenshan Yuan, Weitang Liu, Andy T. Riffel, and John D. Owens. 2017. Gunrock: GPU Graph Analytics. ACM Trans. Parallel Comput. 4, 1, Article 3 (March 2017), 49 pages. https://doi.org/10.1145/3108140

[13] *What is Graph Analytics?* (n.d.). NVIDIA Data Science Glossary. https://www.nvidia.com/en-us/glossary/data-science/graph-analytics/

[14] Morgan, L. (2021). Why using graph analytics for big data is on the rise. *Business Analytics*. https://www.techtarget.com/searchbusinessanalytics/feature/Why-using-graph-analytics-for-big-data-is-on-the-rise

[15] Anshari, M., Almunawar, M. N., Lim, S. A., & Al-Mudimigh, A. S. (2019). Customer relationship management and big data enabled: Personalization & customization of services. *Applied Computing and Informatics*, *15*(2), 94–101. https://doi.org/10.1016/j.aci.2018.05.004

[16] Qasem, Z., Jansen, M., Hecking, T., & Hoppe, H. U. (2016). Influential Actors Detection Using Attractiveness Model in Social Media Networks. *Studies in Computational Intelligence*. https://doi.org/10.1007/978-3-319-50901-3_10

[17] *Deep learning on dynamic graphs*. (n.d.). https://blog.twitter.com/engineering/en_us/topics/insights/2021/temporal-graph-networks

[18] Camossi, Elena & Bertolotto, Michela & Bertino, Elisa. (2023). Implementation Challenges in Spatio-temporal Multigranularity.

[19] Barros Claudio, D. T., MendonçaMatheus, R. F., VieiraAlex, B., & ZivianiArtur. (2021). A Survey on Embedding Dynamic Graphs. *ACM Computing Surveys*, *55*(1), 1–37. https://doi.org/10.1145/3483595

[20] Beck, F., Burch, M., Diehl, S., & Weiskopf, D. (2017). A Taxonomy and Survey of Dynamic Graph Visualization. *Computer Graphics Forum*, *36*(1), 133–159. https://doi.org/10.1111/cgf.12791

[21] Jain, M., Singh, G., & Kumar, A. (2015). Graph-based video representation for human action recognition: A review. Computer Vision and Image Understanding, 137, 3-19.

[22] Liu, S., & Yuan, J. (2018). Tracklet-Graphs for Unsupervised Multi-Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), 261-277.

[23] Chen, X., & Gupta, A. (2017). Spatial memory for context reasoning in object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4102-4111.

[24] Bhatia, K., Jain, A., & Kar, P. (2018). Graph convolutional networks for text classification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 328-339.

[25] Zhang, Y., & Chen, W. (2018). Dynamic graph convolutional networks for sentiment analysis on social media. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), 1749-1758.

[26] Jin, X., Pan, Y., & Fang, Y. (2016). Traffic prediction in a bike-sharing system: A graph convolutional neural network approach. In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), 2224-2230.

[27] Chen, Y., Liu, L., Wu, T., Tong, X., & Vasilakos, A. V. (2017). Dynamic traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 18(11), 2956-2967.

[28] Xu, C., Li, H., Wang, F., Gao, X., & Huang, J. (2019). Representation learning on temporal dynamic networks for health status prediction. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 2772-2780.

[29]Li, Y., Zhang, D., Li, L., & Chen, Y. (2020). Dynamic graph convolutional networks for disease prediction. Information Sciences, 515, 357-371.

[30] Bian, Y., Gao, X., Huang, J., Zhang, X., & Liu, Y. (2020). Graph neural networks for social recommendation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 5414-5424.

[31] Wang, H., Wang, J., Wang, J., Liu, Z., & Zhang, Y. (2019). Neural graph collaborative filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 165-174.

[32] Dhanjal, C., Gaudel, R., & Clémençon, S. (2014). Efficient eigen-updating for spectral graph clustering. *Neurocomputing*, *131*, 440–452. https://doi.org/10.1016/j.neucom.2013.11.015

[33] Görke, R., Hartmann, T. N., & Wagner, D. (2012). Dynamic Graph Clustering Using Minimum-Cut Trees. *Journal of Graph Algorithms and Applications*, *16*(2), 411–446. https://doi.org/10.7155/jgaa.00269

[34] Aggarwal, C. (2011). *On Node Classification in Dynamic Content-based Networks*. https://www.semanticscholar.org/paper/On-Node-Classification-in-Dynamic-Content-based-Aggarwal-Li/1c7ec5f28d034953edbd1040e46a1e1b8a90aac2

[35] *Scalable SVM-Based Classification in Dynamic Graphs*. (2014, December 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/7023382

[36] Xu, D., Cheng, W., Luo, D., Gu, Y., Liu, X., Ni, J., Zong, B., Chen, H., & Zhang, X. (2019). Adaptive Neural Network for Node Classification in Dynamic Networks. *International Conference on Data Mining*. https://doi.org/10.1109/icdm.2019.00181

[37] *Deep learning on dynamic graphs*. (n.d.-b). https://blog.twitter.com/engineering/en_us/topics/insights/2021/temporal-graph-networks

[38] Ma, Y. (2018, October 24). *Dynamic Graph Neural Networks*. https://www.arxiv-vanity.com/papers/1810.10627/

[39] Gunarathna, U., Borovica-Gajic, R., Karunasekara, S., & Tanin, E. (2022). Solving Dynamic Graph Problems with Multi-Attention Deep Reinforcement Learning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2201.04895

[40] Yu, D., Wang, J., & Jiang, C. (2022). Neighborhood Extended Dynamic Graph Neural Network. *2022 14th International Conference on Machine Learning and Computing (ICMLC)*. https://doi.org/10.1145/3529836.3529851

[41] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B., & Leiserson, C. E. (2020). EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. *Proceedings of the . . . AAAI Conference on Artificial Intelligence*, *34*(04), 5363–5370. https://doi.org/10.1609/aaai.v34i04.5984

[42] Kim, S., Yun, S., & Kang, J. (2022). DyGRAIN: An Incremental Learning Framework for Dynamic Graphs. *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. https://doi.org/10.24963/ijcai.2022/435

[43] Zhang, W., Zhang, L., Pfoser, D., & Zhao, L. (2021). Disentangled Dynamic Graph Deep Generation. *Society for Industrial and Applied Mathematics eBooks*, 738–746. https://doi.org/10.1137/1.9781611976700.83

[44] Trivedi, R., Farajtabar, M., Biswal, P., & Zha, H. (2019). DyRep: Learning Representations over Dynamic Graphs. *International Conference on Learning Representations*. https://openreview.net/pdf?id=HyePrhR5KX

[45] Zhu, Y., Ma, J., Yuan, C., & Zhu, X. (2022). Interpretable learning based Dynamic Graph Convolutional Networks for Alzheimer's Disease analysis. *Information Fusion*, *77*, 53–61. https://doi.org/10.1016/j.inffus.2021.07.013

[46] Zhang, M., Wu, S., Yu, X., & Wang, L. (2022). Dynamic Graph Neural Networks for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 1. https://doi.org/10.1109/tkde.2022.3151618

[47] Peng, H., Du, B., Liu, M., Liu, M., Ji, S., Wang, S., Zhang, X., & He, L. (2021). Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. *Information Sciences*, *578*, 401–416. https://doi.org/10.1016/j.ins.2021.07.007

[48] Peng, H., Wang, H., Du, B., Bhuiyan, Z. A., Ma, H., Liu, J., Wang, L. V., Yang, Z., Du, L., Wang, S., & Yu, P. S. (2020). Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, *521*, 277–290. https://doi.org/10.1016/j.ins.2020.01.043

[49] Goyal, P., Chhetri, S. R., & Canedo, A. (2020). dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge Based Systems*, *187*, 104816. https://doi.org/10.1016/j.knosys.2019.06.024

[50] La Gatta, V., Moscato, V., Postiglione, M., & Sperlì, G. (2021). An Epidemiological Neural Network Exploiting Dynamic Graph Structured Data Applied to the COVID-19 Outbreak. *IEEE Transactions on Big Data*, *7*(1), 45–55. https://doi.org/10.1109/tbdata.2020.3032755

[51] *Scalable Temporal Latent Space Inference for Link Prediction in Dynamic Social Networks*. (2016, October 1). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/7511675

[52] Ferreira, C., Ferreira, F., De Sousa Matos, B., & De Almeida, J. C. (2019). Modeling Dynamic Ideological Behavior in Political Networks. *Web Science*, *7*. https://doi.org/10.34962/jws-80

[53] Phan, A. T., Tichavsky, P., & Cichocki, A. (2013). CANDECOMP/PARAFAC Decomposition of High-Order Tensors Through Tensor Reshaping. *IEEE Transactions on Signal Processing*, *61*(19), 4847–4860. https://doi.org/10.1109/tsp.2013.2269046

[54] Balazevic, I., Allen, C. E., & Hospedales, T. M. (2019). TuckER: Tensor Factorization for Knowledge Graph Completion. *arXiv (Cornell University)*. https://doi.org/10.18653/v1/d19-1522

[55] Jinyin Chen, Xuanheng Xu, Yangyang Wu, and Haibin Zheng. GC-LSTM: Graph Convolution Embedded LSTM for Dynamic Link Prediction. arXiv preprint arXiv:1812.04206, 2018.

[56] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In Advances in Neural Information Processing Systems, pages 1024–1034, 2017.

[57] Sarkar, D. (n.d.). *Implementing Deep Learning Methods and Feature Engineering for Text Data: The Skip-gram Model - KDnuggets*. KDnuggets. https://www.kdnuggets.com/2018/04/implementing-deep-learning-methods-feature-engineering-text-data-skip-gram.html