Graph Neural Network Based Network Traffic Classification

Thesis submitted in partial fulfillment of requirements
For the degree of
Master of Technology in Computer Technology
of
Computer Science and Engineering Department
of
Jadavpur University

by

Manas Kanti Saha

Regn. No. - 154204 of 2020-2021
Exam Roll No. - M6TCT23007

under the supervision of

Dr. Mridul Sankar Barik
Assistant Professor

Department of Computer Science and Engineering
Jadavpur University
Kolkata, West Bengal, India
2023

Certificate from the Supervisor

This is to certify that the work embodied in this thesis entitled **"Graph Neural Network Based Network Traffic Classification"** has been satisfactorily completed by **Manas Kanti Saha** (Registration Number 154204 of 2020 − 2021; Class Roll No. 002010504039; Examination Roll No. $M6TCT23007$). It is a bona-fide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata for partial fulfilment of the requirements for the awarding of the **Master of Technology in Computer Technology** degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2022 − 23.

Dr. Mridul Sankar Barik
Assistant Professor,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Forwarded By:

Prof. Nandini Mukhopadhyay
Head,
Department of Computer Science and Engineering,
Jadavpur University.

Prof. Ardhendu Ghoshal
DEAN,
Faculty of Engineering & Technology,
Jadavpur University.

Department of Computer Science and Engineering
Faculty of Engineering And Technology
Jadavpur University, Kolkata - 700 032

Certificate of Approval

This is to certify that the thesis entitled **Graph Neural Network Based Network Traffic Classification** is a bona-fide record of work carried out by **Manas Kanti Saha** (Registration Number 154204 of 2020−2021; Class Roll No. 002010504039; Examination Roll No. $M6TCT23007$) in partial fulfilment of the requirements for the award of the degree of **Master of Technology in Computer Technology** in the **Department of Computer Science and Engineering, Jadavpur University**, during the period of June 2022 to May 2023. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose of which it has been submitted.

Examiners:

(Signature of The Examiner)

(Signature of The Supervisor)

Department of Computer Science and Engineering
Faculty of Engineering And Technology
Jadavpur University, Kolkata - 700 032

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that the thesis entitled **Graph Neural Network Based Network Traffic Classification** contains literature survey and original research work by the undersigned candidate, as a part of his degree of **Master of Technology in Computer Technology** in the **Department of Computer Science and Engineering, Jadavpur University**. All information have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Manas Kanti Saha

Examination Roll No.: M6TCT23007

Registration No.: 154204 of $2020 - 2021$

Thesis Title: Graph Neural Network Based Network Traffic Classification

Signature of the Candidate:

ACKNOWLEDGEMENT

I am pleased to express my gratitude and regards towards my Project Guide **Dr.Mridul Sankar Barik**, Assistant Professor, Department of Computer Science and Engineering, Jadavpur University, without whose valuable guidance, inspiration and attention towards me, pursuing my project would have been impossible.

Last but not the least, I express my regards towards my friends and family for bearing with me and for being a source of constant motivation during the entire term of the work.

Manas Kanti Saha
MTCT Final Year
Exam Roll No. - M6TCT23007
Regn. No. - 154204 of 2020 – 2021
Department of Computer Science and Engineering,
Jadavpur University.

Contents

List of Figures

List of Tables

Abstract

This thesis work surveys the overall concept of Graph Neural Network(GNN). Initially we start with a brief introduction of GNN and what actually GNN is. Then we focused on general design pipeline of GNN by describing all the four steps along with a pictorial representation of the same. Then we give some information about the types of GNN following the graph neural network model. After that, we studied graph embeddings, it's advantages as well as applications. Then we go for working of GNN i.e how actually GNN works. When one try to analyze a graph; it is not that much easy and we try to focus on the challenges to be faced. In the later section, we have discussed about Graph Convolutional Network along with it's type and how it is different from GNN. The learning/training process of neural network is broadly divided into three types i.e Supervised, Memory-based and Unsupervised learning. Two machine learning algorithm- KNN algorithm and Back-Propagation algorithm is descried. Lastly we have discussed about some applications of GNN from theoretical point of view and after that we also studied some real life applications of GNN in today's world. Next, we give our attention to network traffic classification which is no doubt a very important topic in today's scenario where cyber crime is increasing rapidly day by day. In the early days of internet, some traditional traffic classification method is used. But, all of them have some drawbacks and to overcome those researchers have introduced machine-learning in traffic classification. However, in this work we are mainly focused on how machine learning can help to classify a particular network traffic flow. In this thesis work, we have tried to build a GNN model that can be used to classify network traffic. This technique relies on the extraction of topological data in the network traffic. Experiences from the experiments have shown GNN based network traffic classifier as a promising and effective solution. Further a thorough assessment of the proposed technique using actual datasets would validate its acceptability compared to state-of-the-art techniques.

Chapter 1

Introduction

1.1 Preamble

The number of crimes related to internet i.e Cyber Crime is getting increased day by day due to the extensive use of IoT devices. However, few years back the scenario was not the same and at that moment necessity of internet traffic classification is not as much important as today. In today's scenario, real time traffic analysis has become essential to detect any suspicious activity over the network. Furthermore, network operator can have an eye to the growth of different applications and provide network accordingly to accommodate the various needs of the customers.Network traffic classification is the first and most important step of network analysis and this is main building block of Intrusion Detection Systems (IDS). From the security point of view, traffic classification can reduce the harm caused by attackers or avoid it. Moreover, accurate classification of various network traffic can help to identify the applications that are using network resources and facilitates the quality of Services(QoS) for different applications. There are some network traffic classification methods exist like pay-load based traffic classification which is the earliest classification method. But this method is not at all reliable due to inaccuracy in results.To overcome the challenges of Port Based Classification researchers have introduced Pay-Load based classification which sometimes called as Deep Packet Inspection (DPI).It faces difficulties while coping up with large data flows and high speed network traffic. This is also computationally costly. Then researchers strive for algorithms which are light-weight and less computationally costly and can also works for encrypted network data.To suffice this need; machine learning based network traffic classification came into the scenario.To perform ML based traffic classifications we have to follow few steps- Data Collection, Flow Representation, Feature Engineering, Data Set Preparation, Model Building and Model Evaluation.Here, SVM(Support Vector Machine) is considered to be most effective ML based traffic classification due to it's promising performance. Further, How SVM can classifies network traffic with neural network is discussed followed by it's advantage as well as disadvantages. In this thesis work, we have tried to build a GNN model that can be used to classify network traffic.Further a thorough assessment of the proposed technique using actual datasets would validate its acceptability compared to state-of-the-art techniques. we leave it as a future work.

1.2 Research Statement

To show the feasibility of using graph neural network models for network traffic classification and to assess its effectiveness.

1.3 Contribution of the Thesis

In this thesis work we have tried to build a GNN model that can be used to classify network traffic generated by different applications. This technique relies on the extraction of topological data in the network traffic. Experiences from the experiments have shown GNN based network traffic classifier as a promising and effective solution for traffic classification.

1.4 Outline of the Thesis

This thesis work is comprises of 6 chapters at all. The organization of this thesis is depicted as below-

1. In Chapter 2, we have first discussed about Graph Neural Network. Then we go for general design pipelines of it which contains four steps. A pictorial representation of that pipeline architecture is shown. After that we focus about graph embedding which is nothing but an approach that is used to transform nodes, edges and their features into vector space. We have broadly classified it into two types. Furthermore, advantages of using graph embeddings and it's applications are depicted. In the later section, we have discussed about the basic working principle of GNN. After that we have briefly discussed about Graph Convolution Network followed by how it is different from GNN. After that, three types of learning rule i.e Supervised Learning, Memory Based Learning Rule and Unsupervised Learning is discussed. Then we focuses on working principle of KNN algorithm. We conclude this chapter by describing some real life applications of GNN in today's world.

2. In Chapter 3, we have given our attention to network traffic classifications. We start our discussion from the ground level like what is network traffic and why it is necessary to be classified. Then various traffic classification methods like - Port Based Classification, Pay-Load classification, Statistical Classification etc and most importantly Machine Learning based classifications are discussed along with some parameters which works as a catalyst to introduce machine learning in network traffic classification domain. Next, traffic classification metrics are explained through which we can measure the accuracy of classification. Then we introduce one popular and most commonly used ML algorithm namely SVM (Support Vector Machine) algorithm in our chapter with it's advantages,disadvantages, types and some kernel functions. We have concluded this chapter by detail studying of K-nearest neighbours method for classifying user's session in E-commerce website.

3. In Chapter 4, We have discussed about tools like IGNNITION, PyTorch Geometric and some openly available data sets like ASNM Data Sets, IP Network Traffic Flows Labelled with 75 Apps and lastly Computer Network Traffic Data Set. The chapter is concluded with a brief summary.

4. In Chapter 5, Considering the relationship between network traffic, deep learning methods require extracting many features, which is a complicated and time-consuming process. We have extracted information like, graph topology structure and node features expressed as vectors. We have used this data to build a graph neural network model and use it for network traffic classification. Besides that, we have described the workflow of the proposed method in brief.

5. In Chapter 6, We have have discussed about conclusion and future aspect of our work. In this thesis work, we have tried to build a GNN model that can be used to classify network traffic. This technique relies on the extraction of topological data in the network traffic. Further a thorough assessment of the proposed technique using actual data sets would validate its acceptability compared to state-of-the-art techniques. we leave it as a future work.

Chapter 2

Graph Neural Networks

2.1 Introduction

GNN Stands for Graph Neural Network. They are first introduced back in 2005 but they started to gain popularity in the last five years. The first motivation of introducing GNN lies in the long-standing history of Neural Networks for graph. Now-a-days it becomes a very powerful tool for many important problems that can be modeled by using graphs.They are able to model the relationship between the nodes in a graph and can produce a numeric representation of it.[8] Recently, researches on analyzing graphs with Machine Learning receives more and more attention because of the very dominant expressive power of graphs. As an unique non-Euclidean data structure for machine learning, Graph analysis focuses on various tasks like Link Prediction, Clustering, Node Classification etc.[19] It is very favourable to use GNN for graph data analysis due to it's convincing performance.The importance of GNN is quite high since there are so many real-world data that can be represented using graphs [8] and GNN can be used there. Like, social network, transportation systems, maps etc. In nineties, Recursive Neural Networks were first utilized on directed acyclic graphs. Afterwards Recurrent Neural Networks and Feed Forward Neural Networks were used.

2.2 What is GNN

As how it is called is a neural network, that can directly be applied to graphs.[7] It provides a convenient way for Node Level, Edge Level and Graph level prediction task.Basically, GNN is a Machine-Learning based algorithm that can extract important information from graphs and make useful predictions. In other words, we can say that it is a Deep-learning based methods that operates on Graph-domain.[10] It has an important advantage over traditional deep-learning methods that they are able to capture graph-structure of data which is often very rich. Actually, GNN originates from two machine learning techniques namely, Recursive Neural Network and Markov Chains where the main idea is to encode data using graphs and subsequently finding the relationship between nodes. A GNN architecture's main idea is to learn embeddings that contains it's neighbourhood's information. However, GNN can be computationally costly mainly for large graphs. The three main types of graph neural networks are- 1) Recurrent Graph Neural Network, 2) Spatial Convolutional Network and lastly 3) Spectral Convolutional Network.[11]

2.3 General Design Pipeline of GNN

In this section we are going to represent the general design pipeline of a GNN model. Generally, the pipeline contains four steps: 1) Find Graph Structure 2) Specify graph type and scale 3) Design loss function and 4)Build model using computational models.[44] The above four stages are described as below:

1. **Find Graph Structure** At first we have to find the graph structure in the application. There are generally two scenarios: Structural scenarios and Non-structural scenarios. In structural scenarios the graph structure is explicit in the application such as applications on molecules, physical systems, knowledge graphs and so on. In Non-structural scenarios, the graphs are implicit.[44] So first we have to build the graph from the tasks such as building a fully-connected "Word" graph for text or building a scene graph for an image.[44] After we get the graph later the building process attempts to find an optimal GNN model for this specific Graph.

2. **Specify Graph type and scale:** As soon as we get the graph in the application we have to find the Graph type and it's scale. Graph with complex type can provide more information on nodes and their connections.[44] Graphs are usually categorized as:

   - **Directed/Undirected Graph:** Edges in directed graphs are all directed from one node to another which provides more information than undirected graphs. Each edge in undirected graphs can also be treated as two directed edges.
   - **Homogeneous/ Heterogeneous Graph:** Nodes and Edges in Homogeneous graph are of same type where Nodes and Edges in the Heterogeneous graphs are of different types. Types of Nodes and Edges play an important role in the heterogeneous graphs and should be considered further.
   - **Static/Dynamic Graph:** When input feature of the graph change with time then it is known as Dynamic Graph. The time information should be considered carefully in this type of graph.[44] On the other hand,Static Graph Consists of a fixed sequence of nodes and edges.

3. **Design Loss Function:** In this step we will design the loss function based on our task type and training setting. For Graph Learning there are usually three kinds of tasks-

   - **Node-Level** tasks focuses on nodes which includes Node Classification, Node Regression, Node Classification etc. Node classification tries to categories nodes into several classes and node regression predicts a continuous value for each node. Node classification aims to partition the nodes into several disjoint groups where similar nodes are in the same group.
   - **Edge-Level** tasks are edge classification and link prediction, which require the model to classify edge types or predict whether there is an edge existing between two given nodes.[44]
   - **Graph-Level** tasks include Graph Regression, Graph Classification, Graph Matching, all of which need the model to learn graph representation.[44]

   From the supervision point of view; the graph learning tasks can be divided into three different training settings namely Supervised setting, Semi-supervised setting and Unsupervised

setting. With respect to the task and based on training setting we can design a specific loss function.

4. **Building Model using computational modules:** At the last step we can build models using computational modules. Some most commonly used modules are described below-

   - **Propagation Module:** The Propagation Module used to propagate information between nodes so that aggregated information could capture both feature and topological information.[44] In propagation module the convolution operator and recurrent operator both are used to aggregate information from the neighbour while skip connection operation is used to gather information from nodes.
   - **Sampling Module:** When graphs are large then Sampling Module is needed to conduct propagation on graphs. This module is generally combined with Propagation Module.
   - **Pooling Module:** When we need high-level sub-graphs or graphs then we need this module to extract information from different nodes.

   With the above computational models a typical GNN model is usually built by combining them. A typical architecture of GNN model is illustrated in the middle of the above figure where the convolutional operator, Recurrent Operator, Sampling Module and Skip Connection methods are used to propagate information in each layer and then pooling module are added to extract high-level information. These layers are basically stacked to obtain better representation.[44]

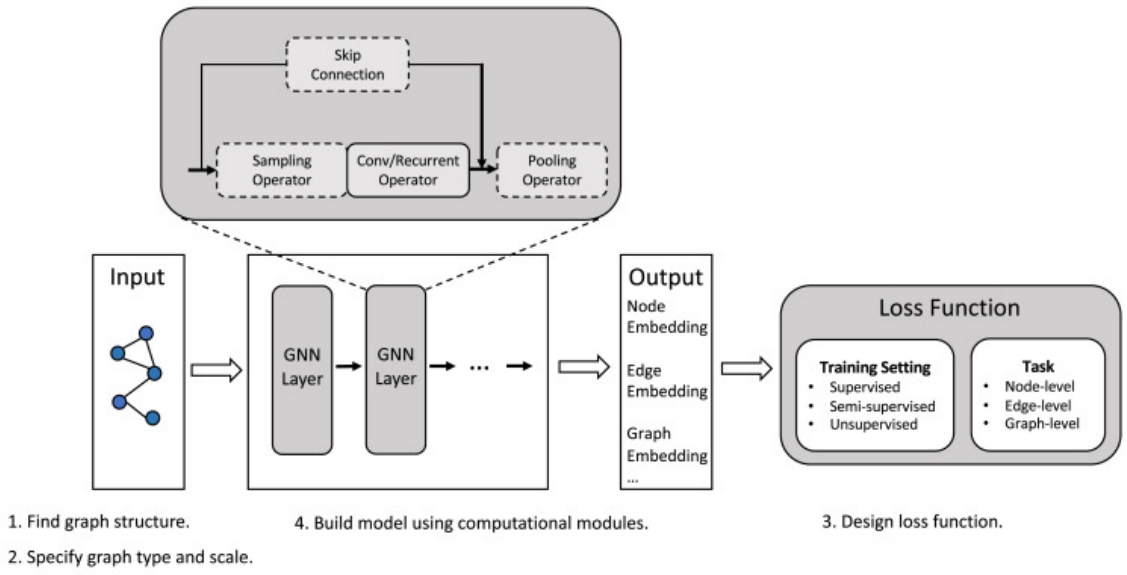

Figure 2.1: General Design Pipeline for GNN Model

In Figure 2.1, we have shown General design pipeline of GNN model.

2.4 Types of GNN

There are several types of graph neural networks exist. In this section we have categorized them in few broad categories-

- **Graph Convolutional Network:** They are very much similar to traditional CNNs. It learns features by inspecting neighbourhood nodes.[5] It sums up node vectors, pass the result to the dense layer. In brief it consists of graph convolution, linear layer and non-linear activation function. There are basically two types of convolution network exists- 1) Spatial Convolutional Network and 2) Spectral Convolutional Network.
- **Graph Auto-encoder Network:** This type of network try to learn graph representation by using an encoder and attempt to reconstruct input graphs by using a decoder.[5] Both the encoder and decoders are joined by a bottleneck layer. They are actually used in link prediction.
- **Recurrent Graph Neural Network:** This type of neural network learn the best diffusion pattern and they can handle multi-relational graphs where a single node may has multiple relations.[5] These type of networks uses less computation power to produce better results. They are used in generating texts and image descriptions, speech recognition etc.
- **Gated Graph Neural Network:** They are better than RGNNs in performing tasks with long-term dependencies. Gated Graph Neural Network improves Recurrent Graph Neural networks by adding a node, edge. Similar to Gated Recurrent Unit (GRU), the gates are used to remember and forget information in different states.[5]

2.5 The Graph Neural Network Model

A graph G is a pair (N,E), where N= set of the nodes and G is the set of edges. Nodes and Edges may have labels that are represented by real vectors. The labels attached to node n and edge (n_1, n_2) will be represented by $l_n \in IR^{l_n}$ and $l_{(n1,n2)} \in IR^{l_E}$ respectively.[38] Let, l denotes the vector obtained by stacking together all the labels of the graph. The notation adopted for labels follow a more general scheme: If y is a vector that contains data from a graph and S is a subset of the nodes then y_s denotes the vector obtained by selecting from y the components related to the node in S. As an example, $l_{ne[n]}$ stands for the vector containing the labels of all the neighbours of n.[38] Labels usually contains features of objects related to nodes. However, when different kinds of edges coexist in the same data set, it is required to distinguish them. This can be easily achieved by attaching a proper label to each edge.
The domain considered here is the set D of pairs of a graph and a node that means D=G*N in which G is the set of graphs and N is the subset of their nodes. We assume a supervised learning framework with the learning set-
L= $(G_i, n_{i,j}, t_{i,j})|$, G_i= (N_i, E_i)$\in$G;
$n_{i,j} \in N_i$; $t_{i,j} \in IR^m$, $1 \leq i, \leq p, 1 \leq j \leq q_i$
where $n_{i,j} \in N_i$ depicts the jth node in the set $N_i \in N$ and $t_{i,j}$ is the desired target associated with $n_{i,j}$.[38]Finally, $p \leq |g|$ and $q_i \leq |N_i|$. Interestingly all the graphs of the learning set can be combined into a unique disconnected graph and learning set can be think of as the pair $L = (G, T)$ where G= (N,E) is a graph and T is a set of pairs $(n_i, t_i)|n_i \in N, t_i \in IR^m, 1 \leq i \leq q$.[38] This compact definition is useful for it's simplicity. The idea is that nodes in a graph represent objects and

edges represent their relationships. Each concept is naturally defined by it's features and related concepts.[38] Thus we can attach a state $x_n \in IR^S$ to each node n that is based on the information contained in the neighbourhood of n. The state x_n contains a representation of the concept denoted by n and can be used to give output o_n.
Let, f_w is a local transition function, which expresses the dependencies of a node n on it's neighbourhood[38] and let, g_w be the local output function that actually described how the output is produced and now x_n and o_n are described as follow-
$x_n = f_w(l_n, l_{co[n]}, x_{ne[n]}, l_{ne[n]})$
$o_n = g_w(x_n.l_n)$
where $l_n, l_{co[n]}$, $x_{ne[n]}$ and $l_{ne[n]}$ are the label of n, the labels of it's edges, the states and the label of the nodes in the neighbourhood of n.[38] In general, the transition and the output functions and their parameters may depend on the node n. In fact, different mechanisms are used to present different kinds of object.

2.6 What is Graph Embeddings

When the Graph data is passed to GNN the features of each nodes are combined with those of it's neighbouring nodes. Finally, the output layer of the GNN produces the embedding, which is a vector representation of the node's data.[8] Actually, it is an approach that is used to transform nodes, edges and their features into vector space. A graph embedding determines a fixed length vector representation for each entity. Basically, Embeddings are a lower dimensional presentation of the graph and preserve the graph's topology.[8] However, we can divide embedding into two types-

- **Vertex Embedding:** This type of embedding is used when we want to perform visualization or prediction on the vertex level. Eg. Visualization of the vertices in the 2D Plane or prediction of new connections based on vertex similarities.[8]

- **Graph Embedding:** Here we represent the whole graph with a single vector.[8] This type of embedding are used when we want to make predictions on the graph level and when we want to compare or visualize the whole graph e.g Comparison of chemical structures.

2.7 Advantages

Till now we came to know about graph embeddings and it's two different types. Now in this section we will discuss a few advantages of graph embedding. They are pointed as below-

- **Choice of Property:** Developing a good graph embedding lead to good representation of nodes.[2] It also stores the relationships between the nodes of a graph.

- **Optimal dimensionality:** The dimensionality of the embedding can be according to the application. Using graph embedding it is possible to find optimal dimensions of the representation of the graph.[2]

- **Scalability:** Today we can see that network are complex and can have large number of nodes and edges. Apply embedding on them will produce a scalable property [2] by which we can process large graphs in a more convenient way.

2.8 Applications

In general we use graph embeddings for compressing the large graph into smaller one by using the information stored in the edges and vertices of the concerned graph. However, here we have listed some applications of it as follows-

- It can be applied to recommendation system. The theory behind is by extracting the vertex embedding we can find similar entities (Persons) and recommend them to the same products of their interest.[2]
- It is most commonly used in NLP to increase the robustness of the models.
- Since it has characteristics of being compact we can use it for dimensionality reduction problems.

2.9 How Actually GNN Works

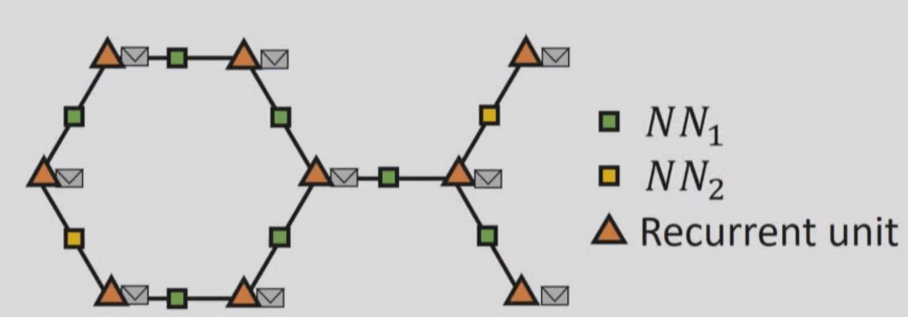

Figure 2.2: Graph Neural Network

As shown in Fig.2.2, Each orange triangle used to be graph node and now it is replaced by a recurrent unit.[8] The envelopes represent the embedding of the nodes that will travel through the entire graph. Each graph edge is also represented by a Neural Network to capture the information of the edge. At a single time step, each node pulls the embedding from all it's neighbours, calculate their sum and and passes them along with it's own embedding to the recurrent unit,[8] which will produce a new embedding. This new embedding contains the information of the node plus the information of all the neighbour's. In the next step it will also contains the information of it's second order neighbours and so on...The process continues until each node knows about all other nodes in the graph. Each of one embedding now has information from all other nodes. The final step is to collect all the embeddings and add them,[8] which will give us single embedding for the whole graph. Although Graph Neural Networks can be understood by similarities with traditional neural networks, a helpful conceptual model unique to GNN is to think of it as message passing between nodes.[6]

2.10 Challenges in Analyzing a Graph

One of the major challenges in analyzing a graph is the non-euclidean nature of the data.[4] The size of the graph is always dynamic in nature.The number of nodes can range from tens or hundreds to the order of millions; similarly, each node can have a variable number of edges.[4] Due to this characteristics, it is quite difficult to represent and analyze graphs by using existing standard methods. It also doesn't help that existing machine learning algorithms have a core assumption that instances are independent of each other. This is false for graph data, because each node is related to others by links of various types. However, Standard convolution that is applied on images cannot be applied here. There have been several attempts to modify convolutions to suit the graph data structure.[4] Therefore, representing graphs by an adjacency matrix is inefficient as it can create very sparse matrices. Also, there can be multiple adjacency matrices to represent the same graph. In[30], the researchers have shown difficult areas and categorized them into four categories- (1) confusing the slope and the height, (2) confusing an interval and a point, (3) considering a graph as a picture or a map and (4) conceiving a graph as constructed of discrete points.

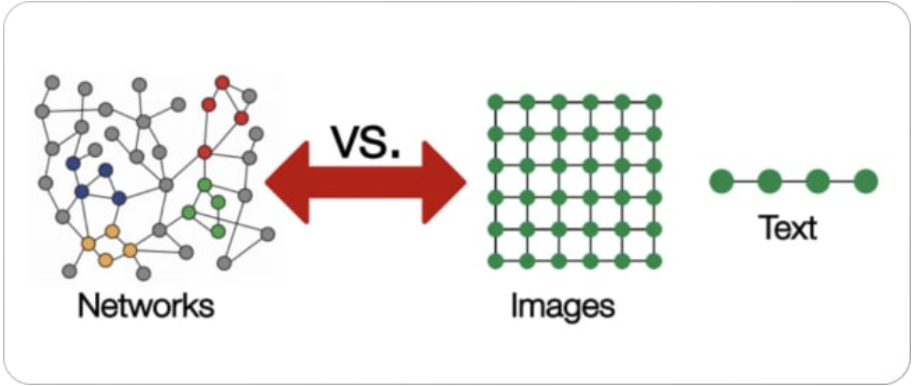

Figure 2.3: Representation of Network vs Images and Text

2.11 Graph Convolutional Network

Graph naturally has wide range of applications due to it's exceptional power of representing complex relationships. However, sometimes it is very challenging to solve learning problems with graphs because of some reasons- 1) many types of data are not originally structured as graphs such as text, images 2) for graph-structured data; the underlying connectivity patterns are complex.[43] Deep learning models on graphs have been introduced in machine learning along with other connected areas and additionally it shows promising results in various problems. In this section, we conduct a comprehensive review mainly on the field of the graph convolutional network which is a very important deep learning models. Despite successes of graph embeddings methods, many of them suffer from the limitations of shallow learning mechanisms.[43] In this context, deep learning models have shown their ability to overcome this in many cases. For examples, convolutional neural network have shown good performance in computer vision and natural language processing applications. The reason behind this success is CNN models can highly exploit the stationary and compositionality properties of that type of data.[43] Due to grid-like nature of images, the convolutional layers of

CNN enables to take the advantages of the hierarchical patterns and extract high level features of image to produce a great expressive capability. A basic CNN model aims to learn a set of fixed-size trainable localized filters which scan each and every pixels in the images and combine the surrounding pixels[43]. We can divide graph convolution neural networks into spectral-based methods and the spatial-based methods. As the earliest convolutional networks for graph data, spectral-based models have achieved impressive results in many graph related analytics tasks. However, spectral-based models are limited to work only on undirected graphs. So the only way to apply spectral-based models to directed graphs is to relax directed graphs to undirected ones, which would be unable to represent the actual structure of directed graphs[29]. Some researchers have combined the recurrent model and spectral-based GCN to process the temporal directed graphs.[34]

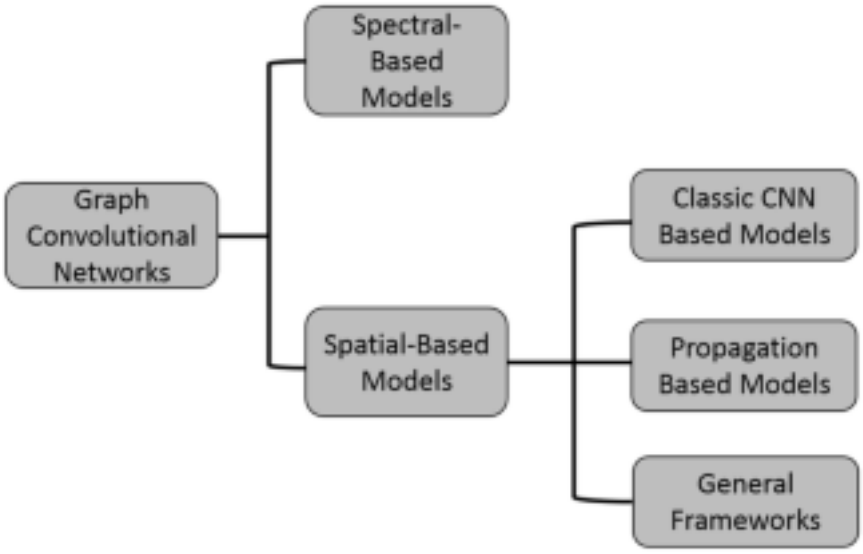

Figure 2.4: Types of Convolution

In the above Fig.2.4,Types of Convolution are presented as a diagram.

2.12 Types of GCN

There are basically two types of GCN available. They are described in below-

- **Spatial Graph Convolutional Networks:** This type of network use spatial features to learn from graphs that are located in spatial space.

- **Spectral Graph Convolutional Networks:** This type of network use Eigen-decomposition of graph Laplacian matrix for information propagation along nodes.[5] These networks were inspired by wave propagation in signals and systems.

2.13 Difference between GNN and GCN

GNN and CNN are two different types of neural network. CNN are specially designed for operating on structured data where GNN are the generalised version of CNN in which the number of nodes can vary and they are unordered.[17] It means that CNN can be applied to structured data like image or text but not on the unstructured data like sound, weather etc. In contrast, GNN can be applied both on structured and unstructured data.

2.14 Learning/Training Process

This is a process by which free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded by means of using labeled or unlabeled patterns. Learning process varies based on the manner in which free parameters are updated. In other words, learning/training process of a neural network is a iterative process in which the calculations are carried out in both the direction i.e forward and backward through each layer in the network until the loss function is minimized. There are basically three types of learning rule-

1. **Error-correcting Learning Rule (Supervised Learning):** Supervised machine learning involves an input variable x and output variable y. The algorithm learns from a training data set. With each correct answers, algorithms iteratively make predictions on the data.The learning stops when the algorithm reaches to an acceptable level of performance.
   An output layer neuron K is driven by a signal vector $x(n)$= x_i(n) produced by one or more hidden layer neurons, which are themselves driven by a input vector to the network. Here, n is the time step of the iterative process. $y_k(n)$: Output signal of neuron k, $d_k(n)$: Desired response or target output and Error signal $e_k(n)$= $d_k(n) - y_k(n)$. Here, $e_k(n)$ acts as control mechanism to apply a sequence of corrective adjustments to the synaptic weights, w of neuron k. Aim is to make $y_k(n)$ come closer to $d_k(n)$ in step-by-step manner so that error will be less and reduce the cost function, $\zeta = 1/2{e_n}^2(n)$. Then adjust weight vector w of neuron K iteratively until the system reaches a steady state. Let, $w_{kj}(n)$: Synaptic weight of the link connected with the k^{th} neuron and the j^{th} element $x_j(n)$ of the signal vector x(n) at the time step n. Adjustment applied to $w_{kj}(n)$ is defined as-
   $\triangle w_{kj}(n) = \eta.e_k(n).x_j(n)$
   $w_{kj}(n+1) = w_{kj}(n) + \triangle w_{kj}(n)$

2. **Memory Based Learning Rule:** The theory behind memory based learning is that concepts which can be classified by their similarity with previously seen concepts. It is done by following two steps- 1) Criterion is used to define "Local Neighbourhood" of x_{test}. and 2) Learning rule applied to the training data in the "Local Neighbourhood" of x_{test}
   There are two types of memory based learning as described below-

   a) Nearest Neighbour Rule:. This is actually the simplest form of memory-based learning

rule. Local neighbourhood is defined as the single training data that lies in the immediate neighbourhood of the test vector x_{test}. To make the rules work in more efficient ways:

- Training/Classified examples (x_i, d_i) are independently and identically distributed, according to the joint probability distribution of the example (x,d).
- The sample size N is infinitely large.

b) K-nearest neighbour classifier: This is a type of supervised machine learning algorithm which can be used for classification problems. Identify the k-classified examples/patterns that lie nearest to the test vector x_{test} for some integer k. Assign x_{test} to the class that is most frequently represented in the k-nearest neighbour to x_{test} using majority voting.[12] i.e class (x_{test})=max class $(x_1, x_2, .., x_k)$. The working principle of this algorithm can be described by using the following steps and in later section will be discussed in details.

- **Step 1:** To implement any algorithm, first thing needed is data sets. So, during the first step of implementing this algorithm we have to load training data and test data.[12]
- **Step 2:** In the next step we need to assign the value of K i.e the nearest data points where K is an integer.
- **Step 3:** For each point in the test data perform the following steps-
  A) Compute the distance between the test data and each row of training data by using Euclidean distance, Hamming distance method.[12] However, Euclidean distance method is very much popular and used more frequently.
  B) Now according to the distance, sort them in ascending order.[12]
  C) Next, it will choose the top K rows from the sorted array.
  D) Now a class will be assigned to the test point on basis of most frequent class of these rows.
- **Step 4:** End

3. **Competitive Learning Rule (Unsupervised Learning):** As the name suggests in this scenario model is not trained using training data sets. Instead of that model itself find the hidden patterns from the given data. It can be compared to the learning that takes place in human brain when learning new things. This type of machine learning rule has input data x and no corresponding output variables. The goal is to model the underlying structure of the data for understanding more about data. There are basically two types of training mode-

   - **Sequential Mode:** Update the weight vectors between input-hidden and hidden-output layers after presenting every example in the epoch. This property provides a smooth, progressive learning journey where every small step allows the learner to be successful in a continuous way.
   - **Batch Mode:** This selects all items at once. In batch mode learning data is stored over a period of time. The machine-learning model is then trained with this stored data from time to time in batches. Additionally, a term batch size is used which refers to number of training examples that propagates through a neural network in one backward/forward pass.

2.15 Working of KNN algorithm

In this section by using an example we will try to understand the working of KNN algorithm. Let's consider we will be dealing with a data set that can be plotted as shown below in Fig. 2.5

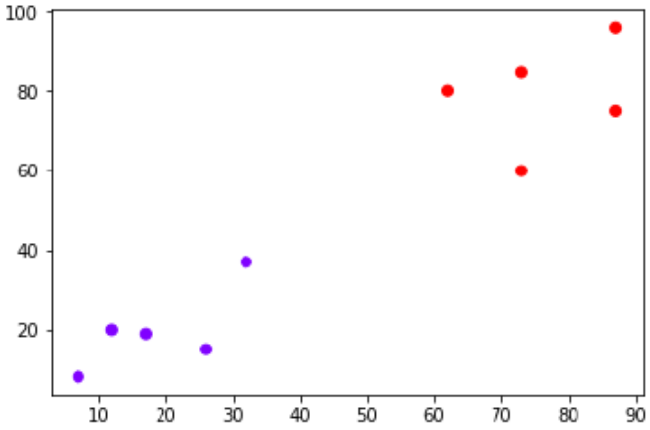

Figure 2.5: Data Set

Now we introduce a new data point with black dot(At point 60,60) in blue or red class [12] and need to classify it. Here, we assume K=3 that means it would find three nearest data points which is shown in the Fig. 2.6 below-

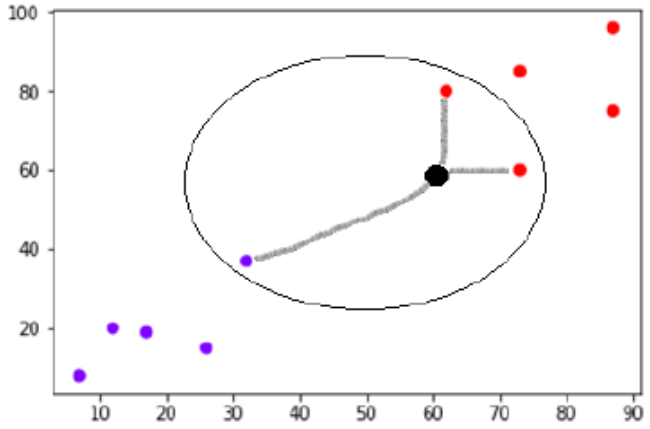

Figure 2.6: KNN Algorithm

According to the algorithm, the three nearest neighbours of the data point with black dot is represented in the above diagram. [12] Among those three, two of them lies in the red class. Hence, the black dot will be assigned in the red class too.

2.16 Back-Propagation Algorithm

Back-Propagation in neural networks is the short term for "Backward propagation of errors".[3] It is the standard method for training of an artificial neural network. This method helps calculate the gradient of a loss function with respect to all the weights in the network. There are few steps to perform the algorithm-

- **Step 1:** Inputs, X arrived via preconnected path.[3]
- **Step 2:** The input is modeled using weight vectors W. However, weights are chosen on random basis.
- **Step 3:** Compute the output of each neuron from the input layer to the hidden layer to the output layer.[3]
- **Step 4:** Calculate errors in the output by using this formula-
  Back propagation error= Actual Output-Desired Output.[3]
- **Step 5:** From the output layer reverse back to the hidden layer to adjust the weights to minimize the error.
- **Step 6:** Repeat the process until desired output is arrived.

2.17 Why Back-Propagation is needed

Most prominent advantages of Back-Propagation are-

- Back-Propagation is fast, simple and easy to program.
- It has no parameters to tune apart from the number of inputs.[3]
- It is a flexible method as it does not require prior knowledge about the network.[3]
- It is a standard method that generally works well.

2.18 Types of Back-Propagation Network

There are two types of Back-Propagation Network as below-

- **Static Back-Propagation:** It is one kind of back-propagation network which produces a mapping of a static output for a static input. It is useful to solve static classification issues like optical character recognition.[3]
- **Recurrent Back-Propagation:** Recurrent Back-propagation is fed-forward until a fixed value is achieved.[3] After that error is calculated and propagated backward.

2.19 Disadvantages of Back-Propagation

1. The actual performance of Back-Propagation on a specific problem depends on input data.
2. It is quite sensitive to noisy data.[3]
3. Need to use matrix-based approach instead of mini-batch.

2.20 Benefits of Neural Network

A neural network derives it's computing power through 1) It's massively distributed structure and 2) it's ability to learn and therefore generalize. Generalization refers to the neural network producing reasonable outputs not encountered during training (Learning).

- Non-Linearity: Neural Network may be linear or non-linear (Neurons are distributed throughout the network). Useful for non-linear input signals.
- Input-Output Mapping : Neural Network can be learned from the labeled samples examples by constructing an input-output mapping.
- Adaptivity: Neural Network has an in-built capability to adapt its synaptic weights to changes in the surrounding environment.
- A neural network can implement tasks that a linear program cannot.[16]
- They have the ability of parallel processing i.e these neural networks have numerical strength that makes them capable of performing more than one task at a single time[1] without affecting the system performance.
- The most important benefit of neural network corresponds high quality and accuracy of the results which is most desirable parameters for any user.
- It can be executed in any applications.[16]
- The input data is stored in it's own network instead of database. That's why any loss of data doesn't affect it's overall working and eventually the performance.[1]

2.21 Sampling Graphs and Batching in GNNs

Batch size in neural networks referred to as the number of training samples used in one complete iteration. The common practice for training neural networks is to update network parameters with the gradient calculated on batch size. The main idea of batching in GNN is to create sub-graphs which will contains the important property of the larger graph in them. However, this practice throws a challenge for the graphs because of variable number of nodes and edges adjacent to each other that means batch size is not constant at all. Graph Sampling operation is highly dependent on context and involves selecting nodes and edges from a graph. These operations might make sense in some context like citation networks and in others, these might be too strong like molecules where a sub-graph simply represents a new, smaller molecule. Each neighbourhood can be considered an individual graph and a GNN can be trained on batches of these sub-graphs.

2.22 Applications of GNN

Now-a-days GNN has a wide range of application in various fields. Here some of them are depicted as below-

- **Node Classification:** Node classification is a common machine learning tasks applied to graph. In node classification the task is to predict the node embedding for each node in a graph.[7] This type of problem is usually trained in a semi-supervised way, where only part of the graph is labeled. There are two major classes of classification problems- binary class and multi-class.[7] In binary-class classifications, the given data set is categorized into two classes and in case of multi-class classifications, the given data set is categorized in several classes. Node classification models don't rely on relationship information. Typical applications for node classification includes citation networks, YouTube Videos, Facebook Relationships etc. The training process can be performed by following below steps-

  1. The input graph is separated into two parts: the first one is train graph and the second one is test graph.
  2. The train graph is further split into a number of validation folds, each consisting of a train part and a validation part.
  3. Every model candidate is trained on each train part and evaluated on the respective validation part.
  4. The training process uses a regression algorithm and the evaluation uses specified metrics. Here, the first metric is called primary metric.
  5. The model with the highest average score according to the primary metric will win the training.
  6. The winning model will be retrained on the entire train graph.
  7. The winning model is examined on both the train graph and the test graph.
  8. The winning model is then retrained on the entire original graph.
  9. Finally the winning model will be registered in the model catalog.

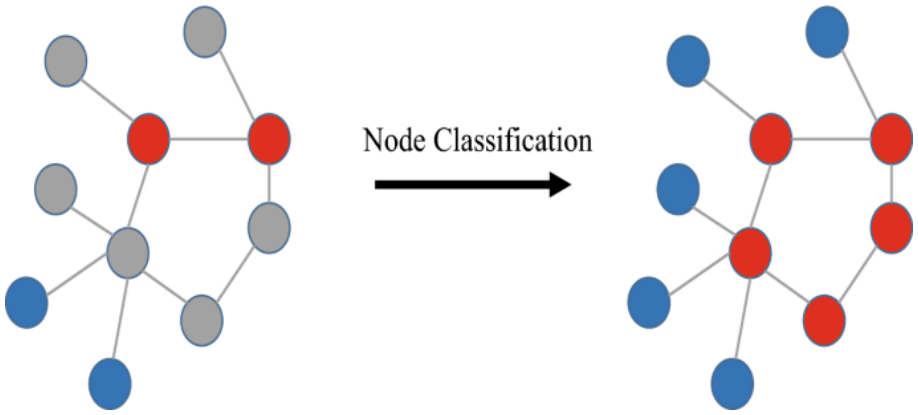

Figure 2.7: Node Classification

In Fig. 2.7, we have shown a pictorial representation of node classification.

- **Edge/Link Prediction:** In link prediction the task is to understand the relationship between entities in the graph and predict if two entities having connection in between.[7] For example, a recommender system can be treated as a link-prediction problem where the model is a set of users' reviews of different products. Here, the task is to predict the users' preference and tune the recommender system to push more relevant products according to the user's interest as recommendation.

- **Graph Classification:** Graph classification is a problem with many practical applications in various different domains. In Graph Classification the task is to classify the whole graph into different categories.[7] It is almost similar to Image Classification but the target change in graph domain. There is a wide range of industrial problems where graph classification can be used. For example, in Chemistry, Bio-medical, Physics where the model is given a molecular structure and asked to classify the target into meaningful categories. The applications of graph classification is large and can range from determining a protein is enzyme or not to categorize documents in NLP.[7] It accelerates the analysis of atom, molecule or any other structured data types.

- **Graph Visualization:** This is connected with the visual representation of graphs that depicts the structure and anomalies that may be present in the data. Therefore, It helps user to understand graphs in a more convenient way. Some advantages of graph visualization are depicted below-

  1. This will need less time to assimilate information from data. [7]
  2. One can achieve a better understanding of a problem by using graph visualization.

- **Clustering:** GNNs can obtain structured information from graphs. However, Clustering is the main tool used in Machine-Learning based marketing. [7] Moreover, they have already found powerful applications in various domains such as Route Planning, Fraud Detection, Network-Optimization etc.

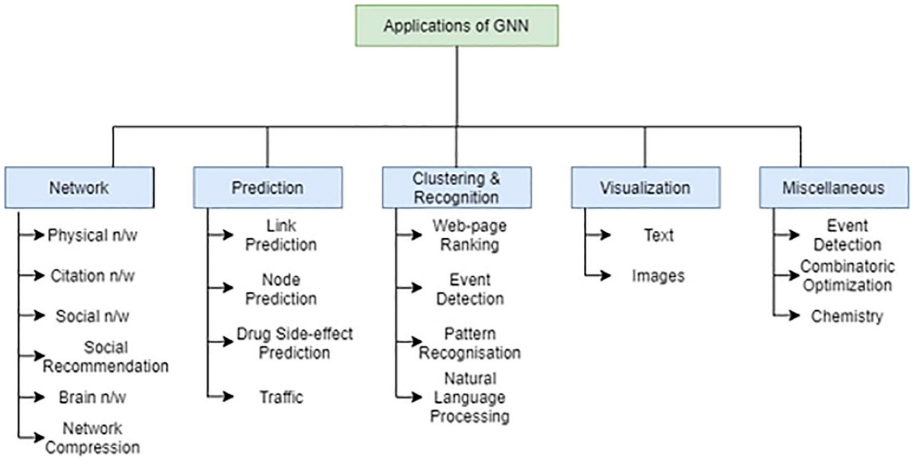

Figure 2.8: Applications of GNN

In the above Fig. 2.8 various applications of GNN are shown.

2.23 Some Real Life Applications of GNN

After understanding what types of analyzing GNN can perform, one must be wondering what are the real things can be done with GNN. Well, This section will give more in-sights into GNN's real-world applications. Let's discuss some pointers elaborately-

- **GNN in Natural Language Processing:** GNN is widely used in Natural Language Processing Now-a-days. Actually, this is also where GNN initially gets started. GNN approaches a problem from a completely different angle. GNN utilized the inner relations of words or documents to predict the categories.[44] For example, the citation network is trying to predict the label of each paper in the network given by the citation relationships and the words that are cited in other papers.

- **GNN in Computer Vision:** One successful employment of GNN in Computer Vision is using graphs to model the relationships between objects detected by a CNN based detector. After objects are detected from Images, they are fed into a GNN inference for prediction. The outcome of GNN inference is a graph that represent the relationship between different objects. Another important information of GNN in Computer Vision is image generation from graph descriptions. This can be interpreted as almost reverse of the process mentioned above.[44] The traditional way of image generation is text-to-image generation using auto encoder. Instead of using text for image description, graph to image generation provides more information on the semantic structures of the images.

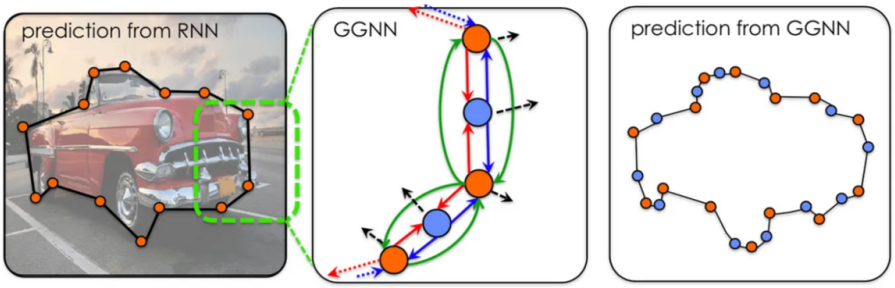

Figure 2.9: GNN in Computer Vision

In the Fig. 2.9, we have shown how GNN is used in computer vision.

- **GNN in Traffic:** GNN can be involved in forecasting traffic speed, road traffic density which is essential components for smart transportation system.[44] Traffic prediction problem can be addressed by using GNN considering the traffic network as a spatial-temporal graph where the nodes are sensors installed on roads, the edges are measured by the distance between pairs of nodes and each node has the average traffic speed within a specific time slot.

- **GNN in Physics:** To understand human intelligence one of the popular approach is to design a real world physical system.[44] By representing objects as nodes and relations as edges;GNN based reasoning can be performed on objects, relations in a more convenient and more effective way. Interaction networks can be trained to reason about the interactions of objects in a complex physical system.[44] It can make predictions about various system characteristics like collision-dynamics (rigid or non-rigid). It simulates these systems by object and relation based reasoning using deep neural network on graphs.

- **GNN in Chemistry:** GNN spread it's applications in chemistry industry too. Researchers can use GNN to know the graph structure of the molecular atoms. In these graphs, molecular atoms can be represented as nodes and chemical bonds can be represented as edges. GNNs can be used to both the cases - learning about the existing molecular structures and also discovering the new chemical structures.

- **GNN in Medical Science:** GNN has spread it's application in medical field also. In today's scenario, Electronic health records (EHRs) are used frequently to help physicians to take decisions by predicting medical events such as diseases, prescriptions, outcomes and so on. The key idea of making of these predictions is how to represent patient longitudinal data. Recurrent neural network is a popular model for patient longitudinal medical data representation from the view of patient status sequences. However, it cannot represent complex interactions among different types of medical information like temporal medical event graphs which can be represented by graph neural network.

- **GNN in Computer Network:** Since the computer applications are increasing the use of computer networks; so if we utilize the use of networks with machine learning, it will be much more easier to secure the networks rather than using traditional security patches.
  IoT firmware version prediction can be done by using GNN. The firmware information of IoT devices contains the information about the manufacturer, the type of device, device model and the firmware version. Knowing firmware version helps to build firmware knowledge graph for many security applications. Traditional firmware information identifying method only uses the content-based information, but not the structure information of the firmware.[44] Additionally, it lacks the use of timing information. Lacking of use of structural information can decrease prediction accuracy. In the same way lacking use of timing information can cause difficulties in predicting firmware version. In this context, to overcome the disadvantages of existing method, GNN comes into the field. It abstracts the directories or files of the correspond firmware into the nodes of the graph and abstracts the relationships into the edges of the graph.[44] Timing information including component creation time and component versions are also attached to the node properties so that the time sequence features can be introduced.

- **GNN in Text Classification:** Like images text is not also explicit that means text can not be treated as structured data over which GNN can be applied directly.[44] Not to worry; there are ways to convert text documents into structured data like graph of words or graph of sentences and then graph convolution can be done on the word graphs. Another approach to convert text documents into structured graph data is to use sentence LSTM which represents the entire sentence as a single node and consisting of sub-nodes in this case which are words. Document citation relation can also be used to construct graph with documents as the nodes. In this context, Text GNNs can be applied to learn embeddings for words and documents. When we are focused about text classification, the studies can be divided from two point

of views- one group of studies have shown that the success of any text classification model depends on the effectiveness of the word embeddings and another group is focused on learning the document and the word embedding together.[9]

- **Graph Embedding:** It maps graphs into vectors, preserving the relevant information on nodes, edges, and structure.[5]
- **Graph Generation:** It learns from sample graph distribution to generate a new but similar graph structure.
- **Handwriting Recognition:** Neural Networks are used rapidly to convert handwritten characters to a digital one so that machine can recognize and understand it.[18]
- **Stock Exchange Prediction:** The stock exchange market is effected by many external factors and it is difficult to track and understand.[18] However, a neural network can examine these factors carefully and can predict the daily prices. This helps the stockbrokers.
- **Community Detection:** It divides nodes into various clusters based on edge structure and learns from edge weights, and distance and graph objects similarly. [5]
- **GNN in Other Domains:** More practical application of GNN includes Human Behaviour Detection, Traffic Control, Molecular Structure Study, Recommender System, Program Verification, Social Influence Prediction. Below fig. shows a graph that models the relationship between people in a social network. GNN can be applied to cluster people into different community groups as shown in Figure 2.10 below-

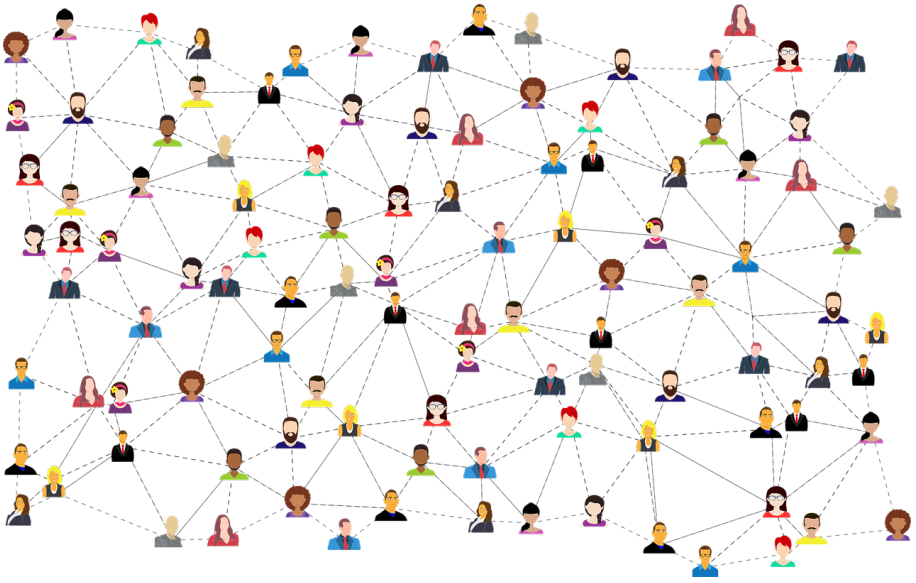

Figure 2.10: Graph of Social Network

2.24 Summary

In this chapter we have mostly discussed about Graph Neural Network (GNN). Over the past few years GNN become a powerful and practical tool for machine-learning tasks in graph domain. Now-a-days GNN is very important for many difficult machine-learning problems that can be modeled using graphs. AI and machine-learning can provide a wealth of personalized choices for worldwide users. As an example, all mobile and web applications try to provide users a better user experience based on their recent search histories and GNN helps to make it possible in a right way. We went through some graph theories. General Design Pipe-line mechanism is described for better understanding of GNN. Various types of learning process of neural-network is described. Later, we focus on an important machine-learning algorithm named "Back-Propagation Algorithm". In terms of application we divide GNN applications in few group and elaborate them accordingly. However,the power of GNN in modeling complex graph structures is truly good. I believe in the near future GNN will play an important role in AI's further development. We wrap-up this chapter with some real life applications of GNN in today's world as well as some theoretical aspects.

Chapter 3

Network Traffic Classifications

3.1 Introduction

Network Traffic is the amount of data that moves across a network within a given time period.Traffic classification is a very important topic now-a-days in the field of Computer Science. It is an automated process which categories computer network traffic into some number of application-aware classes according to various different parameters i.e chat, streaming.[21] In other words, we can say that Network Traffic Classification is the very first step of analyze and understand different types of applications flowing in a network. Traffic classification used in various applications such as Network Management, Network Design and most importantly Network Security.[21] It should be noted that the traditional classification techniques like Port Based, Pay-Load Based, Statistical Method requires patterns or extracted features. This process is time-consuming and expensive. Internet users reached worldwide 4.1 Billion in 2019, an increase of over 53% in compared to 2005 and 5.3% compared to 2018.[21] Due to excessive uses of internet and consequently huge amount of network traffic; traffic classification is getting difficult day by day. However, This process is very important to network service providers since they can monitor which types of network applications flow in that particular network by using this technique and they can manage the overall performance of a network. In below Fig. 3.1 different types of network traffic is shown-

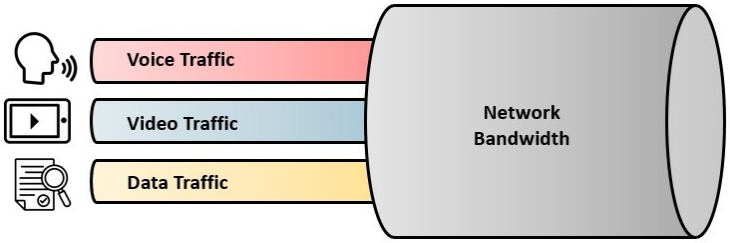

Figure 3.1: Network Traffic Types

3.2 Purpose of Traffic Classifications

Network Traffic Classification is an essential way to monitor network availability and activity to detect anomalies, maximize network performance and keep an eye out to attacks. Moreover, Classifying network traffic is the foundation for enabling QOS features such as traffic shaping and traffic policing on a particular network where QOS stands for Quality of Service. This process also helps an organization to determine whether their data is at risks and implement controls to reduce the risks.

3.3 Traffic Classification Techniques

Network Traffic Classification has generated great interests among researchers and industry field also. Several techniques were proposed and developed over the last two decades. This section depicts traffic classification methods and divides them into four categories based on their evaluation from past to present. They are like Port-Based Classification, Pay-Load Based Classification, Statistical Classification and Behavioral Classification.[21]part from that Machine-Learning based classification techniques are introduced which are used by many researchers and got effective results. Let's discuss these elaborately-

- **Port-Based Classification:** In the early days of internet; traffic identification and classification was not an issue at all. The earliest traffic classification solution uses packet's port numbers to classify the network traffic to their correspondence protocols. The port numbers are classified into three ranges- System Ports(0-1023), User Ports(1024-49,151) and the Dynamic Ports(49,152-65,535).[21] Port-Based classification involved identifying an application based on inspecting the packet header and matching it with the TCP/UDP port number registered with the Internet Assigned Number Authority (IANA).[21] But, unfortunately this traditional method fails and revealed inaccuracy and hence unreliability. The unreliability of this method comes from several causes. Firstly, some modern application flows with unregistered port numbers like P2P software; in this scenario the false negative results of the classifier increases. In the worse case, the non-legitimate applications able to hide themselves behind known ports to avoid being filtered.[21] Even in some cases it is impossible to know the actual port numbers.

- **Pay-Load Based Classification:** To overcome the challenges of Port Based Classification researchers have introduced Pay-Load based classification which sometimes called as Deep Packet Inspection (DPI). The most used Pay-Load based classifications are rely on inspection of packet content and match them with a deterministic sets of pattern. This method comes up with extreme accuracy. Moreover, this method is very important for Network Intrusion Detection System to find malicious activity into the network. However, like every other techniques it also has some major drawbacks. The efficiency of classifier goes down when encrypted network traffic and encapsulation comes in the scenario. It means examining of any encrypted packet is literally impossible by using this method i.e a lot of network traffic remains unclassified. Along with this more importantly inspecting of contents of a particular packet may be a breach of privacy policies and regulations. Moreover, this method imposes high computational cost and load on the classifier devices since it needs several instances of access to the packet contents. Eventually, it faces difficulties while coping up with large data flows and high speed network traffic.

- **Statistical Classification:** Statistical Classification is a rationale-based approach which depicts the statistical characteristics of network traffic flow to identify a particular application. This method uses a number of flow-level measurement; [11,12,8] for example, the packet duration time, packet inter-arrival time, packet lengths and traffic flow idle-time. These measurements are unique for a particular type of applications. Hence, it helps classifiers to differentiate various applications from each other. In order to perform classification based on statistical method classifier requires to involve Machine-Learning algorithms since they need to deal with various traffic patterns from large data-sets. ML algorithms are light-weight and less computationally costly in comparison with Pay-Load based classifications since they do not depends on DPI. Instead of that they uses flow level analysis information. However, this method is getting popular since it doesn't deal with packet contents hence can classify encrypted traffic too and data breach problem is omitted. ML-Based classification can be divided into Supervised Machine Learning Classifiers, Unsupervised Machine Learning Classifiers and Semi-Supervised Machine Learning Classifiers.
- **Behavioral Classification:** This classification technique concentrates on whole network traffic received by the host or end point; trying to identify the network application by examining the network traffic pattern generated by the target host. This is done by inspecting the generated traffic pattern based on how many hosts are contacted, with which transport layer protocol, on how many different ports behavioral traffic classifier tries to understand which types of applications are flowing on the target host. In below Fig. 3.2 different traffic classification methods are shown-

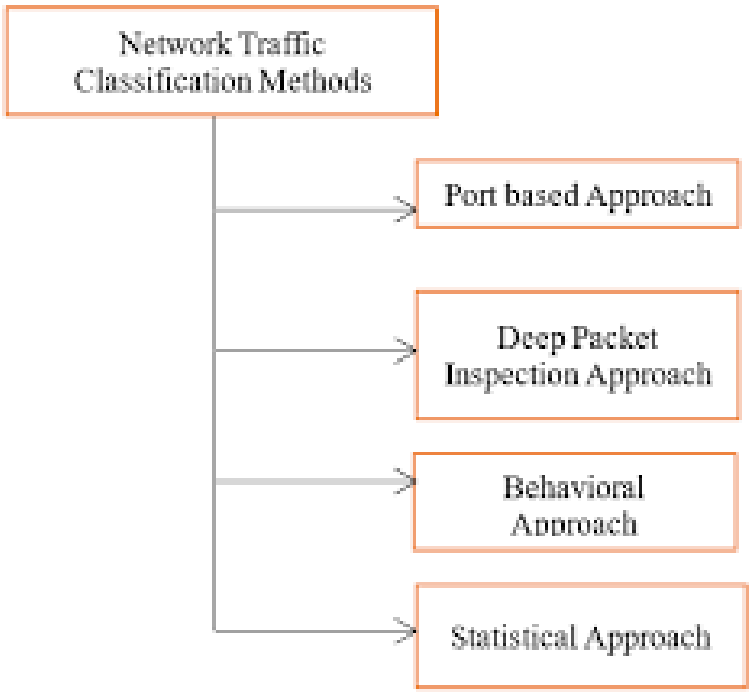

Figure 3.2: Network Traffic Classification Methods

3.4 Traffic Classification Metrics

Traffic classification metrics differentiate how accurately a technique or model is able to make decisions while represented with previously unknown data.[33] Suppose, we have traffic of class A; the goal is to classify packets which are belong to class "A" while presenting previously unseen mixture traffic. That is: **Input**: mixed traffic of packets and **Output**: Whether a unknown packet or flow belongs to class "A" or not.[33] However, there are some classifier evaluation metrics in ML exists-

- **Recall:** % of members of class "A" correctly classified as belonging to class "A". This is also known as sensitivity.[33] In other words, it is actually the fraction of relevant instances which are retrieved over the total amount of relevant instances.
- **Precision:** % of those instances that are truly members of class "A" among all those which are classified as class "A".[33]This is also known as positive predictive value.

3.5 Motivations for using Machine Learning Techniques

As soon as the traditional traffic classification techniques fail due to its unreliability; Machine Learning came in the field. There are some motivations for introducing machine learning in network traffic classifications. They are as follows-

- Researchers are looking for algorithm which is light-weight and less computationally costly. Machine Learning can suffice this need.
- The traffic classification experts must deal with increasing amount of internet traffic and transmission rates.
- The on-growing trends of traffic encryption possesses challenges to the researchers.
- The developers always hunt for finding new ways to prevent traffic being filtered and detected.
- Using a combination of techniques for supervised and unsupervised learning algorithm has shown promising results for network traffic classifications.
- ML algorithm is applied to classify unknown traffic from previously learned rule.
- In case of encrypted data and dynamic port it is getting harder to classify network traffic using traditional traffic classification method. Instead machine learning techniques can be more effective.

These were the most important reasons for researchers to move toward machine learning in the field of traffic classification. However, most of the machine learning techniques used for network traffic classification uses supervised and unsupervised learning while some of them uses hybrid learning which is also known as Semi-Supervised learning.

3.6 Machine Learning Based Traffic Statistical Classification

Now-a-days this is the most popular method of traffic classification. This technique can be implemented by performing few steps. In this section let's discuss it in details. First step is to represent the network traffic in form of flows which is nothing but the aggregation of packets that shares the five tuples- Source and Destination IP addresses, Source and Destination Port Numbers and TCP/UDP Protocol. Next step is to extract the statistical features at the flow level. The general process is described as follows-

1. **Data Collection:** This is first step for any traffic classification methods that must contains a sufficient number of targeted applications. There are two types of approaches used - Private Data Collection and Public Data Set Utilization. In general, raw traces are stored in PCAP files format for further processing.

2. **Flow Representation:** The captured raw traffic must be represented in a form that allows extraction of the statistical features for each connection. The flow representation is nothing but the aggregation of packets that shares the five tuples- Source and Destination IP addresses, Source and Destination Port Numbers and TCP/UDP Protocol. The packets collection may be unidirectional or bi-directional. Here, Unidirectional collection aggregates the packets that shares those five tuples for a single direction where Bi-directional collection aggregates the packets that shares those five tuples for both the direction.

3. **Feature Engineering:** This is the most important step for traffic classification technique as it is responsible for computing different matrices extracted from each flow. Generally feature engineering can be by using two steps- first one is Feature Extraction and second one is Feature Reduction.

4. **Data Set Preparations:** This is another crucial step in traffic classification technique. After extracting features and applying feature-selection algorithm, a data set containing historical data is ready to be used for building and testing the classifier using separate training and testing data sets. Training phase requires a large set of training data for optimal results. Similarly, accepting data set is required to examine the classifier properly and take decisions. Involving the same data set in both cases i.e training and Testing is a bad practice since the results could be misleading.

5. **Model Building:** The training data set generated in the last step is used to build a model that classifies network flows. There are various machine learning algorithms are being developed to solve tasks like classification and clustering. However, the selection of machine learning algorithm depends on types of task. However, in general two types of machine learning algorithms are in use likely supervised and unsupervised learning algorithm. Along with that a hybrid learning algorithm i.e semi-supervised learning algorithm is also there. In later section we will discuss this in details.

6. **Model Evaluation:** After building of the model and before deployment of the same in the production environment, the performance of the built classifier needs to be assessed. The performance metric is computed based on classification goal.

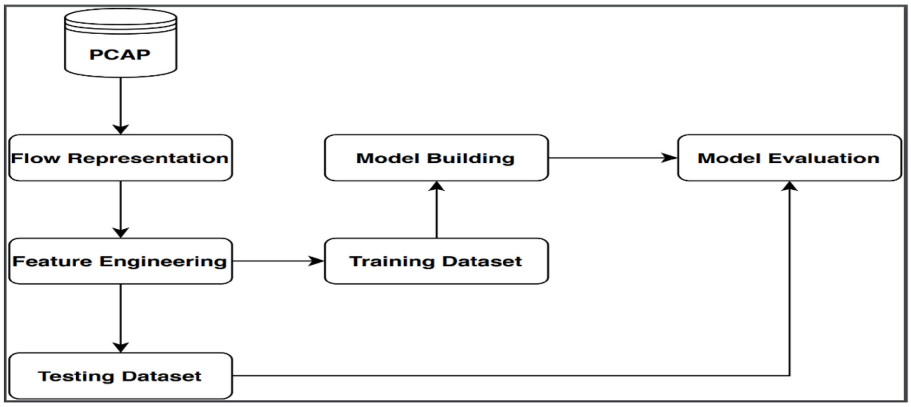

Figure 3.3: Supervised Learning Flowchart

In above Fig. 3.3, Supervised Learning Flowchart is displayed.

3.7 ML Algorithms for Network Traffic Classification

There are many machine learning algorithms which is used to classify network traffic. In this section we will discuss two supervised machine-learning algorithms namely Support Vector Machine and Decision Trees.

- **Support Vector Machine:** Support Vector Machine shortly known as SVM is used for both classification and regression. However, it is best suited for network traffic classification. In other words, we can say that Super Vector Machines are set of supervised learning methods that can be used for classification and regression. The goal of SVM is to create the decision boundary that can segregate n dimensional sample space into classes so that we can add new data point in the appropriate category in future. This best decision boundary is sometimes called as Hyperplane.
- **Decision Trees:** Like SVM this is also a supervised machine learning technique that is used for traffic classification. It is a tree structured traffic classifier where internal nodes represent the data set features, each branches represent the decision rule and leaf nodes represent the output.

3.8 Why SVMs are used

They are used in applications like face detection, intrusion detection etc.SVMs gain more popularity today because it can handles both classification and regression on linear as well as on non-linear data.[15] Another reason for using SVM is that it can find relationship between data without doing a lot of transformations. SVM can provide more accurate results when compare to other algorithms because of it's ability to handle small, complex data sets more efficiently. Moreover, it is more effective in high dimension spaces. According to various researches they works best for traffic classification problems. It also works well for unstructured and semi-structured data like text and images. In below Fig. 3.4, a typical SVM classifier is shown-

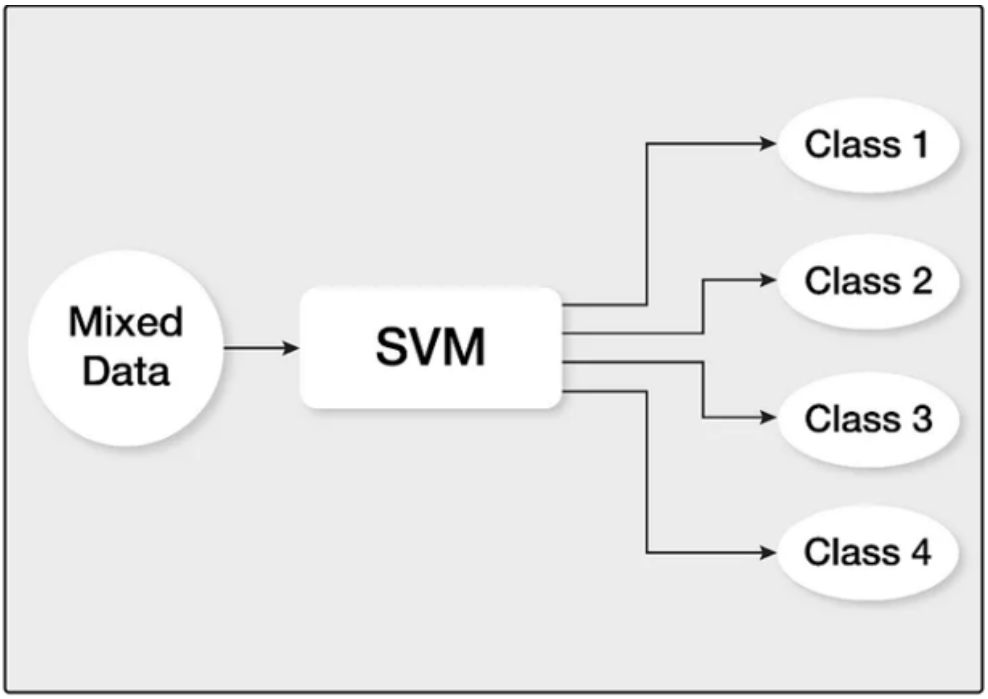

Figure 3.4: SVM Classifier

3.9 How an SVM (Support Vector Machine) works

A simple linear SVM works by making a straight line in between two classes. That means all data points on one side of the line will represent a category and data points on another hand of the line will be put into different category.[14] That implies there can be an infinite number of lines to choose. Actually SVM is an algorithm that takes the data as input and outputs a hyperplane that distinguishes those classes if possible. Let's discuss the mechanism of SVM with an example. Suppose, we have a data set like below and have to separate the red rectangles and the blue ellipses. So in this case our work is to find an ideal line that can separate the data sets into two classes. But, in this case we can see there is not an unique line to perform the job. Rather, we have an infinite number of lines that can separate two classes.[14] In this context SVM helps to determine which one is best for separation. Sample data set for understanding of SVM is shown in below Fig. 3.5 and in Fig. 3.6, Two Separating Lines are shown. Additionally, in Fig. 3.7 Optimal Hyperplane by using SVM is shown.

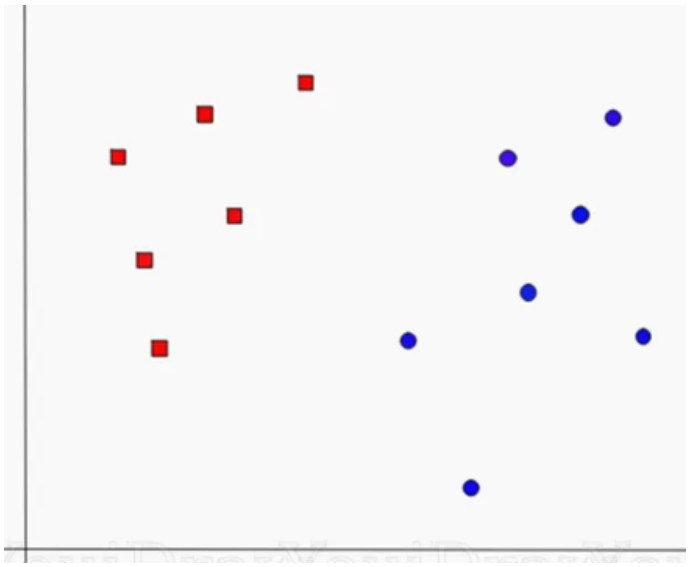

Figure 3.5: Sample Data Set

Let's discuss how SVM choose the ideal line for separation among infinite options. Suppose we have two lines here one is yellow and another one is green. Now the concern is which line between them best separates the data. In this case answer is yellow line since it is visually very intuitive that yellow classifies better than green one. But we need something concrete to fix the best separating line.[14] The green line in the image below is quite close to the red class rather than the blue class. Though, the green line classifies the current data set; however it is not a generalized line and in context of machine learning the main idea is to find more generalized separator.

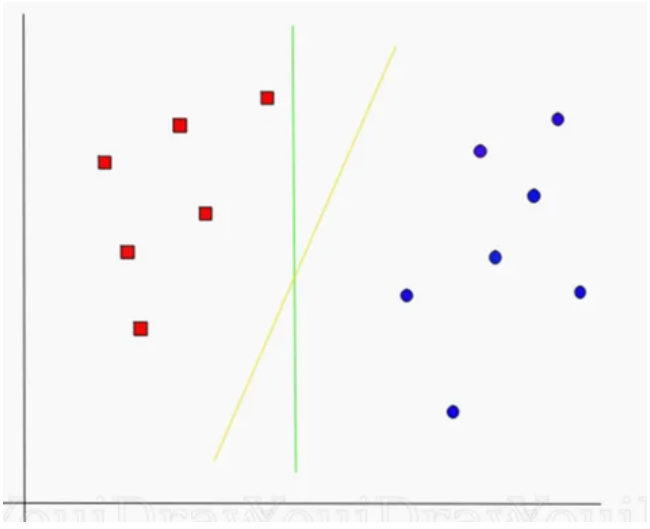

Figure 3.6: Two Separating Lines

According to the SVM algorithm first we have to find support vectors i.e the points that are closest to the separating line from both the classes. In the next step we have to calculate distance between the line and the support vectors. This computed distance is known as margin and our idea is to maximize the margin.[14]The hyperplane for which the margin is maximum is the desired optimal hyperplane. Thus SVM tries to draw a decision boundary in such a manner so that the separation between those two classes is as wide as possible.[14]

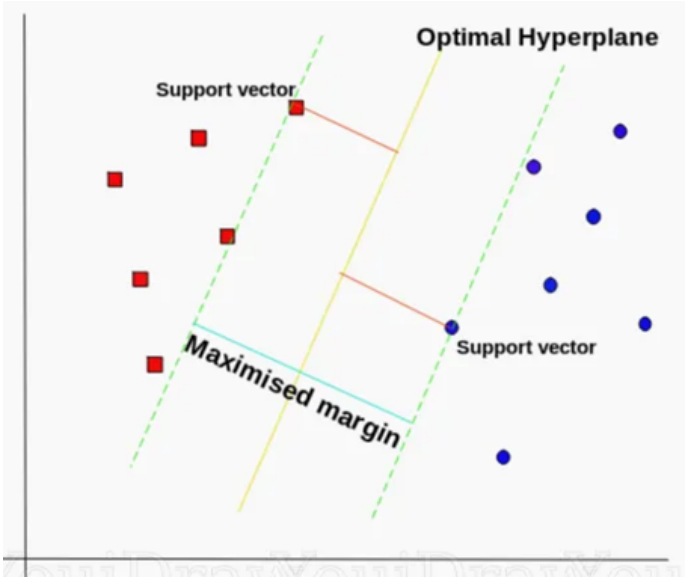

Figure 3.7: Optimal Hyperplane using SVM algorithm

3.10 Network Traffic Classification using SVM

Internet traffic control have attracted an increased amount of interest in the last few years.[35] From the security point of view, traffic classification can reduce the harm caused by attackers or avoid it. Moreover, accurate classification of various network traffic can help to identify the applications that are using network resources and facilitates the quality of Services(QoS) for different applications. From the resource utilization and Quality of Service (QoS) requirement, the network applications can be divided into few groups. In below Fig. 3.4 shows the different applications based on that we perform traffic classification processes.
As of now, we are dealing with supervised type of machine learning techniques to classify network traffic.[35] The pre-processing of data is very important for any supervised machine learning traffic classification technique. Initially, a data set was prepared manually by collecting raw network data form the packet header.[35] In this context, SVM is chosen since it has proven to be the most efficient machine learning algorithms for classifying network traffic.

Class	Representative Application
FTP	ftp
WWW	http, https
P2P	Bit torrent, Gnutella , Skype
NETBIOS	NetBios-ns, NetBios-ds
DNS	Dns
MAIL	Smtp
TELNET	Telnet

Figure 3.8: Internet Traffic Classes

3.10.1 Related Research

Researchers mostly concentrates on monitoring monitoring network traffic, try to detect anomalies in the network and cyber-attack traffic pattern.[35]They represent their work on machine learning based traffic classification method to identify command and control traffic of IRC- based botnets with a compromised set of hosts that are collectively commanded using IRC (Internet Realy Chat).[35] Their work is divided into two parts- a) differentiating between IRC and Non-IRC traffic and b) distinguishing between botnet and real IRC traffic.
In [23],the researchers have created an extensive E-dare system for early detection of malicious code in network traffic.
The research[28],provides us with the idea of feature selection in which traffic classification has been examined by evaluating three classification approaches based on transport layer ports, host behavior and flow features.In this work, the researchers mainly interested about different flow features that they have used in their experiment. In this work they they have taken classification approaches like SVM and port based classification into account and observed advantages as well as limitations of each approach. In addition with that they have also discussed the methods of overcoming those limitations.
In [32], Classification of network traffic into few broad classes like WWW, mail, attack etc is performed by Andrew Moore. His work's primary idea is to classify the network traffic broadly based on the similar properties of the same class. His work provide us with a over all idea of network traffic classification.
In[31],The researchers have studied on network security risk and analyzed the traditional or existing risk evaluation method. After that they have proposed a new method based on SVM. This is different from the traditional one since it is proved to be a novel type of algorithm. Compared to ANN about the classification point of view, Generalization Performance, learning and testing time, they proved that the SVM method is superior over others since it possess better generalization performance and less learning and also testing time.
The book [25] neural network gave the idea how to understand the basics of algorithm and Herv´e Abdi [20]additionally threw some light on the same. Vikramaditya Jakkula[26] gave a overview of SVM in connection with traffic classification which actually depicts the theoretical aspects for pattern classification.

In this work[24], researchers tried to make sense about the working of back propagation algorithm and suggested that this algorithm is used by layered feed forward neural networks.
Ruixi Yuan & Zhu Li & Xiaohong Guan & Li Xu in their work [42] have performed network traffic classification in broad categories of the applications using SVM algorithm. They suggested that, port based traffic classification technique is not so much suitable in case of dynamic port allocation and encrypted network traffic. They have basically work with the methods that classify the Internet traffic into broad application categories according to the network flow parameters collected from the packet headers. The experimental results using traffic from campus backbone showed that an accuracy of 99.42% was achieved with the regular biased training and testing samples, and an accuracy of 97.17% was achieved when un-biased training and testing samples were used with the same feature set. Finally, they suggested that since all the feature parameters are computable from the packet headers, the proposed method is also applicable to encrypted network traffic. [35]

3.11 Types of SVM

There are specific types of SVMs that can be used for particular machine-learning problems,like Support Vector Regression (SVR) which is extension of Support Vector Classification (SVC). There are two types of SVM. Each of them used for different applications-

- **Simple/Linear SVM:** This is generally used for linear regression and classification problems. In other words we can say that Linear SVM is used for linearly separable data. Those data Set which can be classified into two classes by using a single straight line is known as Linearly Separable Data and the classifier used is known as Linear SVM classifier.[14]

- **Kernel/Non-Linear SVM:** Kernel functions are used in SVM to solve regression and classifications problems. SVM uses kernel to transform linearly inseparable data into linearly separable one, thus finding an optimal boundary for possible outputs. Non-linear data which can not be separated using a straight line can be classified using non-linear SVM.[14] It posses more flexibility for non-linear data over linear data since one can add more features to fit a hyperplane instead of a two-dimensional space. [14] SVM uses various types of Kernel functions such as linear kernel, polynomial kernel etc. These Kernel functions will be discussed in the later section.

3.12 Advantages of SVM

SVMs are most powerful supervised machine learning algorithm. It is superior than many other traditional algorithms. Some advantages of SVM are listed below-

- This works for data sets with multiple features like financial data, medical data etc.

- SVM works well when there exists a clear margin of separation between two classes.[15]

- It is effective on such cases where number of features are more than number of data points.

- It can perform effectively on high dimensional spaces.

- It uses a sub-set of training points in decision function, which makes it memory efficient.

- SVM has Convex Optimization nature which is very important since we are assured of optimality in results.
- There exist many algorithms for classification purpose but among them SVM is most popular because of better accuracy in results.[15]
- The execution time is less in compare with other algorithms like Artificial Neural Network.

3.13 Disadvantages of SVM

Previously we have discussed some advantages of SVM. But, now we will discuss some disadvantages of SVM. This is depicted as below-

- SVM is not that much suitable for larger data sets.[15]
- Training time is long for large data sets.
- More the features taken into consideration; higher the complexity is.
- Unsatisfactory performance on high-noise. When the data has noise it has many overlapping points. In this case, drawing of a clear hyperplane is difficult.
- Selecting appropriate kernel function is not an easy task.
- SVM is not suitable for data sets with missing values. Instead of that it requires complete data sets.
- Algorithm complexity and memory requirement is high.

3.14 Kernel functions used by SVM

The function of Kernel is to take data as input and transform it into required form. SVM has used various types of kernel function. These can be linear, Polynomial etc. Few of them are listed below-

- **Linear Kernel:** These types of functions are commonly used for text classification since these types of problems are linearly separable in most of the cases.[14] This kernel function is faster than any other functions. Below is the function that defines the linear kernel-
  $f(X) = w^T * X + b$
  In the above mentioned equation, W is denoted as the weight vector, X is the data which is to be classified, b is the linear coefficient. This equation defines the decision boundary.
- **Polynomial Kernel:** This type of function is not used very often since it is not as computationally efficient as others and it's predictions are not so good as compare to others.[13] Below is the function which defines polynomial kernel-
  $f(X1, X2) = (a + X1^T * X2)^b$
  Here, f(X1, X2) defines the polynomial decision boundary that will separate the data. X1, X2 represents the data. The polynomial kernel function has some parameters which can be tuned to enhance it's overall performance including the degree and the co-efficient of the polynomial.

- **Sigmoid Kernel:** This is more useful in neural networks than SVM. But there are special cases. This kernel function is similar to a two-layer perceptron model of the neural network, which works as an activation function for neurons.[13] Below is the function for sigmoid kernel-
  $f(X, y) = tanh(alpha * X^T * y + C)$
  In the above equation, alpha is a weight vector and C is some off-set value to account for some mis-classification of data that can happen.

Besides the above mentioned there are many kernel functions exist in the scenario. Like, ANOVA radial basis, Laplace RBF etc. One should be cautious while choosing kernel function for use.

3.15 K-Nearest Neighbors Method for Classifying Users Session in E-Commerce Scenario

The most important application of KNN algorithm in E-Commerce domain is recommender system.[41] In general product recommendation in online stores may be classified into two categories- A) Content based recommendation and B) Collaborating filtering (CF) recommendation.[41] Content-based recommendation systems are based on similar types of products i.e the similarity of the products and in another side CF recommender system tries to find similar users and recommend what similar users like. In this type of recommendation system we classify the users into clusters of similar types and recommend product to each users according to the priority of the cluster. The main task in CF recommender system is computing the all-to-all similarity between customers to form a group of most similar customers in terms of preferences.[41] The most popular method for forming neighbourhood is a centre-based scheme which forms a neighbourhood for a particular customer just by selecting the K-nearest other customers.[37] Categorization of items in very large data set was formulated as a supervised classification problem where the categories are the target classes and words composing description of the products are the features. Besides recommender systems other application of KNN algorithm in E-commerce is based on web content analysis.[41]The working methodology of the entire process is discussed in details in the later section.

3.16 Related Research

[40], A K-nearest neighbour algorithm is a supervised learning technique used in pattern recognition for classification, moreover it can also be used for estimation and prediction. In [39], K-NN based method was used in a hierarchical approach to the problem for large scale text based categorization on an e-commerce website. Categorization of items was treated as a supervised classification problems where the categories are the target class and words compromising of some textual description of the items that are the features.
In [22], the researchers have shown two most commonly used techniques to determine the neighborhood size. They are- 1) correlation thresholding technique and 2)best K-neighbour technique. In correlation thresholding technique, the nearest neighbors are customers and in case of best K-neighbour technique; the best K correlates are selected for the neighbors.
In [27], the researchers applied a clustering based k-NN approach which combines the k-NN and the iterative clustering algorithm. The iterative clustering approach allows them to solve the data sparseness problem by fully exploiting the voting information first. Then, a cluster-based k-NN algorithm is applied to improve the performance of CF.

3.17 Working Methodology

1. **Analysis of the website contents and distinguishing session states:** In general, a website consists of many pages; each of which is corresponds to some functions like reading general information about the store, searching for a particular product, adding product to cart etc. At any certain time different users may have accessed the same website and perform different functions that means there may be many active user sessions on the server.[41] The website contents was analyzed in details and as a result each page was assigned to one of the following 15 session states: Home- the homepage of the web-store, Information- pages containing general information about the web-store, Entertainment- pages with entertainment contents, shipping- pages that contains information about shipping charges, terms and condition about shipping [41]i.e on above certain order amount eligible for free delivery, shipping check out- shipping information during the check out process, Browse- browsing information, Search- interactions in connection with the searching for a product, Details- pages containing product information, Add- add to shopping cart, $Register_{Success}$ -successful registration of an user,[41] $Register_{Try}$ -pages connected with the registration process except successful registration, $Login_{Success}$ -successful users logging into the website, Log_{Off} -user logout,[41] $Checkout_{Try}$ - pages which are connected with check out process other than confirmed purchase, $Checkout_{Success}$ -purchase confirmation.

2. **Source Data:** A single user session in an online store is nothing but a sequence of HTTP requests sent to the server by client (Internet Browsers).[41] All HTTP requests received by the server are registered in a server-access log. The below data are written in a log-file for each request- client IP address, identifier and username, date and time, HTTP method used, which version of HTTP protocol used, URI (Uniform Resource Identifier) of the requested server resource, HTTP status code, volume of data transferred in response to a HTTP request, a referrer that linked the user to the store website.[41]

3. **Reconstruction and description of user sessions:** Data is read from log files using program. Based on IP addresses; request streams for individual users were distinguished. Each user's session is described using 23 session features. [41] The first group of features contains 15 elements among 23 connected with visits to the session states occurred in a session. A variable $V_{CheckoutSuccess}$ is a boolean variable. This variable value is equal to one when purchase transaction made successfully and otherwise it is zero.[41] Other variables like V_{Home}, $V_{Information}$, $V_{Entertainment}$, $V_{Shipping}$, $V_{ShippingCheckout}$, V_{Browse}, V_{Search}, $V_{Details}$, V_{Add}, $V_{RegisterSuccess}$, $V_{RegisterTry}$, $V_{LoginSuccess}$, V_{Logoff}, $V_{CheckoutTry}$ are connected with numbers of visits to the corresponding session states occurred in session.[41] The second group of features consisting of 6 elements as below-

   - $V_{Requests}$ - gives the number of HTTP requests in one session. [41]
   - $V_{Transfer}$ - amount of downloaded data in kilobytes in session. [41]
   - V_{Pages} - number of visited page in session.[41]
   - $V_{Duration}$ - the session duration in seconds.[41]
   - $V_{Timeperpage}$ - mean time per page in seconds and V_{Source} - the source of the visit.[41]

Except the above mentioned, there are last two session characteristics as described- V_{Isbot} and $V_{Isadmin}$ are boolean features which indicates whether the session is performed by any automated bot or by the website admin himself/herself.[41]

4. **Problem Formulation:** In this scenario, the research goal is to find user's session that ended with successful purchase. So, the session is classified based on purchase transaction was made in session or not. For that reason, two session classes are taken into account. One is browsing session and another one is purchasing session. [41] K-NN classifiers for different values of K were built based on the training set and their performance was examined based on the test set.

However, the performance of a classifier can be evaluated by measuring accuracy, error rate and sensitivity. The accuracy is nothing but the percentage of all correct classifications and can be formulated as follow-

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

By default, the error rate is the percentage of all incorrect classifications and can be formulated as below-

$$ErrorRate = \frac{FP + FN}{TP + TN + FP + FN} \tag{3.2}$$

And lastly sensitivity is the percentage of correct classifications of buying sessions. It is basically, a probability of buying sessions and can be formulated as below-

$$SesitivityRate = \frac{TP}{TP + FN} \tag{3.3}$$

In the above three equations,
TP= True Positive
FP= False Positive
TN= True Negative
FN= False Negative
In the above context, True Positive is the number of correctly classifying buying sessions, False Positive is the number of browsing sessions that are incorrectly classified as browsing sessions, [41] True Negative is the number of correctly classifying browsing sessions, False Negative is the number of buying sessions wrongly classified as browsing sessions.
However, this measurement is mainly useful for retailer's to identify the potential buyers in an online store which makes them able to apply different offers to the concerned customers to encourage them buying that particular product. For future work, the research is continued on K-NN classification of e-customer session by examining various similarity measures and voting schema among the K-nearest neighbours.[41]

Chapter 4

Tools and Data Sets

4.1 Introduction

In this chapter, we survey various tools and open data sets for experimentation on GNN based traffic classifiers. At first, we have discussed about IGNNITION framework with it's basic overview and why it should be used. Then we go for PyTorch Geometric (a Python library) along with it's advantages and some reasons for choosing PyTorch Geometric over some other libraries. In the later section, we have some open data sets and briefly discussed their purpose, contents,limitations etc. The three data sets we have discussed in this section are namely- ASNM Data Sets, IP network traffic flows labeled with 75 apps and Computer network traffic data set.

4.2 Tools

4.2.1 IGNNITION

IGNNITION[1] [36] This is an ideal framework for beginners who has just stepped into the world of neural network programming. It is developed by passionate network experts specially for research scientists and engineers. This can build a custom GNN by adapting any particular network scenario. However, one can design and run his/her own GNN model by following the below steps-

- Define a GNN architecture with an YAML interface.
- Adapt data set.
- Execute the training just by writing three lines of code.

[1] https://ignnition.org/

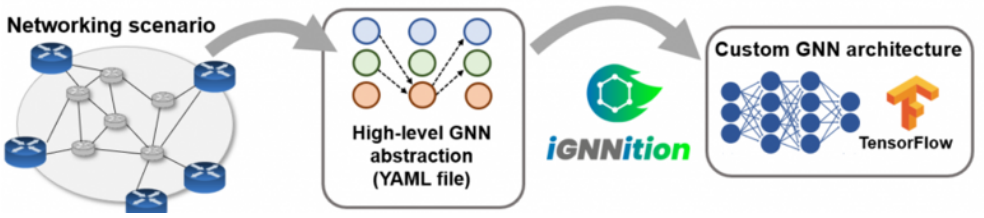

Figure 4.1: Overview of IGNNITION

In above Fig. 4.1 Overview of IGNNITION is shown for basic understanding.

4.2.2 Why IGNNITION?

There are few advantages that made it popular. Let's discuss those pointers in details-

- **High-level abstraction:** With the help of IGNNITION user can design their own GNN model via a high-level abstraction called Multi-stage Message Passing (MSMP) graph. This allows user to define model in a YAML file without focusing the underlying mathematical formulation. As a result, a typical network engineer and researcher is able to produce a functional GNN prototype according to specific problem without prior knowledge of ML programming libraries like Pytorch, TensorFlow etc.

Figure 4.2: High Level Abstraction

- **No TensorFlow Code:** IGNNITION has targeted user groups who have no prior experience or have some little knowledge about neural network programming. A GNN model can be defined easily by using YAML. Then the concerned model can be trained by using simple code while other well-known GNN framework requires to deal with complex tensor-based programming.
- **High Flexible Design:** IGNNITION provides very flexible and user friendly interface that supports all the popular GNN architecture and provide flexibility to implement a wide variety of existing GNN architectures. Additionally, it enables to combine individual components(e.g., Graph Convolution Network, Graph Recurrent Networks etc) of them to create a new custom design.

- **Easy Debugging:** IGNNITION provides an advanced error checking way that identifies bugs as well as show how to fix them. It is helpful for users.
- **Easy Integration:** It comes with an interface which enable users to easily feed their model with data sets from different sources.
- **High Performance:** It has been tested over many GNN architecture and after examining the results; it can be observed that it is as efficient as TensorFlow implementation coded by experts. This is the most important reason for choosing IGNNITION over other GNN frameworks.

4.3 PyTorch Geometric

This is a python library used for geometric deep learning. It is used to implement graph neural network on irregular structure such as graphs, point clouds etc. PyTorch Geometric is capable of running both on CPU and GPU with it's optimized throughput. It contains various methods for geometric deep learning which can be utilized to perform various tasks like node classification, link prediction etc. This is a powerful framework which offers researchers to easily implement graph neural networks for different kind of applications related to structured data.

4.3.1 Advantages

- **Optimized throughput for highly sparse and irregular data:** PyTorch Geometric comes with CUDA kernels for sparse data and mini-batch handlers for varying size. This makes it possible for users to make the most of their GPU resources while dealing with graph-structured data.
- **Both GPU and CPU computing support:** This can provide support for both GPU and CPU programming, that makes it accessible to a large variety of users irrespective of hardware they have.
- **Easy to use:** It contains easy to use mini-batch loaders which enables researchers to quickly implement learning tasks related to graph structures with minimum of effort.
- **Comprehensive library of methods:** PyTorch geometric is consisting of various method for deep learning from published papers which provide users chance of applying latest research to their data.

4.3.2 Reasons to use PyTorch Geometric

This is a very powerful library used generally for graph based machine learning tasks. It is an excellent option while working with graph-structured data or complex relational data. There are many reasons why this is used now-a-days. Let's go through some of these-

- **Rich graph data representation and handling:** Graphs are convenient way to represent complex relationships and PyTorch geometric is good at handling graph data. With it's flexible data structures, PyTorch Geometric makes it easy to represent and work with graphs along with node feature, edge feature etc. It makes this library a better choice for different types of graph based task like social network analysis, recommendation systems and more.

- **Extensive library of GNN layers and models:** PyTorch Geometric is made up of wide range of pre-built GNN layers and models. This allow users to experiment with different graph architectures and find the best solution for specific problems.
- **Seamless integration with PyTorch:** As an extension of famous PyTorch framework PyTorch Geometric seamlessly integrates with the PyTorch eco-system. That is why one can easily take advantages of existing PyTorch functionalities with various optimization algorithms.

4.4 Data Sets 1

4.4.1 ASNM Data Sets

This data sets generally contains many features that depicts various properties and characteristics of TCP communication. These features are called Advance Security Network Metrics and designed to help differentiate between legitimate and malicious connections. ASNM features are collected from tcpdump captures and do not perform deep packet inspection during the computation.

4.4.2 Purpose

ASNM data sets can be used for machine learning based network traffic classifications based on labels indicating presence of legitimate/ malicious communication.

4.4.3 Types of ASNM Data Sets

The following listing contains references to description of particular data sets-

- **ASNM-NPBO Data Set:** This data set contains non-payload-based obfuscation techniques applied onto malicious and some of legitimate traffic also. It was created back in 2015.
- **ASNM-TUN Data Set:** This data set contains tunneling obfuscation techniques applied onto malicious traffic. It was created in 2014.
- **ASNM-CDX-2009 Data Set:** This data set contains ASNM features extracted from tcpdumps of CDX 2009 data sets. However, it misses some newer ASNM features. It was created in the year of 2013.

4.4.4 Limitation

The limitation of this data set is related to ASNM features which can be intentionally influenced by an attacker to match the behavioral characteristics of legitimate traffic. However, this limitation is addressed in the last ASNM data set that deals with non-payload based obfuscation of network traffic in this case it is ASNM-NPBO data sets.

4.5 Data Sets 2

4.5.1 IP Network Traffic Flows Labeled with 75 Apps

The data presented here was collected in a network section from Universidad Del Cauca, Popayán, Colombia by performing packet captures at different hours during morning and afternoon over six days (April 26,27,28 and May 9,11,15 of the year 2017). A sum of 3.577.296 instances were collected and they are stored in a CSV (Comma Separated Values) file. Most of the network classification data sets aimed to determine the type of application an IP flow holds. But this data sets goes in more depth by generating machine learning models which is capable of detecting specific applications like Facebook, Instagram, YouTube etc from IP flow statistics.

4.5.2 Content of the data set

This data sets consist of 87 features. Each instance contains the information of an IP flow generated by a network device i.e., source and destination IP addresses, ports, inter-arrival times, layer 7 protocol (application) used on that flow as the class. Most of the attributes are numeric type.

4.6 Data Sets 3

4.6.1 Computer Network Traffic

Computer network traffic data - A- 500K CSV contains summary of some real network traffic data from the past. This data set has approx 21K rows which covers 10 local work station's IPs of three months.

4.6.2 Content of the data set

Each row of the data set contains four columns.

- date: yyyy-mm-dd (from 2006-07-01 through 2006-09-30)
- l_ipn : local IP (coded as an integer from 0-9)
- r_asn : remote ASN (an integer which identifies the remote ISP)
- f: flows (count of connections for that day)

4.7 Summary

In this chapter, at first we have discussed about IGNNITION framework with it's basic overview and why it should be used. Then we go for PyTorch Geometric (a Python library) along with it's advantages and some reasons for choosing PyTorch Geometric over some other libraries. In the later section, we have some open data sets (three in this case) which can be accessed totally in free of cost and briefly discussed their purpose, contents, limitations etc. The three data sets we have discussed in this section are namely- ASNM Data Sets, IP network traffic flows labeled with 75 apps and Computer network traffic data set.

Chapter 5

Experiments

5.1 Introduction

Graph Neural Network (GNN) is one of the popular machine learning technique - due to their capacity to recognize morphological topologies in graph-based data. GNN's can be used in in various real-world scenarios, such as social networking sites, biology, telecommunications, etc.

In this thesis we have tried to build a graph neural network model based on network traffic characteristics. We first extract information about network traffic from packet captures (`.pcap` files). In addition to considering the relationship between network traffic, deep learning methods require extracting many features, which is a complicated and time-consuming process. We have extracted information like, graph topology structure and node features expressed as vectors. We have used this data to build a graph neural network model and use it for network traffic classification.

5.2 Brief Description

In this section we describe the workflow of the proposed method. Given a collection of applications with and without labels, the following actions are taken:

1. Collecting the application's network traffic data
2. Extracting network traffic to create a heterogeneous graph from the hosts and traffic, with edges determined by the traffic
3. Updating the hidden state of each node and combining adjacency data with node attributes to construct the GNN model with the network traffic graph
4. Determining un-classified network traffic as being either malicious or benign based on their extracted features

In our experiments we have used a network traffic data set available from Kaggle[1].

[1] https://www.kaggle.com/jsrojas/ip-network-traffic-flows-labeled-with-87-apps

The data set was collected in a network section from Universidad Del Cauca, Popayán, Colombia by performing packet captures at different hours, during morning and afternoon, over six days. A total of 3,577,296 instances were collected and are stored in a CSV (Comma Separated Values) file.

This dataset contains 87 features. Each instance holds the information of an IP flow generated by a network device i.e., source and destination IP addresses, ports, interarrival times, layer 7 protocol (application) used on that flow as the class, among others. Most of the attributes are numeric type but there are also nominal types and a date type due to the Timestamp.

After initial pre-processing, we have used the data to build a GNN model. We have used PyTorch Geometric[2] for implementing the model.

[2]https://pyg.org/

Chapter 6

Conclusion and Future Work

6.1 Conclusion

With the proliferation of Internet, more and more application services have emerged. The network management activities such as quality of service (QoS) setting, network policy, network security as well as intrusion detection, majorly depend on the accuracy of network traffic classification for applications. Traditional classification schemes, such as simply checking source/destination IP addresses in the network layer or source/destination port numbers as well as protocols in the transport layer - are not always effective because the traffic might pass through different Network Address Translation (NAT) or Virtual Private Network (VPN), or they may not utilize the standard port numbers or fields. Therefore, for proper QoS provisioning, we need manual verification or identification of the network application traffic. This tasks are tedious and are prone to errors.

So, we need an application-based online and offline automated network traffic classification, which is one of the important aspect for network management.

In this thesis we have tried to build a GNN model that can be used to classify network traffic generated by different applications. This technique relies on the extraction of topological data in the network traffic. Experiences from the experiments have shown GNN based network traffic classifier as a promising and effective solution.

6.2 Future Work

A thorough assessment of the proposed technique using actual datasets would validate GNN's acceptability compared to the state-of-the-art techniques. We have left it as a future work.

Bibliography

[1] Advantages of neural networks – benefits of ai and deep learning. https://www.folio3.ai/blog/advantages-of-neural-networks//. Accessed: 2023-03-17.

[2] All you need to know about graph embeddings. https://www.simplilearn.com/what-is-graph-neural-network-article. Accessed: 2023-03-19.

[3] Backpropagation in data mining. https://www.geeksforgeeks.org/backpropagation-in-data-mining//. Accessed: 2023-03-19.

[4] A beginner's guide to graph neural networks. https://www.v7labs.com/blog/graph-neural-networks-guide/. Accessed: 2023-04-25.

[5] A comprehensive introduction to graph neural networks (gnns). https://www.datacamp.com/tutorial/comprehensive-introduction-graph-neural-networks-gnns-tutorial/. Accessed: 2023-03-22.

[6] Deep learning on graphs for computer vision — cnn, rnn, and gnn. https://www.gnu.org/software/octave/. Accessed: 2023-03-14.

[7] Graph neural network and some of gnn applications: Everything you need to know. https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications. Accessed: 2023-03-16.

[8] Graph neural networks: An overview. https://theaisummer.com/Graph_Neural_Networks/. Accessed: 2023-02-16.

[9] How to use graph neural networks for text classification? https://analyticsindiamag.com/how-to-use-graph-neural-networks-for-text-classification//. Accessed: 2023-05-11.

[10] Introduction to graph neural network (gnn). https://www.analyticssteps.com/blogs/introduction-graph-neural-network-gnn/. Accessed: 2023-02-23.

[11] An introduction to graph neural network(gnn) for analysing structured data. https://towardsdatascience.com/an-introduction-to-graph-neural-network-for-analysing-structured-data. Accessed: 2023-03-16.

[12] Knn algorithm - finding nearest neighbors. https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm. Accessed: 2023-03-20.

[13] Seven most popular svm kernels. https://dataaspirant.com/svm-kernels/. Accessed: 2023-04-23.

[14] Support vector machine algorithm, explained with code examples. https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/. Accessed: 2023-04-23.

[15] Support vector machines(svm) — an overview. https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/. Accessed: 2023-04-23.

[16] What are the advantages and disadvantages of artificial neural networks? https://www.tutorialspoint.com/what-are-the-advantages-and-disadvantages-of-artificial-neural-networks/. Accessed: 2023-03-17.

[17] What is graph neural network? | an introduction to gnn and its applications. https://www.simplilearn.com/what-is-graph-neural-network-article. Accessed: 2023-03-19.

[18] What is neural network: Overview, applications, and advantages. https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-neural-network/. Accessed: 2023-04-02.

[19] A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks*, 32(1):4–24, 2021.

[20] H. Abdi, D. Valentin, and B. Edelman. *Neural networks*. Number 124. Sage, 1999.

[21] A. Azab, M. Khasawneh, S. Alrabaee, K.-K. R. Choo, and M. Sarsour. Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks*, 2022.

[22] Y. H. Cho and J. K. Kim. Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert Systems with Applications*, 26(2):233–246, 2004.

[23] Y. Elovici, A. Shabtai, R. Moskovitch, G. Tahan, and C. Glezer. Applying machine learning techniques for detection of malicious code in network traffic. In *Proceedings of the 30th Annual German Conference on Advances in Artificial Intelligence*, KI '07, page 44–50, Berlin, Heidelberg, 2007. Springer-Verlag.

[24] C. Gershenson. Artificial neural networks for beginners. *CoRR*, cs.NE/0308031, 2003.

[25] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.

[26] V. Jakkula. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37(2.5):3, 2006.

[27] X.-M. Jiang, S. Wg, and W.-G. Feng. Optimizing collaborative filtering by interpolating the individual and group behaviors. pages 568–578, 01 2006.

[28] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: Myths, caveats, and the best practices. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, New York, NY, USA, 2008. Association for Computing Machinery.

[29] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[30] G. Leinhardt, O. Zaslavsky, and M. K. Stein. Functions, graphs, and graphing: Tasks, learning, and teaching. *Review of Educational Research*, 60(1):1–64, 1990.

[31] C.-C. Li, A.-l. Guo, and D. Li. Application research of support vector machine in network security risk evaluation. In *2008 International Symposium on Intelligent Information Technology Application Workshops*, pages 40–43, 2008.

[32] A. W. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '05, page 50–60, New York, NY, USA, 2005. Association for Computing Machinery.

[33] T. T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, 10(4):56–76, 2008.

[34] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, and C. E. Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *CoRR*, abs/1902.10191, 2019.

[35] A. Pradhan. Network traffic classification using support vector machine and artificial neural network. 10 2011.

[36] D. Pujol-Perich, J. Suárez-Varela, M. Ferriol, S. Xiao, B. Wu, A. Cabellos-Aparicio, and P. Barlet-Ros. Ignnition: Bridging the gap between graph neural networks and networking systems. *IEEE Network*, 35(6):171–177, 2021.

[37] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, EC '00, page 158–167, New York, NY, USA, 2000. Association for Computing Machinery.

[38] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[39] D. Shen, J.-D. Ruvini, and B. Sarwar. Large-scale item categorization for e-commerce. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 595–604, New York, NY, USA, 2012. Association for Computing Machinery.

[40] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. Artif. Intell.*, 2009:421425:1–421425:19, 2009.

[41] G. Suchacka, M. Skolimowska, and A. Potempa. A k-nearest neighbors method for classifying user sessions in e-commerce scenario. 2015:64–69, 01 2015.

[42] R. Yuan, Z. Li, X. Guan, and L. Xu. An svm-based machine learning method for accurate internet traffic classification. *Information Systems Frontiers*, 12:149–156, 04 2010.

[43] S. Zhang, H. Tong, J. Xu, and R. Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6, 11 2019.

[44] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434, 2018.