A TRANSFORMER-BASED GRAPH NEURAL NETWORK AGGREGATION FRAMEWORK FOR 5G RADIO LINK FAILURE PREDICTION

by

KAZI HASAN

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
August 2023

Table of Contents

List of Tables

List of Figures

Abstract

The prediction of Radio Link Failures (RLF) in Radio Access Networks (RANs) is crucial to ensure smooth communication and meet the demanding requirements of high data rates, low latency, and improved performance in 5G networks. However, weather conditions like precipitation, humidity, temperature, and wind have a significant impact on these communication links. Typically, RLF prediction uses a learning-based model to capture the relationships between historical radio link Key Performance Indicators (KPIs) and nearby weather station data. However, existing models often lack the capability to effectively encode context-aware time series sequences and fail to be generalized for unseen radio links. To address these issues, this thesis proposes a new RLF prediction framework that employs a state-of-the-art time series transformer model as a temporal feature extractor and incorporates a graph neural network (GNN) based dynamic aggregation method for surrounding weather stations' data to achieve better model generalization. The proposed aggregation method can be integrated into any existing prediction model to enhance its generalizability. The framework was evaluated in rural and urban deployment scenarios with 2.6 million KPI data points, demonstrating significantly higher F1 scores compared to previous methods (0.93 for rural and 0.79 for urban).

List of Abbreviations Used

RLF	Radio Link Failure
RS	Radio Site
WS	Weather Station
LSTM	Long Short-Term Memory networks
CV	Cross Validation
KPI	Key Performance Indicators
GNN	Graph Neural Network
GPS	Global Positioning System
SMOTE	Synthetic Minority Over-sampling Technique
RAN	Radio Access Network
ITU	International Telecommunication Union
RL	Radio Link
AR	Augmented Reality
VR	Virtual Reality
SLO	Service Level Objective
mmWave	millimeter wave
CNN	Convolutional Neural Network

Acknowledgements

I want to express my deepest thanks to my supervisor, Dr. Israat Haque, who guided me diligently throughout my Master's program. Her unwavering support and guidance fostered my development as a researcher. Her encouragement helped me overcome academic challenges, and I feel privileged to have had her as my mentor during my time at Dalhousie.

Subsequently, I would like to recognize the input of Dr. Thomas Trappenberg, professor of Computer Science at Dalhousie University. He has provided invaluable assistance during my research pursuits.

I extend my heartfelt appreciation to my dear Mother, Father, and cherished sister, for their indispensable presence in my journey. The immeasurable support and boundless love they have bestowed upon me form a debt I can never truly repay. The profound joy of knowing that my endeavours throughout this thesis and my pursuit of a Master's degree have made them proud is one of life's most gratifying rewards.

Chapter 1

Introduction

The rapid advancement of modern networking applications, such as Industry 4.0, smart transportation systems, health informatics, and augmented or virtual reality (AR or VR), has brought an increasing demand for high network bandwidth, high reliability, and fast communication speeds [18]. The mobile and wireless networks formed by these applications can be ad-hoc, mesh, sensor, or cellular networks [7,8,23, 30,32,33,36,37,48,51], where cellular can offer high speed and bandwidth along with a high reliability. Specifically, fifth-generation (5G) cellular networks have emerged with the goal of meeting diverse service level objectives (SLOs). To achieve this, 5G networks rely on millimeter-wave (mmWave) spectrums, which operate in the frequency range of 24GHz to 100GHz. These mmWave frequencies offer the capability to reliably transmit data over short distances, and they support a wide array of applications.

A fundamental component of the 5G infrastructure is the 5G radio access network (RAN), which plays a critical role in enabling seamless and eficient communication between devices and the core network. The key to the success of 5G RAN lies in the deployment of a dense array of base stations that communicate through mmWave radio links. These base stations, being strategically positioned, ensure comprehensive network coverage and eficient data transfer for the end-users. However, this dense deployment also introduces challenges, particularly concern is the impact of weather phenomena on the mmWave radio links. Weather phenomena, such as precipitation, humidity, temperature, and wind, can significantly affect the performance of mmWave radio links [62]. Distortions and attenuations arising from these weather conditions can cause signal degradation, leading to interruptions and compromised reliability in data transmission [2]. Consequently, maintaining a high level of robustness and effectiveness in the face of these challenges becomes a paramount concern for 5G network operators.

To ensure a seamless and uninterrupted user experience, 5G radio links must consistently adhere to stringent key performance indicators (KPIs) [3]. These KPIs serve as benchmarks that measure the network's eficiency, reliability, and overall performance. Parameters such as signal strength, data latency, signal-to-noise ratio, and connection stability are some of the critical metrics that are monitored and evaluated regularly. By continually monitoring these KPIs, network operators can proactively address any potential issues and optimize the performance of the 5G RAN.

In the domain of mobile networks, predicting link failures is of utmost importance for mobile operators, who invest significantly in proactive measures to maintain a robust and uninterrupted live network. To achieve this, researchers have explored various approaches that leverage historical radio link key performance indicators (KPIs) and weather station observations to forecast the probability of link failures in the upcoming days [3, 40, 50]. Aktas et al. [3] introduce a branched LSTM architecture that eficiently integrates time series and static data to improve predictive performance. Islam et al. [40] emphasize the significance of data preprocessing and demonstrate the potential of LSTM-autoencoder-based approaches. Meanwhile, Agarwal et al. [50] leverage decision trees and random forests to achieve accurate predictions. These research endeavours collectively contribute to the advancement of predictive analytics in the telecommunications industry and aid mobile operators in making informed decisions to ensure optimal network performance and customer satisfaction.

To meet optimal network performance, machine learning has been successfully applied for failure management in different parts of the 5G RAN (e.g., robust base station configuration) [59] in order to learn from data instead of having pre-programmed instructions. By introducing such automation, network operators have removed human action. This way large quantity of performance data can be exploited using learning based model and create solutions that scales to ever increasingly large 5G networks [14]. As such, researchers have designed the deployment of a failure prediction model in a 5G network data center from where it will monitor radio link KPI metrics of all base stations and reliably predict failure probability for coming days [69].

The ideal machine learning based failure prediction system does not only predicts

failures but also initiates proactive and reactive solutions to mitigate the effects of failure and maintain customer satisfaction [55]. The scope of such networks encompasses Self-Configuration, Self-Optimization, and Self-Healing [21]. One of the scopes involves networks to have self healing properties which allows the network to take actions in order to mitigate the effects of failures. The RLF prediction system enables this self-healing property by identifying potential link failures and allowing the network to automatically select another optimal route for trafic so that the overall delay in the network is minimized [14]. In order to build a RAN routing topology with more than one route between base stations, we can choose any state-of-the-art approaches like [31, 33, 34, 38].

Cutting-edge solutions have recognized the importance of sequence modelling and have utilized LSTM to process time series data for predicting link failures [3, 40]. However, these models face scalability issues in large deployments with numerous links and struggle to capture long-term dependencies due to vanishing gradients. Moreover, existing methods for associating radio links with weather stations rely on heuristics and lack generalizability for different network topologies. To address these challenges, an effective and scalable prediction approach must employ a reliable temporal feature extraction algorithm and a weather station aggregation method that can be applied to new links.

In this thesis, a new radio link failure prediction framework is introduced. The framework combines a time-series transformer architecture and a graph neural network-based node aggregation method to effectively handle temporal dependencies and weather station context. This approach leads to exceptional performance in real-world radio access network deployment data. The time-series transformer model assigns varying attention levels to each element in the input sequence to capture temporal dependencies. Treating surrounding weather stations as a graph structure allows the model to learn weather effects on a link through graph representation learning. The learned embeddings enable the model to generalize to new links by capturing neighbourhood topologies and feature dependencies. Additionally, the GNN-aggregation method further enhances the performance of existing prediction architectures for radio link failure.

1.1 Motivation

In the rapidly evolving landscape of mobile networks, ensuring the seamless and reliable operation of radio links has become a critical concern for mobile operators. Predicting link failures in advance can offer significant advantages, enabling operators to take proactive measures and make informed decisions to maintain a robust live network. In recent years, state-of-the-art solutions have leveraged the power of Long Short-Term Memory (LSTM) models for effective sequence modelling of time series data, which includes historical radio link Key Performance Indicators (KPIs) and weather station observations [[50], [3], [40]]. However, despite their success, these models exhibit certain limitations that hinder their scalability and overall performance in real-world deployment data.

One of the primary challenges faced by LSTM-based models is their sequential processing nature when dealing with time series data. This sequential processing can lead to ineficiencies, making them less suitable for large-scale deployments with a vast number of interconnected elements. As the number of elements grows, the sequential nature of LSTM causes a linear increase in computational time, which ultimately impacts the model's ability to process the data in a timely manner.

Moreover, the vanishing gradients issue remains a significant concern in LSTM-based architectures [83]. This issue occurs when the gradients used for training the model become extremely small, leading to dificulties in updating the network's parameters effectively. Consequently, long-term dependencies within the time series data might not be adequately captured by the model, which can lead to suboptimal performance and reduced prediction accuracy.

In addition to the vanishing gradients issue, LSTM models also lack the ability to weigh the importance of different elements within a sequence when making predictions [71]. This limitation can be attributed to the inherent structure of the LSTM architecture, where each element in the sequence is treated equally during the processing, without considering its relative significance in contributing to the overall prediction.

In addition to the challenges related to sequence modelling, correctly associating each link to the relevant weather stations is another critical aspect in accurately predicting link failures (RLF). The link-weather station association plays a pivotal

role in understanding the impact of weather conditions on the link performance.

Currently, the existing approaches resort to heuristics for establishing the link-weather station associations. These heuristics often lack the generality required to handle diverse Radio Access Network (RAN) topologies effectively. Consequently, when faced with a new deployment scenario, these models need to be retrained or modified, which can be time-consuming and resource-intensive.

One common heuristic used in these approaches involves associating each radio site with the closest weather station. This simple proximity-based association assumes that the closest weather station data will have the most significant impact on the link's performance [50]. While this may hold true in some cases, it oversimplifies the complex interplay between weather conditions and link failures, leading to suboptimal performance in certain scenarios.

Another heuristic employed is the aggregated k-nearest weather stations association. Here Aktas et al. [3], instead of relying on a single weather station, the model considers the influence of multiple nearby weather stations. The number of weather stations considered is controlled by the hyperparameter k. While this approach captures a broader view of the weather conditions surrounding the link, determining the optimal value of k can be challenging, and the model's performance heavily relies on this choice.

Alternatively, Islam et al. [40] calculate the optimal distance between the radio link and its surrounding weather sites for the association. The maximum of the minimum distances of radio link and weather station pairs is considered to be the optimal distance. Then, any weather station within this optimal distance from a radio link, is concatenated with the link features to create multiple link-weather station pairs with the same link feature values. While this tries to capture the combined effect of surrounding weather stations, it only considers one pair of link-weather stations at a time. This approach increase the problem complexity as the model needs to learn from different data points with same link but different weather stations and also gives the same weight to each associated weather stations. Not to mention, this optimal distance hyperparamter needs to be recomputed when the topology changes as it has great influence on model performance.

To overcome the limitations of these heuristics and to develop an effective RLF

prediction system, two key components must be addressed:

1. Robust Temporal Feature Extraction Algorithm: To capture long-term dependencies effectively, the RLF prediction model needs an effective temporal feature extraction algorithm. This algorithm should be capable of identifying patterns and trends within the time series data that may not be apparent in a LSTM based approach.

2. Generalizable Weather Station Aggregation Method: Instead of relying on fixed heuristics, the link-weather station association process should be flexible and adaptable to different RAN topologies. An ideal approach would be to utilize machine learning techniques that can learn the association from the data itself. By learning the associations from the data, the model can be generalized to unseen links and new deployments, reducing the need for retraining.

1.2 Research Objective

To develop a robust and generalized RLF prediction system, two key components need addressing. First, a temporal feature extraction algorithm is required to capture temporal dependencies effectively and identify patterns and trends in time series data. Second, a generalizable weather station aggregation method using machine learning techniques should be employed to adapt to different network topologies, learn associations from data, and reduce the need for retraining. By incorporating these components, the RLF prediction system can achieve higher accuracy and generalization, better addressing weather-induced link failures in modern communication networks. Thus, the objective of our research is to create a cutting-edge framework, which aims to revolutionize radio link failure (RLF) prediction in real-world Radio Access Net-work (RAN) deployments. By effectively incorporating temporal dependencies and weather station context, this novel approach promises to outperform existing RLF prediction architectures.

The core of the framework lies in its utilization of a time-series transformer architecture. Unlike traditional methods (e.g., an LSTM-based one) that struggle to capture long-term dependencies in sequential data, the time-series transformer model excels at encoding temporal dependencies. It achieves this by dynamically assigning attention weights to each element of the input sequence, allowing it to focus on the

most relevant information [81]. This capability ensures that the framework can effectively capture the complexities of real-world link KPI and weather station data, where temporal patterns and trends play a crucial role.

Moreover, weather conditions can significantly impact signal propagation and link performance, so taking them into account is crucial for accurate predictions. To achieve this, the framework treats the surrounding weather stations as a graph structure and adopts graph representation learning techniques [28]. By doing so, it effectively captures the weather effects on radio links and learns meaningful embeddings that encode topological structures and feature dependencies present within the neighbourhood.

Additionally, the framework leverages Graph Neural Network (GNN)-based node aggregation to enhance the predictive capabilities of existing architectures. The GNN method effectively aggregates information from neighbouring weather stations and incorporates it into the prediction process. This approach leads to improved performance compared to non-GNN aggregation RLF prediction architectures.

1.3 Contribution

In this research, we present a novel approach to address the challenge of link failure prediction in 5G Radio Access Networks (RANs) using a combination of transformer based time-series modeling and graph neural networks (GNNs). Our model eficiently processes variable number of weather station data for each radio link, capturing essential spatial and temporal dependencies from historical observations.

The first step of our approach involves leveraging a time-series transformer model to effectively model the temporal aspects of the radio link KPI and weather station data. This allows us to learn patterns and trends in the historical observations, aiding in accurate link failure prediction. Through a pooling layer, we create temporal representation vectors, which serve as condensed vectors of the time-series data.

To account for the spatial correlations between radio links and their surrounding weather stations, we apply graph neural network based aggregation. This aggregation step is crucial in capturing the intricate spatio-temporal relationships that influence link failures. The resulting latent vector captures the joint effect of the radio link and its surrounding weather stations, creating a spatio-temporal representation vector.

In parallel with the spatio-temporal processing, we handle the categorical features using a feed-forward network that operates separately from the transformer model branch. This approach allows us to incorporate non-numeric information, such as modulation type and surrounding station environment, into the prediction process. By combining these categorical representation vectors with the spatio-temporal ones, we obtain a holistic view of the factors influencing link failure events.

The integration of the spatio-temporal and categorical representation vectors is facilitated through concatenation, leading to a rich feature representation for each radio link. Subsequently, a feed-forward network is employed to predict the likelihood of link failure for the upcoming day.

The main contributions of our work are as follows:

- State-of-the-art Performance: Through the combination of time-series transformers and GNN-based aggregation, we achieve impressive results in 5G RAN link failure prediction, attaining an F1-score of 0.92. This performance outperforms existing approaches and demonstrates the effectiveness of our proposed framework.

- General Framework for RLF Prediction: To the best of our knowledge, our work is the first to introduce a generalized RLF prediction framework that can be seamlessly integrated with existing models. This flexibility allows researchers to extend their current models by incorporating our approach, leading to enhanced predictive capabilities.

- Robust Generalization Capability: We conducted rigorous experiments using real-world deployments with varying network topologies. The results show that our framework exhibits high generalization capability even when trained on a partial topology. This attribute is crucial in real-world scenarios, where networks often undergo dynamic changes.

- Enhancement of Existing Models: We validate the effectiveness of our approach by applying it to existing RLF schemes, such as LSTM+ and LSTM autoencoder. The incorporation of our framework leads to significant improvements in F1-score, showcasing its versatility and superiority.

- Open-source Implementation: In an effort to promote reproducibility and foster further research, we openly share the proposed framework (GNNTransformer) [1] prototype code. This allows others to not only replicate our experiments but also adapt and extend our proposed framework to suit their specific requirements.

1.4 Thesis Outline

The remainder of this thesis is organized structurally as follows: Chapter 2 is divided into two sections, the first of which, Section 2.1, lays the fundamental background required to better understand the work in this thesis. Next, Section 2.2 presents the important research works relevant to our work. Chapter 3 delves into the detailed methodology and design of the framework implemented. Here, the workflow and problem definition is also explored in Section 3.1 and Section 3.2. This is then followed by the detailed description of the dataset 3.3, comprehensive data preprocessing steps 3.4, and design details of model training 3.5. Following these, Section 3.6 presents the evaluation of the proposed framework, in terms of performance metrics and generalization capability, along with comparison with previous approaches. Chapter 4 discusses future research directions in Section 4.1 and concludes this thesis in Section 4.2.

Chapter 2

Background and Related Work

In this chapter, we begin by providing the essential context needed to understand the content presented in this thesis. Subsequently, we conduct a thorough examination of existing literature relevant to our research.

2.1 Background

2.1.1 5G RAN

The adoption of 5G technology has surged significantly owing to its ability to cater to emerging applications that demand high bandwidth, exceptional reliability, and ultra-low latency [25]. The deployment of 5G networks leverages millimeter waves (mmWaves) to achieve these remarkable capabilities, but this comes at the cost of reduced coverage area and increased penetration+ loss [80]. Nevertheless, to address these limitations, 5G deployment strategies have embraced the utilization of small cells, which act as intermediaries to collect and transfer user network trafic to the radio sites (RS). These radio sites are interconnected via 5G radio links (RL) that facilitate communication with each other and the core network, ultimately providing seamless internet connectivity to end-users [27]. These 5G radio links are also surrounded by a multitude of weather stations (WS) and the weather variations of these stations can significantly impact the performance and stability of the radio links [3].

Fig. 2.1 illustrates an overview of the described 5G deployment scheme, showcasing the strategic placement of small cells, radio sites, and weather stations in a coordinated manner. The relationship between these components require the 5G network to adapt to changing conditions and optimize its performance accordingly.

The central focus of this study is to utilize key performance indicators (KPIs) from the radio sites, coupled with the relevant weather station data, to develop a predictive model for forecasting radio link failures in advance, specifically for the

forthcoming day. By accurately predicting such failures, network providers can take preemptive measures and implement necessary precautions to safeguard critical services and maintain a seamless user experience.

The integration of weather station data into the prediction model holds significant promise. Weather conditions, such as heavy rainfall, extreme temperatures, or dense fog, can cause signal attenuation and degradation of radio link performance [2, 62]. Furthermore, factors like ice buildup on antennas or strong winds may lead to physical damage, disrupting the radio links' functionality. By incorporating weather-related variables into the prediction algorithm, the system gains the ability to discern patterns and correlations between weather patterns and radio link failures, thereby enhancing the accuracy of the forecasts. Note that once we predict a failure, we can deploy various measures of protection. For instance, one common is approach is constructing a reliable routing topology and redirect affected trafic over the alternative available routes, which is common both in wired [35, 53, 58, 65, 67, 68] and wireless [31, 34, 36, 37, 63] networks.

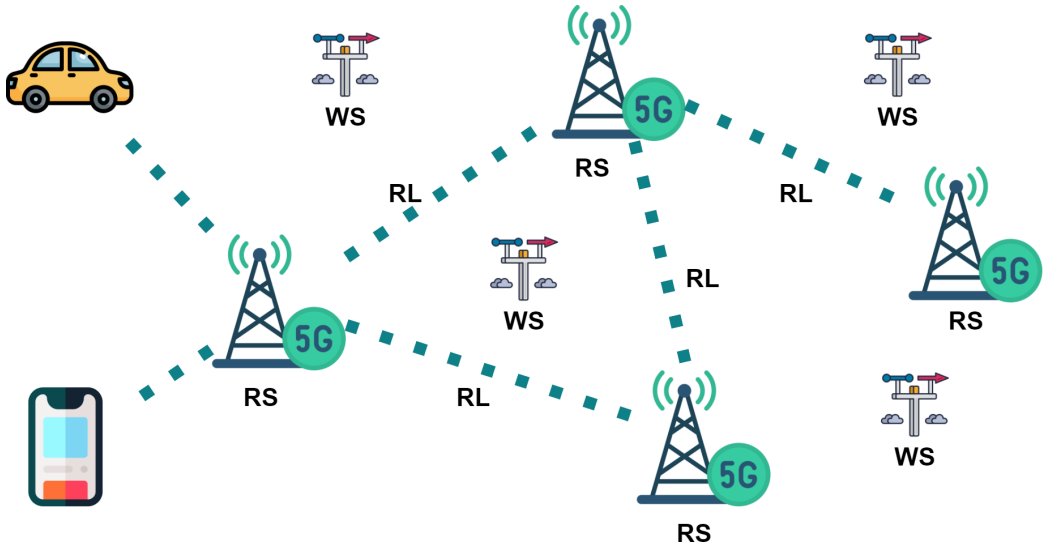


Figure 2.1: An example of 5G RAN deployment.

2.1.2 Imbalanced dataset

The presence of radio link failures constitutes a relatively minor portion when compared to normally operating links, resulting in imbalanced data that necessitates

careful processing. The dataset utilized in our research, the ITU dataset [3], showcases this phenomenon, with rural deployments showing a 0.3% failure rate and urban deployments experiencing a 0.06% failure rate [3]. Consequently, when employing deep learning models on such highly imbalanced datasets without addressing data imbalance issues, their performance tends to excel solely on the majority class [43].

Addressing the challenges posed by imbalanced data is a critical task, and researchers have proposed several typical approaches to tackle this issue. One such technique involves random undersampling of the majority class [3]. The underlying principle of this method is straightforward; it involves randomly removing instances from the majority class until a more balanced class distribution is achieved. While this approach may enhance the performance of models on minority class samples, it does come with a caveat. The random removal of majority class instances might inadvertently lead to the loss of informative data points [57]. Consequently, this might undermine the overall effectiveness of the model in handling the data distribution accurately.

On the other hand, another approach for dealing with imbalanced data is known as Synthetic Minority Over-sampling Technique (SMOTE) [40]. Unlike random undersampling, SMOTE aims to enhance the representation of the minority class by generating synthetic examples. The underlying principle of SMOTE involves selecting individual minority class samples and creating synthetic instances along line segments that connect them to their k nearest neighbours [15]. This process expands the minority class, providing the model with more examples to learn from. While SMOTE offers an effective way to handle imbalanced data, it is not without its limitations. One of the drawbacks of SMOTE is that it might generate noisy synthetic samples. Although SMOTE usually improves the model performance on minority class to some extent, it also has the risk of introducing noisy instances and overfitting problems because it doesn't consider the distribution of adjacent samples [42]. This can also adversely impact the model's generalization ability. These artificially created examples might not accurately represent the underlying distribution, leading to potential misclassifications during inference. Additionally, the selection of k nearest neighbours in the SMOTE process could introduce bias into the generated synthetic

samples [42]. In certain cases, sub-optimal neighbour selection might lead to an inaccurate expansion of the minority class, further affecting the overall performance of the model.

To tackle the limitations of SMOTE several augmentations have been proposed. One of the popular method is Borderline-SMOTE [29], which concentrates on enhancing class boundary information by targeting samples within a specific region. Another technique, RCSMOTE [66], manages the range of artificially generated instances. But SMOTE and it's variants do not consider the effects of spatially correlated data points. We cannot directly apply these techniques to synthesize link KPI data that considers the effect of surrounding weather stations. There has also been other approaches for synthetic data generation. For example, Alzantot proposed [4] a deep learning model with LSTM based generator and discriminator architecture to create synthetic sensor data. As this is an LSTM based model, it is unable to capture long term dependencies and focus on important elements in a time series.

To address the limitations of undersampling, oversampling and synthetic data generation techniques, we employ a weighted cross-entropy loss function, which provides an alternative and effective approach to handle imbalanced data [6,56,61]. This specialized loss function tackles the challenge of imbalanced datasets by incorporating prior probabilities as weights, making it a cost-sensitive approach [6]. The standard cross-entropy loss function is symmetrical and treats both classes equally, aiming to minimize the error for all classes at the same logarithmic rate. However, in the context of imbalanced data, this can lead to biased results, as the majority class tends to dominate the total loss during training, overshadowing the importance of the minority class [6]. The weighted cross-entropy loss function addresses this issue by introducing class-specific weights, which effectively balance the influence of each class on the overall error.

The incorporation of prior probabilities as weights is a crucial step in this approach. By utilizing prior probabilities, we assign higher weights to the minority class and lower weights to the majority class, effectively leveling the playing field during training [6]. As a result, the model pays more attention to the minority class, leading to improved performance and better generalization on imbalanced datasets.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} - y^{i} \log(\hat{y}^{i})(1 - \lambda) - (1 - y^{i}) \log(1 - \hat{y}^{i})\lambda \tag{2.1}$$

In Equation 2.1, $J(\theta)$ and m are the total loss and number of samples, respectively. The weight for actual failure event is represented as λ. y is the ground truth (y = 1 for failure) and $\hat{y}$ is the model prediction. This weighted crossy entropy loss function is only for binary predictions as we can only have failure or normal events. By utilizing the weighted cross-entropy loss function, we effectively mitigate the effects of class imbalance in the dataset without resorting to undersampling or oversampling. This approach allows the model to learn from the entire dataset, avoiding the loss of informative data points or the introduction of synthetic samples. As a result, the model becomes more adeptive at handling imbalanced data and is better equipped to make accurate predictions on real-world scenarios where class distributions are often skewed. Incorporating prior probabilities as weights into the cross-entropy loss function represents a significant step forward in addressing the challenges posed by imbalanced datasets and contributes to the development of more robust and effective RLF prediction models.

2.1.3 Attention

An attention mechanism can be defined as a process that takes a query along with a collection of key-value pairs and produces an output [71]. In this setup, all the components - the query, keys, values, and the output - are represented as vectors. The resulting output is determined by calculating a weighted sum of the values (Figure 2.2. Each value's weight is determined by a compatibility function that assesses the relationship between the query and the corresponding key.

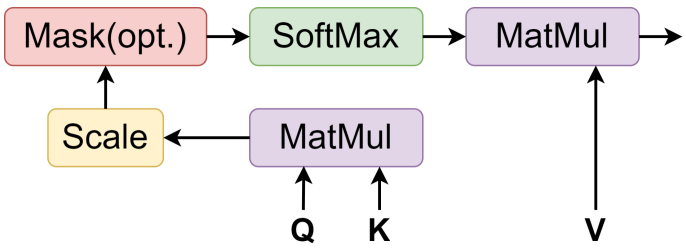


Figure 2.2: Scaled Dot-Product Attention.

The inputs to this mechanism include queries and keys, both having a dimension of dk, as well as values with a dimension of dv. The process involves computing the dot product between the query and each key, and then dividing the results by the square root of dk. The obtained values are then processed through a softmax function to obtain the weights assigned to the values. This entire attention mechanism is performed collectively on a set of queries, which are grouped together in a matrix Q. Similarly, the keys and values are arranged in matrices K and V, respectively. The ultimate matrix of outputs is calculated following [71] as:

$$\text{Attention}(Q, K, V) = \text{softmax}\frac{QK^T}{\sqrt{d_k}}V$$

2.1.4 Multi-head Attention

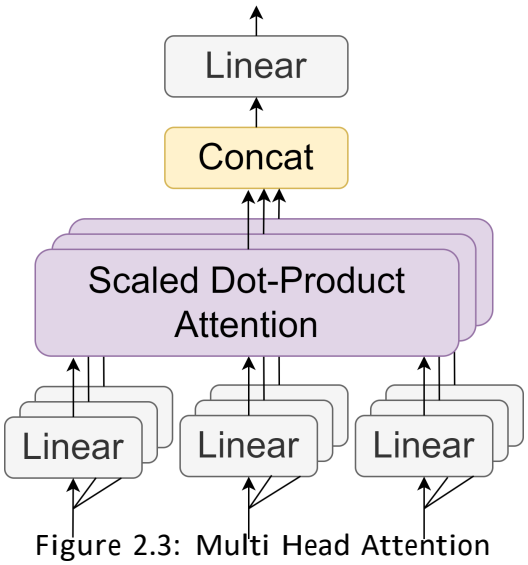


Figure 2.3: Multi Head Attention

Instead of using a single attention function with keys, values, and queries, it is advantageous to project the queries, keys, and values h times using distinct learned linear transformations to dimensions d_k, d_k, and d_v, respectively. The attention function is then applied concurrently to each of these transformed versions of queries, keys, and values, producing output values of dimension d_v. These outputs are combined after concatenation and subjected to another projection to obtain the ultimate

values, as illustrated in Figure 2.3. Multi-head attention enables the model to simultaneously focus on information from various representation sub-spaces at different locations.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O \tag{2.2}$$

where

$$\text{head}_i = \text{Attention}(QW_{Q_i}, KW_{K_i}, VW_{V_i}) \tag{2.3}$$

Where the projections are parameter matrices $W_Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_K \in \mathbb{R}^{d_{model} \times d_k}$, $W_V \in \mathbb{R}^{d_{model} \times d_v}$, and $W_O \in \mathbb{R}^{h \times d_v \times d_{model}}$.

2.1.5 1D Convolution

1D convolution is an operation used for feature extraction and transformation. It involves sliding a filter or kernel over an input to compute the output [47]. The input can be x, where $x = [x_0, x_1, x_2, \ldots, x_{N-1}]$ is a sequence of N elements. The kernel, also known as the filter, is another sequence of values that slide over the input signal. The kernel is h, where $h = [h_0, h_1, h_2, \ldots, h_{M-1}]$ is a sequence of M elements. The convolution operation at position i is calculated by multiplying corresponding elements of the input signal and the kernel and then summing up the results. The resulting output after performing the convolution operation is denoted as y. It is a sequence of values obtained by sliding the kernel over the input signal and calculating the convolution at each position. The operations can be expressed as:

$$(y * h)_i = \sum_{j=0}^{M-1} x_{i+j} \cdot h_j$$

where:

$(y * h)_i$: Output value at position i

x_{i+j} : Input value at position i + j

h_j : Kernel value at position j

M : Length of the kernel

$$y = [(y \circledast h)_0, (y \circledast h)_1, (y \circledast h)_2, \ldots, (y \circledast h)_{N+M-2}]$$

2.1.6 Time-series Transformer

The RLF prediction problem entails the utilization of historical radio link and weather station data as inputs, where learning-based models are employed to capture temporal dependencies. In recent years, there has been a surge in the popularity of transformer-based time series representation learning models [43]. These Transformer models are rooted in a multi-head attention mechanism [71], which imparts them with a spe-cial suitability for handling time-series data [74]. The crux of the matter lies in the self-attention module, which exhibits the remarkable capability of concurrently representing each element within the input sequence by encompassing dependencies with other elements in the same sequence [71]. Furthermore, the presence of multiple attention heads within the Transformer architecture enables it to account for diverse representation contexts [81]. In other words, different attention heads can aptly capture various types of relevance existing between input elements in the time series sequence. These varied types of relevance may correspond to multiple kinds of periodicities observed in the multivariate data.

The architecture of the time-series transformer module, as illustrated in Fig. 2.4, is composed of two essential sub-modules, each contributing distinct functionalities to the overall model. This design is tailored to effectively address the challenges posed by time-dependent data while capturing both local and global patterns.

In the first sub-module, the input time series sequence undergoes a series of operations to facilitate effective feature extraction and temporal representation learning. To begin with, batch normalization is applied across the feature dimension of the input sequence (Fig 2.4). This normalization process enhances the stability and convergence of the model during training, as it reduces internal covariate shift [39]. By normalizing the features, the model can focus on learning meaningful representations from the data without being hindered by the scale of individual features.

Subsequently, the key aspect of the first sub-module comes into play - the multi-head attention mechanism [71]. This mechanism enables the model to jointly attend to information from different representation sub-spaces at different positions in the input

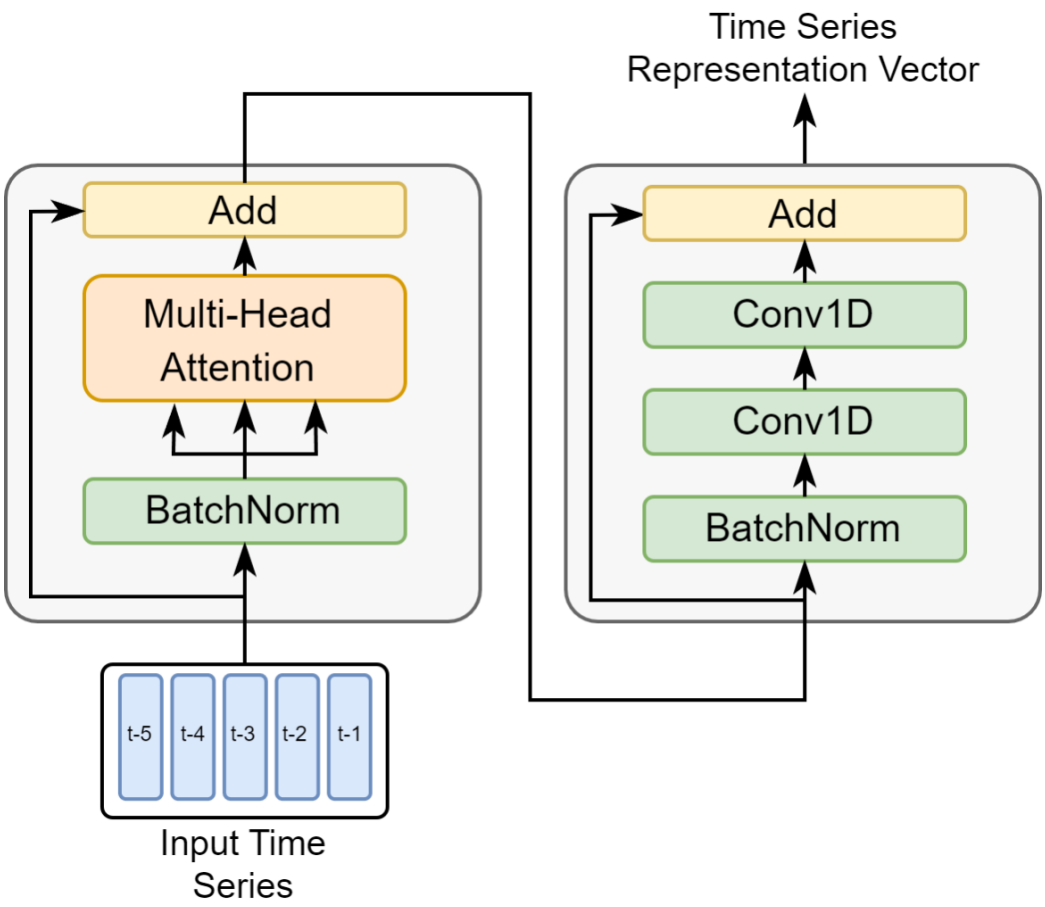


Figure 2.4: Transformer module for time-series representation learning.

sequence. By doing so, the model can effectively capture long-range dependencies and temporal patterns that may span across various segments of the time series. The multi-head attention mechanism has proven to be highly effective in sequence-to-sequence tasks, as it empowers the model to weigh the importance of each element's context with respect to others, thereby generating a robust temporal representation.

To ensure the seamless flow of information and mitigate the vanishing gradient problem, which is often encountered in deep neural networks, residual connections are established around each of the two sub-modules (Fig 2.4). These connections enable the preserved information from the input to be directly added to the output of each sub-module. This mechanism ensures that the gradients can propagate effectively through the network during backpropagation, allowing the model to learn and adapt to complex temporal dependencies in the data. The presence of residual connections in the transformer architecture is a distinguishing feature from traditional LSTM

(Long Short-Term Memory) networks, as LSTMs do not inherently have residual connections [82].

Moving on to the second sub-module, its primary role is to further extract local patterns and features from the time series representation generated by the first sub-module. This additional extraction process is performed to enrich the representation with fine-grained details and intricate temporal features. Within the second sub-module, there is a batch normalization layer, which serves the same purpose as in the first sub-module, promoting stability and faster convergence during training [11].

The feature extraction process is accomplished through the use of two 1D convolution layers with a ReLU activation function in between. These convolutional layers effectively act as filters, sliding across the time series representation to detect and emphasize local patterns and relevant features. The use of convolutional layers in this context is inspired by the success of convolutional neural networks (CNNs) in image-related tasks, where the filters can detect spatial patterns in two-dimensional data. Similarly, in the time-series domain, the 1D convolution layers act as effective tools for recognizing and capturing temporal patterns [52].

As a result of the processing within the second sub-module, the output is a set of compressed vector representations of the original time series sequence. These representations are highly enriched with both global temporal dependencies captured by the multi-head attention mechanism and local patterns extracted through the 1D convolution layers. The compressed vectors retain essential information from the original sequence while eficiently summarizing the data in a way that is conducive to classification and regression.

In summary, the architecture of the time-series transformer module is a well-thought-out design that harnesses the power of multi-head attention and 1D convolution to effectively handle time-dependent data. The combination of the two sub-modules, along with residual connections, ensures the seamless extraction of global and local temporal features, thus enabling the model to produce compact yet informative vector representations that can be readily employed in various downstream tasks.

2.1.7 Weather station aggregation

The propagation of 5G waves is affected by various challenges, including attenuation, interference, loss, and interruptions caused by adverse weather conditions such as rain, fog, snow, wind, and temperature variations [62], [2]. To make reliable RLF prediction, previous studies [3], [40], [50] have integrated information from surrounding weather stations of a radio link. They achieved this by utilizing derived features, optimal distance, and data from the nearest weather stations to enhance the performance of radio link failure prediction. Despite these efforts, relying on a fixed number of neighbouring weather stations proved to be ineffective since the optimal number may vary across different links, even within the same deployment scenario. To overcome this limitation, it becomes imperative to adopt a learning-based approach that dynamically selects the most relevant surrounding stations, thereby better capturing their impact on the radio link.

Another critical consideration is the weight assigned to each neighbouring weather station. Treating all stations equally could introduce bias into the prediction since their influence on the radio station should be proportional to various aspects, e.g., their distance from a link. Hence, it is essential to deploy deep learning techniques that can learn the weighted aggregation of these neighbouring weather stations to ensure an accurate prediction.

Moreover, it is important to recognize that 5G deployments can significantly vary, ranging from densely populated urban areas to sparser rural regions. Even within a single deployment, there may be patches of densely populated regions. Consequently, conventional algorithms that fail to extract all contextual information struggle to generalize effectively over new radio links. These traditional methods may rely on different sampling techniques, such as oversampling minority samples to enhance generalization [40], undersampling majority samples to introduce regularization [3], or employing ensemble models to avoid overfitting [50]. However, a more generalized and robust algorithm should possess exceptional representation learning capabilities to encode relevant context for each radio link and incorporate effective regularization properties to reduce overfitting.

In order to address the aforementioned challenges in predicting radio link failure in 5G wave propagation, we propose leveraging the power of graph neural networks

(GNNs) [16,28,76,78]. GNNs have shown great promise in various applications and are particularly well-suited for handling graph-structured data, such as the relationships between radio stations and their surrounding weather stations.

At the core of GNNs lies the concept of node embedding, which plays a crucial role in compressing the high-dimensional information of a node's neighbourhood into a lower-dimensional vector representation. This node embedding is then fed into neural networks for tasks such as classification, clustering, and prediction. By effectively capturing the relevant features and relationships from a variable number of neighbouring weather stations, GNNs enable us to create informative and compact node embeddings that can be used to enhance the prediction performance.

One popular GNN model that we will employ in our approach is GraphSAGE [28]. GraphSAGE is a versatile framework that excels at aggregating features from a local neighbourhood, allowing it to generalize effectively across different radio links and weather station configurations. By leveraging the power of GraphSAGE, we can aggregate information from variable number of surrounding weather stations for each radio link, thereby improving the accuracy of our prediction model. Formally, the graph aggregation step can be described by Equation 2.4.

$$\begin{aligned} \{e_{l'}, l' \in N(l)\} &= \sigma(W.\{z_{l'}, l' \in N(l)\}) \\ e_l &= \max(\{e_{l'}, l' \in N(l)\}) \end{aligned} \tag{2.4}$$

Here, $N(l)$ is the set of neighbouring nodes for l and $z_{l'}$ is the set of feature representations of these nodes. A transformation by weights W (can be any neural network) and non-linear function σ generates the set of learned feature vectors $\{e_{l'}, l' \in N(l)\}$ of the neighbouring nodes. Finally, a max operation over these vectors produces the aggregated embedding e_l for the node l.

The utilization of GNNs and, in particular, GraphSAGE, holds several advantages for our prediction task. Firstly, GNNs can naturally handle graphs with varying numbers of neighbours, making them ideal for scenarios where the number of surrounding weather stations varies across different radio links. This inherent flexibility ensures that our model can dynamically adjust its focus on the most relevant weather stations for each prediction, thereby avoiding the limitations of fixed or arbitrary station selection.

Secondly, GNNs excel at capturing complex relationships and dependencies within the graph. In our case, this means that the model can effectively learn and exploit the spatial and temporal patterns between radio stations and weather stations, thereby enhancing the prediction accuracy under diverse weather conditions.

Moreover, the node embeddings generated by GNNs offer a concise yet informative representation of each radio link's context. By compressing the high-dimensional neighbourhood information into low-dimensional vectors, we can significantly reduce the computational overhead while still retaining the essential information necessary for accurate predictions.

In summary, the adoption of graph neural networks, with a focus on the powerful GraphSAGE model, provides a compelling solution to the challenges faced in aggregating information from surrounding weather stations for each radio link. Leveraging the node embedding capabilities of GNNs helps creating informative and compact representations for each radio link. This, in turn, enhances our prediction model's ability to generalize across diverse deployment scenarios and weather conditions, making it a promising approach for achieving more reliable and accurate 5G communication networks.

2.2 Related Works

The popularity of deep learning-driven failure predictions is on the rise due to the ability of these models to effectively grasp the intricate spatio-temporal characteristics in 5G networks and handle the vast volumes of data generated. As a result of this finding, we introduce two sets of studies in this section: one focused on learning-based failure predictions in 5G, and the other exploring the utilization of graph neural network (GNN) aggregation for capturing spatial correlations.

2.2.1 Learning-based Failure Prediction

In their respective studies, Khunteta et al. [45] and Boutiba et al. [12] introduced the Long Short-Term Memory (LSTM) network in RLF prediction, which proved effective in capturing temporal feature correlations for predicting link failures. However, one limitation of their approach was the lack of consideration for weather effects, which can significantly impact the performance of radio links. Subsequent research endeavours sought to address this gap by incorporating historical radio link Key Performance Indicators (KPIs) in conjunction with weather observation data, similar to the dataset utilized in our present study.

For instance, Agarwal et al. [50] took a step further and combined individual link features with data from the closest weather station, employing the Random Forest classifier as their prediction model. This hybrid approach proved to be quite promising in capturing both the temporal dynamics of link failures and the potential influence of weather conditions. Meanwhile, Aktas et al. [3] devised a sophisticated branched architecture, which integrated LSTM and feed-forward networks to account for both temporal and categorical feature dependencies, respectively. Their work highlighted the significance of considering various types of dependencies in link failure prediction.

In another study, Islam et al. [40] harnessed the power of LSTM-autoencoder's reconstruction capabilities. They trained their model on normal link data and used high reconstruction errors during testing to identify potential link failures, achieving remarkable results in link failure prediction. Although these previous approaches made use of LSTM's ability to extract valuable information from temporal sequences, they still faced challenges in quantifying the importance of individual elements within

a time series and capturing all possible influences among time series variables [71]. On the other hand, Tunnell et al. [24] and Zhao et al. [83] both used statistical methods to show that LSTM cannot fully represent long memory effects in the input.

Recently, transformer models have emerged as a promising alternative for time series forecasting [74]. Unlike traditional LSTM-based approaches, transformers can effectively capture long-range dependencies and selectively weigh the importance of different elements in a time series sequence [81]. It is these distinct advantages that motivated us to explore the application of a time series transformer model in the context of link failure prediction. Building upon this foundation, we propose a novel branched architecture that leverages the strengths of the time series transformer model. One of the key innovations of our approach lies in the graph aggregation technique applied to each link's surrounding weather stations. By aggregating data from these stations, we aim to gain a more comprehensive understanding of the impact of weather conditions on individual links' performance. The graph aggregation pro-cess ensures that each link's weather-related influences are appropriately considered during the prediction task.

Our comprehensive experiments and rigorous evaluations demonstrate that our proposed approach outperforms previous works in the domain of link failure prediction. The incorporation of transformer models and the strategic use of graph aggregation enable our method to not only accurately capture temporal dependencies but also to effectively weigh the importance of different elements within a time series. Moreover, the ability to model long-range dependencies grants our approach a distinct advantage over traditional LSTM-based methods.

2.2.2 GNN Aggregation to Capture Spatial Correlations

Effectively capturing the spatial dependencies of surrounding weather station data plays a crucial role in predicting Radio Link Failure (RLF). This step is of paramount importance in the accurate estimation of RLF, which is vital for ensuring reliable and seamless communication in wireless networks. To achieve this objective, researchers have turned to Graph Neural Networks (GNNs) and their aggregation methods, which have demonstrated their effectiveness in various applications.

For instance, Wu et al. [75] utilized a GNN-based aggregation method to capture spatio-temporal relationships of weather radar data for precipitation forecasting, showcasing the potential of GNNs in handling complex weather patterns. Similarly, Fan et al. [20] applied GNN-based aggregation techniques to aggregate weather data for crop yield prediction, leading to improved accuracy in forecasting agricultural outcomes. Moreover, Gao et al. [22] made use of GNNs to encode weather parameters for solar radiation prediction, highlighting the versatility of GNNs in different environmental contexts.

In contrast, previous works on RLF prediction using weather station data adopted simpler heuristics in their data pre-processing steps to account for spatial relations of weather stations. For example, Agarwal et al. [50] opted to combine only the closest weather station features with each radio link, which overlooked potentially valuable information from more distant but still relevant weather stations. Likewise, another study by Aktas et al. [3] involved calculating derived features based on a fixed number (k) of nearest weather stations, neglecting the dynamic nature of radio links' relationships with their surrounding weather stations. Furthermore, Islam et al. [40] introduced an optimal distance threshold to associate all weather stations within that range with a particular radio link, overlooking the varying degrees of influence weather stations might have on different radio links.

To address the limitations of these prior approaches and harness the power of GNNs for RLF prediction, our proposed model leverages GNN-based aggregation methods. By employing GNNs, we can dynamically consider a variable number of relevant weather stations for each radio link, acknowledging the varying impact of different weather stations on different links. Additionally, we use a max function in the aggregation process, effectively selecting the most influential weather station features for each radio link. This allows our model to capture and emphasize the most significant spatial relationships, further improving its predictive capabilities.

An important advantage of our approach is the regularization effect introduced by the GNN aggregation. The GNN inherently learns to emphasize important connections and discard noisy or irrelevant signals, thereby enhancing the model's robustness and reducing overfitting tendencies. Consequently, our model demonstrates superior generalization performance compared to previous heuristic-based methods.

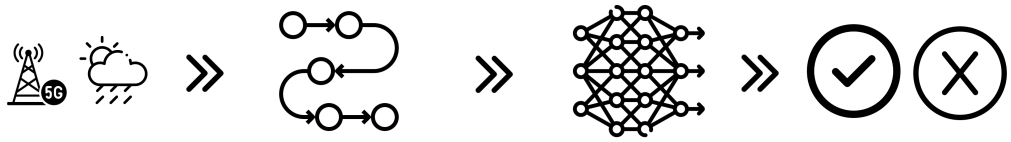
Our work showcases the potential of GNN-based aggregation methods in capturing spatial dependencies from surrounding weather station data for RLF prediction. By dynamically considering varying numbers of relevant weather stations and employing a max function, our model effectively addresses the limitations of previous approaches. Furthermore, the regularization properties of GNNs enhance the model's generalization capabilities, making it a promising solution for accurate RLF prediction in wireless communication networks.

Chapter 3

Design and Evaluation

3.1 Research Methodology

The chapter introduces the dataset and design details of transformer-based weather station aggregation framework for radio link failure (RLF) prediction, along with LSTM+ and LSTM Autoencoder architectures. The workflow is composed of four main components: dataset description, data preprocessing, model training and validation, and model testing. The entire RLF prediction workflow is visually represented in Figure 3.1.



Input	Preprocessing	Training	Testing
- Radio link - Weather observation	- Data Cleaning - Correlate Links and weather - Handle missing values - Encode categorical features - Perform time series split	- Train on first 70% data of two settings - Validate on the next 20% - LSTM+ - LSTM Autoencoder - Transformer framework	- Test on the last 10% of both deployment data

Figure 3.1: Link failure prediction workflow.

At first the dataset is introduced in great detail. Then the second stage, data preprocessing, involves preparing the data for the prediction models. It begins by cleaning the raw data, ensuring its quality and reliability. Handling missing values is essential to maintain data completeness and accuracy. The approach further employs encoding techniques for categorical features, transforming them into numerical representations for model compatibility. To understand the model's performance at different time of the year, a time series cross validation split is also performed.

In the third stage, the focus shifts to model training and validation. The framework evaluates two existing models, LSTM+ and LSTM-autoencoder, alongside the

proposed transformer-based model. This comparison allows for a comprehensive assessment of the transformer's potential advantages in RLF prediction. Rigorous validation ensures that the chosen models are well-performing and capable of generalization.

Lastly, the framework tests the selected models' performance on real-world, unseen link Key Performance Indicators (KPIs) and weather observations. This testing phase assesses the models' ability to make accurate predictions in practical scenarios, offering valuable insights into their real-world applicability.

3.2 Problem Statement

Mobile operators invest significantly in predicting link failures within live networks to proactively implement preventive measures. Researchers have explored various approaches to forecast link failure probability, including using historical radio link KPI and weather station data. Some propose specialized LSTM architectures to process both time series and static data in a single model, while others employ decision trees and random forest classifiers.

However, these state-of-the-art solutions face challenges. LSTM models do not consider all possible influences of each element in a time series, and they are incapable of putting different weights to different elements in a sequence. Additionally, associating each link with the relevant weather station remains a problem, often relying on heuristics, which limits the generalizability of these models to different network topologies.

An effective and scalable solution for predicting radio link failures must address these challenges by incorporating a robust temporal feature extraction algorithm and a weather station aggregation method that can be applied to new, unseen links.

3.3 Dataset Description

The dataset used in this study encompasses a collection of information related to radio link configuration and key performance indicators (KPIs) data, and time-aligned weather station observations from, two distinct deployments: urban and rural. The data spans a period, ranging from January 2019 to December 2020 for the urban

deployment and January 2019 to December 2019 for the rural deployment.

To ensure data privacy and confidentiality, certain configuration parameters and performance data of the radio links have been anonymized. This process involves removing sensitive information such as equipment names and link IDs without sacrificing the overall integrity and value of the data. Furthermore, the dataset does not disclose the exact Global Positioning System (GPS) locations of the radio stations. Instead, it provides pairwise relative distances between the stations, enabling researchers to maintain spatial relationships without revealing precise locations.

A comprehensive understanding of the dataset requires a thorough examination of the tables. First, we have the ”rl-sites” table, which contains identifiers for the radio sites. Each entry in this table includes site-specific parameters like height and clutter class, which describes the surrounding environment of the site (e.g., open urban, open land, dense tree area). It is important to note that a single radio site can have multiple radio links, as each site communicates with different sites through different links.

The ”rl-kpis” table presents daily KPIs for each radio link. Some of the essential KPIs include severe error seconds, error seconds, unavailable seconds, block bit error, etc. Moreover, this table also comprises link-specific configuration parameters such as card type, modulation, frequency band, and others. To uniquely identify each link, a pair of radio site ID and mini link ID is used as a composite key.

The ”met-stations” table contains data for unique weather stations, and each entry includes parameters like height and clutter class information. The clutter class values describe the surrounding environment of the weather station, including categories like dense tree areas, open land, airport, and more. These features provide insights into the spatial characteristics of the weather stations.

For a holistic analysis of weather-related trends, the ”met-real” table plays a crucial role. It offers hourly historical weather observations, such as temperature, humidity, precipitation, etc. These observations are then daily aggregated to align with the radio link KPI data. By capturing the temporal properties of the weather stations, researchers can investigate how weather conditions influence radio link performance.

Furthermore, the ”met-forecast” data provides valuable information about the upcoming five-day weather forecast for each weather station. This includes predictions

for weather phenomena like snow, rain, scattered clouds, as well as numerical values for humidity, temperature, wind speed, etc. Additionally, the maximum and minimum predictions for forecast features like temperature and humidity are also made available. This comprehensive weather forecast data allows researchers to study the potential impact of anticipated weather conditions on radio link performance.

Lastly, the ”distances” table contains pairwise relative distances between all radio sites and weather stations. These distances are expressed in specific units, which are not explicitly mentioned but may vary depending on the location and context of the study. By considering the distances between different radio sites and weather stations, researchers can incorporate spatial aspects into their analyses and potentially identify relationships between performance metrics and geographic proximity.

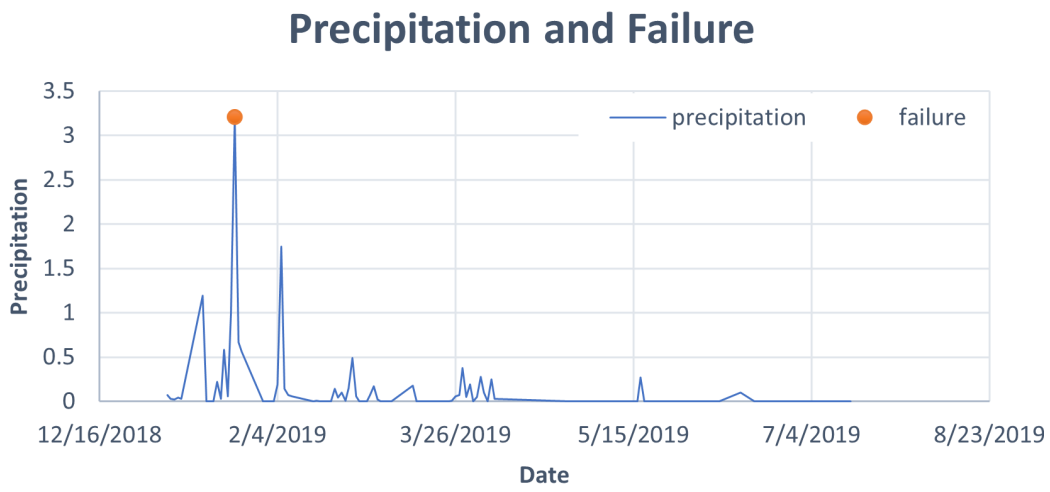


Figure 3.2: Effects of precipitation on failure.

To better understand the dataset and the relationship between weather effects and radio link failure, we plot the precipitation of weather station against one of the radio links that suffered failure (Fig. 3.2). It is worth noting that although the urban and rural deployments share similar features across the datasets, they differ in terms of the number of radio sites and weather stations included (Table 3.1).

Table 3.1: Summary of the Dataset.

	Urban	Rural
Number of radio sites	1674	1674
Number of weather stations	20	117
Number of time-series radio link KPI features	7	7
Number of time-series weather features	9	9
Total sample size	around 1.8 million	around 0.4 million
Total Number of features with missing values > 20%	3	5

3.4 Data Preprocessing

3.4.1 Data Preparation

The calibration of training data plays a crucial role in determining the effectiveness, precision, and complexity of machine learning tasks, as emphasized by Gupta et al. in their research [26]. In our own investigation, we encountered a significant challenge related to inconsistent values present in both weather station and radio link data. These inconsistencies, such as unexpected string values in the radio and weather data, have a detrimental impact on subsequent data transformations, hindering the casting of features into their appropriate data types.

To address these issues and ensure the integrity of our dataset, we adopted a multi-step approach. Our first priority was to handle the inconsistent values effectively. For instance, we employed a method of removing data samples that contained unexpected string values in numerical features. This step helped us eliminate potential errors and inconsistencies that could have otherwise affected the accuracy of our machine learning model.

With the data inconsistencies addressed, our next focus was on preparing the dataset for the machine learning process. To achieve this, we systematically cast all numerical features to the floating data type while converting categorical features to the string data type. This conversion process was crucial in ensuring that each feature was appropriately formatted and ready for the subsequent stages of our analysis.

Real weather data alignment. Our dataset has data from different entities

(e.g., weather stations and their observations, radio sites and their link performance data). In order to merge weather observations with radio link KPIs, their temporal frequencies need to be maintained. Radio site KPIs and real weather realizations are collected in the chosen dataset over daily and hourly time intervals, respectively. We use the standard mean aggregation [3] to transform hourly realizations into daily weather data to align historical weather realizations with radio link KPIs.

Data imputation. The majority of statistical and machine learning algorithms lack robustness in handling missing values, thereby being susceptible to the impact of incomplete data [41]. We calculate the percentage of missing values for each feature in our dataset. Some features from historical radio link KPIs and real weather station data have a high percentage of missing values. We use a simple heuristic of dropping features with missing values of 20% or higher. Also, some numerical features suffer from missing segments over time, but the data can be reliably interpolated if the percentage of missing values is under 15% [49]. Thus, we deploy time series linear interpolation to impute missing numerical KPIs and historical weather observations [60].

Data Merging. We need to use historical KPIs and weather data to predict following-day link failure. Thus, we append a label column in the KPIs table, representing the next-day link status. Also, each radio site can have multiple links, so we merge the KPI features with the corresponding site features by matching the site id. Weather station features are also merged with weather observation data similarly.

Tackling data imbalance. We use the weighted cross-entropy loss function to tackle the data imbalance, which incorporates prior probabilities into a cost-sensitive cross-entropy error function. Unlike traditional cross-entropy, this weighted approach accounts for the imbalanced nature of the data, giving a larger influence to the majority class while minimizing overall error. The loss function puts the prior minority to majority class ratio λ (0.003 for rural and 0.0006 for urban) into the regular cross entropy (Eq. 2.1). In rural deployment, this ensures that both classes have an equal influence because when $y = 0$ for a non-failure instance, the remaining term $(1 - y^i) \log(1 - \hat{y}^i)$ only contributes $\lambda = 0.3$ percent to the loss. Similarly, when $y = 1$ for a failure instance, the remaining term $-y^i \log(\hat{y}^i)$ contributes $(1 - \lambda) = 99.7$ percent to the loss.

Time series split.Our dataset contains time series data for radio link KPI metrics and weather station observations [3]. Time series prediction models are essential to understand historical context and predict future radio link failures (RLF) [54]. Researchers when they propose a new model, are interested to know whether the new method performs better than the state-of-the-art models. The standard procedure for non-time series regression and classification problems is to use cross validation as the model selection procedure. Cross-validation is a technique where a dataset is split into different folds to train and evaluate a model's performance multiple times, aiding in estimating its effectiveness on unseen data [10]. However, in time series prediction, the method for understanding model effectiveness can vary from problem to problem. This is because characteristics of the series such as the number of observed values, periodicity, or training complexity can be very different, as well as the types of forecasts needed (one-step-ahead, many-step-ahead, etc.) [13].

Cross-validation makes complete use of the available data for both training and validating [10]. But if the same method is used with time series data, the training and validation set will not be independent even if randomly chosen because time series might be generated by a process that evolves over time; affecting the fundamental assumptions of cross-validation that data is independent and identically distributed [5]. The problem of time dependency within training and validation can be solved by using blocks of data rather than choosing data randomly. Usually for time series, the end of each series is reserved and not used during model training. This kind of validation by taking a block from the end of series is called "last block validation" [70]. By simulating real-world application, last block validation overcomes the issues with traditional cross validation. Performing validation this way corresponds to real systems where continuous forecasting of upcoming values is needed.

However, there are variants of last block evaluation in terms of how to make different folds of training and validation sets. One evaluation technique is rolling-window evaluation [70], where the amount of training data is kept constant in each fold, by discarding old data from beginning of the train series. This is method can be effective if the model needs to be retrained in every window and has statistical advantages by providing model confidence on closely sized training data. Another similar evaluation technique is known as rolling-origin technique [70], which is probably the most

common use case for applications. Forecasts are performed by sequentially moving values from validation set to training set and changing the forecast origin accordingly. This produces folds with increasingly more train series. This method is also known as n-step-ahead evaluation, with n being the forecast horizon used during evaluation. This is appropriate for our use case because for real-world application model will be built once by experts and later the model will be used with updated data as new values become available. Also, if needed the model will be fine tuned with the combined dataset (existing and new values) to make future predictions. Given, the KPI and weather time series data for our real world use case, we choose rolling-origin evaluation technique.

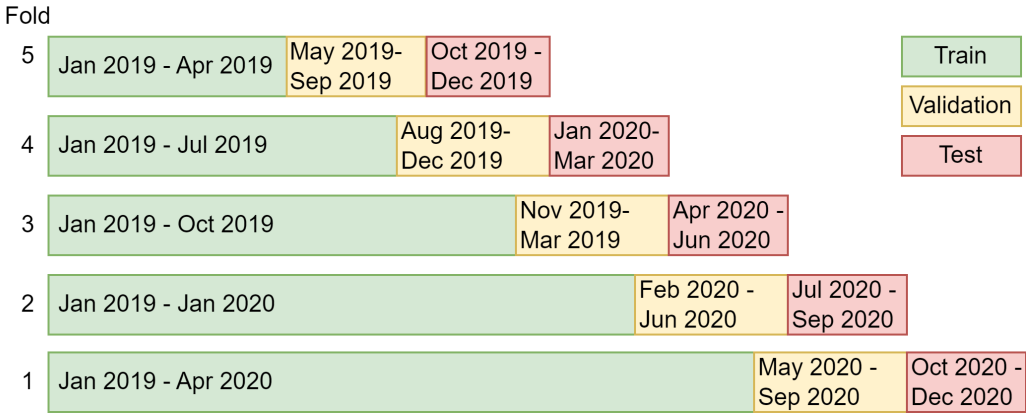


Figure 3.3: Time series folds for urban deployment.

We use rolling-origin evaluation technique and create the 5 folds by first sorting the data across time and splitting them into the first 70% train, the next 20% validation, and the last 10% test set to create the first fold with largest training set. For instance, in the urban deployment - data ranging from January 2019 to December 2020 - this results in train, validation, and test set containing January 2019 to April 2020, May 2020 to September 2020, and October 2020 to December 2020 data, respectively. The last 10% test data for this largest fold is considered the test block size. Subsequent folds are created by offsetting the splits by the number of samples in the test block. So the second fold would contain first 60% as train, next 70%-80% as validation, and the next 80%-90% as test. (Fig. 3.3). Similarly, we create the rest of the folds for both deployment scenario.

3.5 Model Training and Validation

This section begins by introducing our GNN aggregation approach. Subsequently, we elaborate on the model training process for our proposed transformer-based framework and also provide a comprehensive explanation of the existing LSTM+ and LSTM-autoencoder models along with their GNN aggregation augmented versions.

3.5.1 GNN Aggregation

The GNN aggregation process consists of two crucial components, as depicted in Algorithm 1. These components play a vital role in handling the variable number of weather station (WS) inputs and performing maximum aggregation of representation vectors. In the context of each mini-batch of m links, an essential step involves randomly selecting k closest weather stations (Line 2). The subsequent procedure entails iterating over the k closest weather stations for each radio link (RL) within the mini-batch (Line 5). During this iteration, the KPI feature vector is concatenated with the time-aligned weather station observation vectors, resulting in the generation of k WS + RL vectors for the selected radio link (Line 6).

To obtain context-aware representation vectors, these k WS + RL vectors are passed through the Transformer module (Line 7). This module plays a critical role in transforming the vectors, thereby enabling them to effectively capture the contextual dependencies. Following this transformation, global average pooling is employed to create k temporal embedding vectors, which are instrumental in capturing the time series dependencies for the radio link. An important step is the subsequent max aggregation across these vectors (Line 11). This process leads to the derivation of the final node embedding vector denoted as L^mNodeEmbd for the chosen radio link.

It is worth emphasizing that this process is carried out for all m links, resulting in the calculation of node embedding vectors for each of them. The iterative nature of the algorithm and the involvement of various steps make it highly eficient in handling variable inputs and ensuring effective aggregation of representation vectors.

Expanding on the method, the random selection of k closest weather stations serves a crucial purpose in mitigating potential biases and ensuring a representative set of inputs for each radio link. By incorporating randomness in the selection process,

Algorithm 1 Weather Station Aggregation

Input: Historical Radio Link KPIs of l links for t days, where each link $L = \{L^1, L^2, L^3, \ldots, L^l\}$, and $L \in R^{l \times t \times features}$; Historical weather station observations of n stations for t days, where $W = \{W^1, W^2, W^3, \ldots, W^n\}$, and $W \in R^{n \times t \times features}$; Transformer weight matrices T; Differentiable aggregator function max; M mini batches with each of size m.

Output: Node embeddings for all links in a mini batch

```
for minibatch ← 1 to M do
    k ← Random[1, n]
    for L^m ← 1 to m do
        EmbdList ← ∅
        for W^k ← 1 to k closest stations do
            WS + RL ← concat(L^m, W^k)
            ConReps ← T(WS + RL)
            TempEmbd ← AvgPooling(ConReps)
            EmbdList ← append TempEmbd
        end for
        L^m NodeEmbd ← max(EmbdList)
    end for
end for
```

the algorithm becomes robust and is better equipped to handle diverse scenarios. Additionally, the iterative concatenation of KPI feature vectors with time-aligned weather station observation vectors is a powerful technique that facilitates the fusion of different data sources. This fusion is particularly useful in the context of radio link performance, as it enables the algorithm to take into account the influence of various weather conditions on the link's behaviour.

The utilization of the Transformer module for generating context-aware representation vectors is an innovative approach. Transformers have gained immense popularity in natural language processing tasks for their ability to effectively model dependencies among sequence elements. In this context, applying them to capture dependencies between the WS + RL vectors allows the algorithm to effectively understand and utilize the temporal relationships in the data. As a result, the node embedding vectors derived from this process are rich in context and carry valuable information about the radio link's behaviour over time.

The global average pooling step, which generates temporal embedding vectors, is an essential technique for aggregating temporal information from the WS + RL vectors. This pooling operation helps to summarize the temporal aspects of the data while preserving its key characteristics. The subsequent max aggregation step is a strategic decision in generating the final node embedding vector. By choosing the maximum value from the temporal embedding vectors, the algorithm focuses on the most relevant and salient features, leading to a more compact yet informative representation of the radio link's behaviour.

3.5.2 Transformer GNN Architecture.

The primary objective of the transformer-based framework presented in this research is to convert time series sequences into probability vectors. The model utilizes preprocessed radio link Key Performance Indicators (KPIs) and weather station observations, following the mentioned steps, as inputs. Subsequently, it generates a probability vector predicting the occurrence of link failures on the subsequent day. By leveraging this approach, the study aims to enhance the accuracy of link failure prediction and contribute to the understanding of time series analysis in relation to radio link performance and weather data.

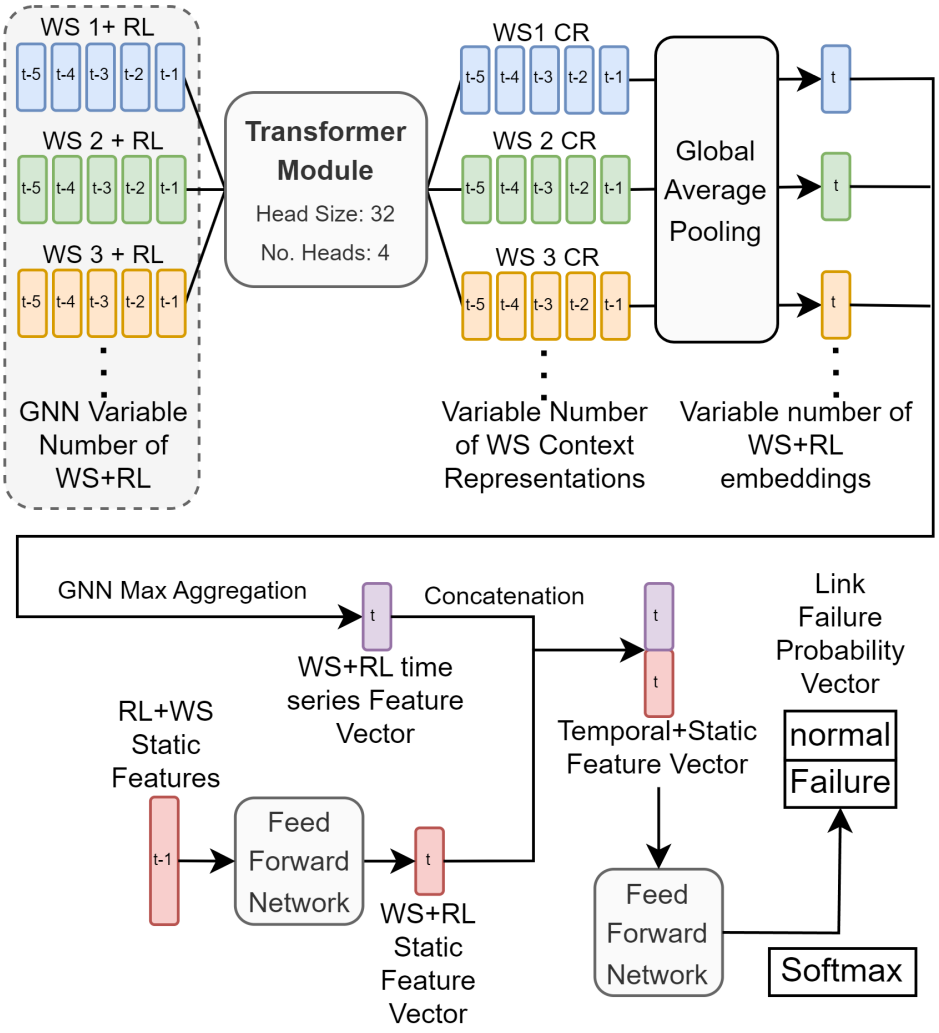


Figure 3.4: TransformerGNN architecture

This work presents a comprehensive approach for radio link failure prediction using a time series transformer module and a variable weather station aggregation method. The main goal is to predict the probability of link failure and no failure for the following day based on the input data of radio link and weather station time series.

The complete architecture of the proposed model is depicted in Figure 3.4. It consists of three main parts: the generalized transformer branch, the static feature branch, and the feed-forward output branch. The transformer module (illustrated in Figure 2.4) plays a crucial role in processing the time series data and capturing the contextual information of radio links and relevant weather stations.

The model's working mechanism involves feeding the radio link and weather station time series data as inputs to the transformer module, followed by global average pooling and max aggregation functions. This process results in an embedding vec-tor that encapsulates the contextual information of the radio link and its associated weather station. Additionally, a feed-forward network processes the one hot encoded static feature, representing configuration parameters of the link and weather station. The static feature branch outputs a latent representation that complements the context feature vector derived from the transformer module. These two feature vectors are concatenated and further passed to another feed-forward network responsible for generating the final output vector. The output vector contains two elements: one expressing the probability of link failure for the following day and the other expressing the probability of no failure.

By dividing the model into distinct branches, each with a specific purpose, the approach effectively combines both temporal and static information to enhance prediction accuracy. This holistic architecture demonstrates a well-rounded method for radio link failure prediction that considers the influence of dynamic and static factors in the prediction process.

Generalized transformer branch. The research introduces a generalized transformer branch that processes input data from radio links (RL) and weather stations (WS). The RL time series data consist of 9 features, such as severe error seconds, available time, and bbe, while the WS time series data include 7 features, such as temperature, humidity, and precipitation. All features are available daily, and to incorporate temporal information into the transformer, a time step is added as an extra feature as part of positional encoding. The incorporation of positional encoding enables the transformer to consider temporal patterns effectively and make accurate predictions.

The time series vectors for a radio link are denoted as L, where $L = L_1, L_2, L_3, \ldots, L_t$, and each L_i represents the data for a specific day. The dimension of L is $t \times 9$, where t is the total number of days. On the other hand, the weather station vectors are denoted as W, where $W = W^1, W^2, W^3, \ldots, W^n$, ordered by ascending distance from the radio link. Here, n represents the total number of weather stations in the deployment. Each weather station, W^i, contains its own time series data, denoted as

$W^i = W^i_1, W^i_2, W^i_3, ..., W^i_t$. The dimension of W is $n \times t \times 7$, capturing data from all weather stations across time.

The research focuses on predicting link failure probability on a daily basis using KPIs from the previous five days, as it yields the best prediction performance (Islam et al. [40]). The input data consists of concatenated feature vectors from radio links and weather stations, denoted as $WS1 + RL$ and $WS2 + RL$, where WS1 and WS2 represent the first and second closest weather station time series data, respectively. These vectors contain nine link features, seven weather features, and one time-step number column, resulting in an input tensor of shape $(\text{batchsize}, 3 \times 5 \times 17)$ for the transformer module (Fig. 3.4). Each weather station time series passes through the transformer module and the number of weather station time series can vary. This mimics the variable number of neighbouring nodes in a graph structure. So, we use GNN based variable number of neighboring node processing to capture dependencies for each time series using a transformer module.

The study identified that using a small batch size (e.g., 32 or 64) with extremely low minority-to-majority class ratios (0.003 for rural and 0.0006 for urban) caused unstable training. To address this, we adopted larger batch sizes of 1024 for rural deployment and 6000 for urban deployment. This decision aimed to ensure that each batch contains at least two link failure events on average, which stabilized the model training process.

The architecture depicted in Fig. 3.4 involves a transformer module that operates on the time series vectors $WS + RL$, producing embedding vectors that encompass the interactions between elements within the sequence. During the training phase, each batch comprises only the initial n $WS + RL(5, 17)$ tensors, where n ranges from 1 to 3. Consequently, a radio link may be associated with its n nearest weather stations at different iterations. This data augmentation technique enhances the model's ability to generalize effectively.

However, during the inference stage, the augmentation step is removed, and n is set to 3, allowing all surrounding weather station contexts to be provided for a given link. This configuration ensures comprehensive contextual information for accurate predictions in real-world scenarios.

The transformer module utilized in the study comprises four heads, each with a

size of 32, alongside two 1D convolution filters of sizes 32 and 17, respectively (see Fig. 2.4). Both the input and output shapes of the transformer remain consistent, being (batchsize, $3 \times 5 \times 17$). The objective is to capture temporal dependencies for each weather station and radio link pair. To achieve this, global average pooling is performed across the time dimension, generating an output of shape (batchsize, 5, 17) for each pair, where different colors represent different pairs.

In order to further condense the information, the max function is employed as an aggregator, performing an element-wise max operation across the embedding vectors. As a result, a single feature vector of shape (batchsize, 17) is generated. This final feature vector effectively encapsulates the effects of a variable number of closest weather stations on each radio link, thereby facilitating a concise representation of the data. By adopting this generalized transformer module approach, the study aims to capture and leverage the interactions between weather stations and radio links in a flexible and eficient manner, paving the way for improved performance.

Static and output branch. This work focuses on utilizing the generalized transformer module to handle time series radio links and weather station data. To complement this, a feed-forward network is employed to encode static radio links and weather station features. The radio link features (e.g., modulation type, frequency band) and weather station features (e.g., clutter class, weather day) are treated as categorical features, and one hot encoding is used to process them before passing to the feed-forward network, which comprises two layers with 32 and 17 neurons.

The outcome from the static branch and the generalized transformer branch is merged through concatenation, forming a representation vector of size (batchsize, 34). This representation vector effectively captures both temporal and static dependencies. The concatenated vector is then fed into another feed-forward network with 2 layers, consisting of 16 and 2 neurons, respectively. To obtain the final probability vector for link failure, a Softmax layer is employed.

During the model training process, a weighted categorical cross-entropy loss function is utilized, and the Adam optimizer with learning rate of 0.001 is employed. Subsequently, during inference, a binary prediction for each input is made by selecting the maximum probability score from the two calculated probabilities.

3.5.3 LSTM+

In this thesis report, we employ the LSTM+ method proposed by [3] as a basis for comparison with our proposed framework. To ensure consistency, we adopt the same data pre-processing steps as previously mentioned, with a single modification: we compute derived features (mean, minimum, maximum, standard deviation) based on the 7 weather station features for each radio link, utilizing data from its 3 nearest weather stations, following the methodology in [3].

The pre-processed data is then fed into the LSTM+ model, which incorporates separate branches to capture temporal and spatial features (depicted in Fig. 3.5). For this purpose, the model employs 4 LSTM layers to capture temporal dependencies between radio links and derived weather station features. In contrast, configuration parameters are encoded using one-hot encoding and processed by a feed-forward network, similar to the approach utilized in Transformer GNN model. Subsequently, the output vectors from both branches are concatenated and fed into another feed-forward network to obtain the final probability score vector.

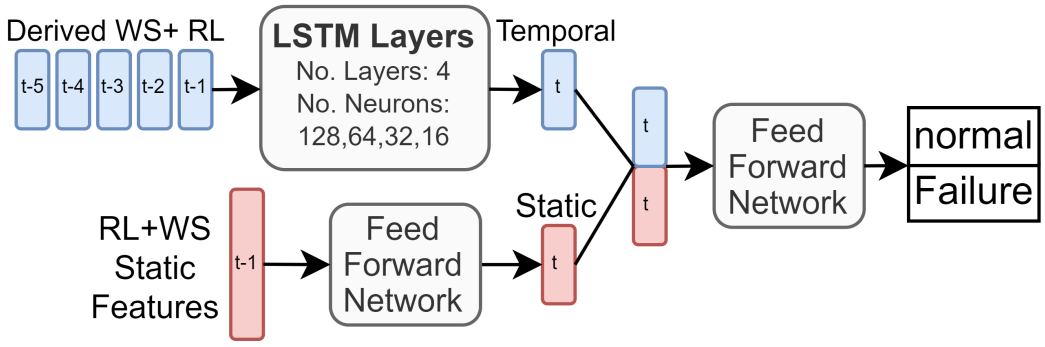


Figure 3.5: LSTM+ Architecture

In this study, we enhance the LSTM+ architecture by incorporating our generalized graph aggregation method to assess the performance boost achieved by our framework. To achieve this, we follow a data pre-processing procedure similar to that of our proposed framework. We introduce two key components of our aggregation method: the incorporation of a variable number of weather station inputs and the utilization of a max aggregation step (illustrated in Fig. 3.6).

During the batch processing, the LSTM layers analyze data from n weather stations, where n varies from 1 to 3, generating diverse feature representations that

effectively capture the temporal dependencies between radio link and weather station pairs. Subsequently, a max aggregation function is applied to merge the representation vectors. The static and output branches of this augmented model remain consistent with our proposed approach. Furthermore, to maintain uniformity and ensure fair comparisons, we employ the same optimizer and loss function for both LSTM+ and the augmented model experiments. This comprehensive evaluation allows us to gauge the eficacy of our generalized graph aggregation method in enhancing the LSTM+ model's overall performance.

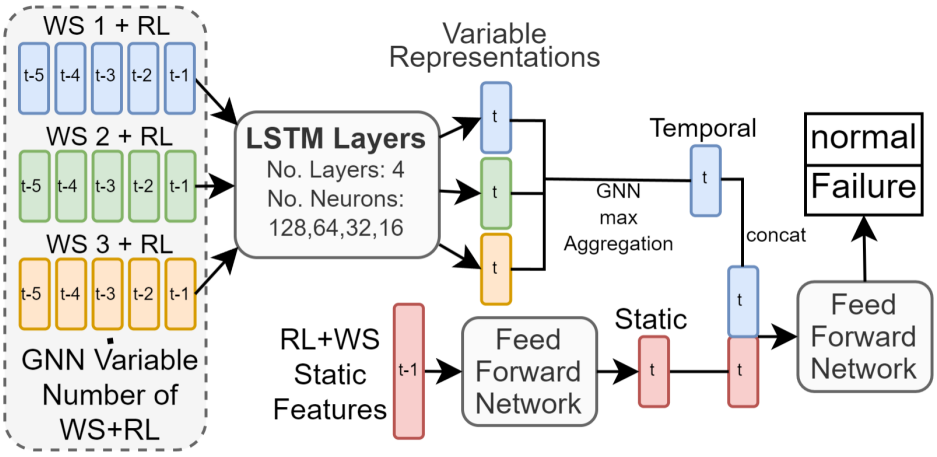


Figure 3.6: LSTM+ with proposed GNN aggregation.

3.5.4 LSTM-AutoEncoder

In this thesis report, we conduct a comparative study by implementing the LSTM-Autoencoder approach introduced in [40] and comparing its performance against GNNTransformer and other models. The data preprocessing steps remain consistent with those mentioned earlier. A crucial distinction between GNNTransformer and LSTM-Autoencoder lies in how we handle the "scalability score" feature, which is considered as numerical rather than categorical due to its floating-point values and high cardinality (1200). As such, an encoder-decoder LSTM model is utilized to transform normal input sequences into latent representations, which are then decoded back to output sequences closely resembling the original sequences (Fig. 3.7) [64].

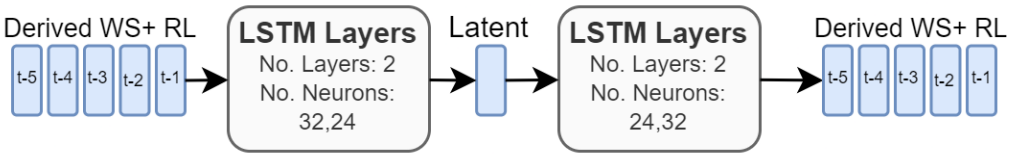


Figure 3.7: LSTM Autoencoder architecture.

It is important to note that only normal radio links are employed to train the encoder-decoder LSTM network, enabling the capture of feature dependencies in a normal scenario [40]. During the validation and testing phases, where both failure and normal link data are present, the trained model may not effectively decode failed input sequences back to their original forms, indicating a link failure.

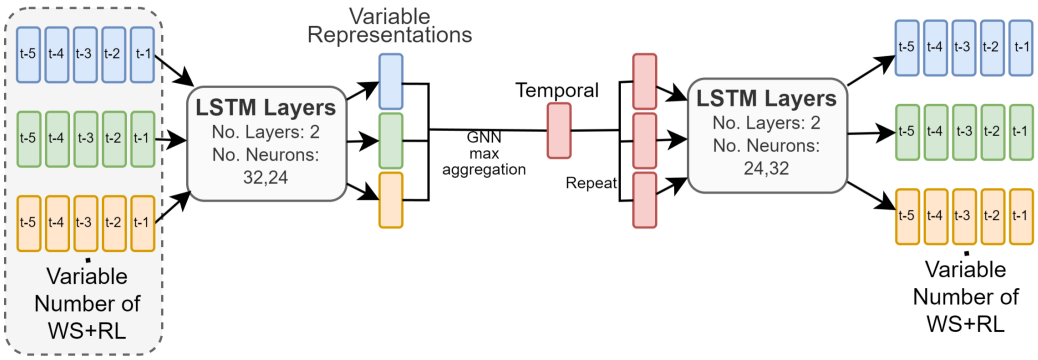


Figure 3.8: LSTM Autoencoder with proposed GNN aggregation..

To further enhance the LSTM-Autoencoder architecture, we integrate our generalized graph aggregation method to evaluate potential performance improvements. We incorporate the variable number of weather station input handling and max aggregation techniques (Fig. 3.8), similar to those utilized in GNNTransformer. This augmented network employs encoding of input sequences into a single latent vector, which is then replicated n times, where n represents the number of weather stations in the current batch. Subsequently, the replicated vectors pass through a decoder network, such as the LSTM Decoder, to generate output sequences closely resembling the input sequences.

By conducting a thorough analysis of the LSTM-Autoencoder approach and assessing its performance with the integration of the generalized graph aggregation method, this study aims to provide valuable insights into the effectiveness of the

techniques in addressing link failure detection and overall model robustness.

3.6 Evaluation and Results

This section of the thesis report we fist present the evaluation metrics and setup for our experiments. Then we showcase the outcomes of our transformer-based framework, in real-world settings encompassing both urban and rural deployments. Comparative analyses are conducted against LSTM+ and LSTM-autoencoder models, considering the incorporation of the proposed GNN aggregation step. Additionally, the assessment also focuses on the generalization ability of the framework, its potential beyond the specific deployment scenarios. The results shed light on the framework's eficacy and versatility.

3.6.1 Performance Metrics and Evaluation Setup

The evaluation process involves assessing various methodologies through precision, recall, and F1-score metrics. Each approach's performance is measured by calculating true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for both failure and non-failure events. True positives represent correctly predicted failures in the test dataset, while true negatives indicate accurately predicted non-failure events. False positives occur when non-failure events are mistakenly predicted as failures, and false negatives happen when failures are incorrectly predicted as non-failures.

To quantify the evaluation, we compute the metrics for both failure and non-failure classes using the following formulas: $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, and $F1score = \frac{2PrecisionRecall}{Precision+Recall}$. The reported results encompass the average precision, recall, and F1-score for both failure and non-failure scenarios, providing a comprehensive assessment of the methodologies' effectiveness in handling both types of events. The combined evaluation outcomes gives valuable insights into the overall performance of each approach and enable a thorough comparison of their capabilities.

The experiments were conducted on a machine equipped with an Intel(R) Xeon(R) Silver 4210R CPU running at 2.40GHz, 32 GB of memory, and an Nvidia Quadro RTX 8000 GPU boasting 50GB VRAM. The operating system and GPU versions employed were Ubuntu 20.04.6 LTS and CUDA 11.7, respectively. Data pre-processing

utilized numpy, pandas, and sklearn libraries, while the models were implemented using TensorFlow [1]. This robust hardware and software setup ensured eficient and accurate execution of the experiments, facilitating reliable results and analysis.

3.6.2 Performance comparison of different models

In this study, we trained and evaluated several models, including Transformer GNN (GNNTransformer), LSTM+ with GNN aggregation (GNNLSTM+), LSTM+, and LSTM-autoencoder with GNN aggregation (GNNLSTMAE), along with LSTM autoencoder. Here, we use rolling-origin evaluation method to compare the performance of different models. The models were trained on different training folds where each fold has training set of different size, and their performances were assessed on the 5-fold test data. This let's us understand how these models perform when the train-ing size is gradually increasing as new data is available similar to real world scenario. The evaluation focuses on predicting radio link failure for the following day. The results, presented in Table 3.2, include F1-scores along with corresponding precisions and recalls.

Notably, GNNTransformer emerged as the top performer, consistently surpassing all other existing approaches. For rural deployments, GNNTransformer achieved an average F1-score of 0.93, while for urban deployments, it attained an average F1-score of 0.79. These findings demonstrate the superior predictive capabilities of GNNTransformer and highlight its potential for accurate failure prediction in both rural and urban scenarios. The impressive and consistent performance of GNNTransformer showcases its effectiveness as a transformative framework in the field of radio link failure prediction. One limitation of the evaluation scheme is that we run each training fold only once. That is why we did perform any evaluation of the uncertainty (in terms of standard deviation of the achieved F1 scores) in our results. Ideally, we would want to run the experiments multiple time for each training fold and report the average and standard deviation of the F1 scores for each fold. We have further discussed the advantages and limitations of our evaluation scheme in the future work section.

The subpar performance of LSTM+ and LSTM Autoencoder can be attributed to their inability to assign varying weights to previous day data, lacking an internal

Table 3.2: The performance comparison of GNNTransformer for rural deployment.

Date Range	Approach	Precision	Recall	F1-Score
Nov-Dec 2019	GNNTransformer	0.9994	0.8600	0.9183
	GNNLSTM+	0.8456	0.8593	0.8523
	LSTM+	0.7049	0.7581	0.7049
	GNNLSTMAE	0.6860	0.6192	0.6452
	LSTMAE	0.6650	0.5793	0.6070
Oct-Nov 2019	GNNTransformer	0.9775	0.913	0.9431
	GNNLSTM+	0.8949	0.9275	0.9105
	LSTM+	0.7713	0.7973	0.7837
	GNNLSTMAE	0.5092	0.5336	0.5138
	LSTMAE	0.5241	0.5455	0.5314
Sep-Oct 2019	GNNTransformer	0.9622	0.9758	0.9689
	GNNLSTM+	0.8571	0.9999	0.9166
	LSTM+	0.8172	0.6687	0.7198
	GNNLSTMAE	0.6520	0.7128	0.6771
	LSTMAE	0.5790	0.5255	0.5377
Aug-Sep 2019	GNNTransformer	0.8571	0.9999	0.9166
	GNNLSTM+	0.8425	0.8466	0.8445
	LSTM+	0.5881	0.7993	0.6359
	GNNLSTMAE	0.5052	0.6909	0.5103
	LSTMAE	0.5033	0.5980	0.5032
Jul-Aug 2019	GNNTransformer	0.9020	0.9115	0.9067
	GNNLSTM+	0.7600	0.7875	0.7731
	LSTM+	0.6634	0.8451	0.7214

Table 3.3: The performance comparison of GNNTransformer for urban deployment.

Date Range	Approach	Precision	Recall	F1-Score
Oct-Dec 2020	GNNTransformer	0.8999	0.9799	0.9363
	GNNLSTM+	0.7544	0.9197	0.8168
	LSTM+	0.7247	0.8347	0.7688
Aug-Oct 2020	GNNTransformer	0.7383	0.8404	0.7803
	GNNLSTM+	0.6711	0.9061	0.7407
	LSTM+	0.7527	0.7361	0.7441
Jun-Aug 2020	GNNTransformer	0.6693	0.7378	0.6978
	GNNLSTM+	0.7270	0.6189	0.6560
	LSTM+	0.5414	0.5057	0.5100
Apr-Jun 2020	GNNTransformer	0.6734	0.8694	0.7360
	GNNLSTM+	0.6025	0.8476	0.6583
	LSTM+	0.5822	0.7824	0.6273
Feb-Apr 2020	GNNTransformer	0.8332	0.7856	0.8076
	GNNLSTM+	0.6599	0.6903	0.6738
	LSTM+	0.5940	0.6902	0.6257

mechanism to prioritize crucial information such as recent feature values or significant weather events. Additionally, the limited context window of LSTM, restricted to the previous context, results in the vanishing gradients issue, limiting its ability to capture long-spanning complex dependencies in sequences.

In contrast, the transformer architecture addresses these limitations through its self-attention mechanism, enabling it to focus on the most relevant elements within the input sequence. Furthermore, the transformer's larger context window allows for a more comprehensive understanding of the relationships between feature values that are distantly spaced in the sequence. These inherent advantages of transformer-based time series encoding contribute significantly to the superior performance of GNNTransformer.

By leveraging the transformer's self-attention and expanded context window capabilities, GNNTransformer excels in capturing intricate temporal patterns and crucial dependencies, surpassing the performance of LSTM-based approaches. The ability to effectively prioritize relevant information and capture long-range dependencies enhances GNNTransformer's predictive power and showcases the transformative potential of the transformer-based framework in advancing time series analysis for radio link failure prediction.

Figure 3.9 displays the distribution and variability of F1-scores for various approaches using a box and whiskers plot. The box in the plot represents the interquartile range (IQR), encompassing the middle 50% of the data. Notably, the GNNTransformer scores are more tightly clustered in the middle of the box, indicating less variability and a more consistent performance across different evaluations.

Conversely, the F1-scores from other approaches exhibit a broader spread, suggesting greater variability in their performance across the test data. The observed concentration of GNNTransformer scores within a narrower range implies a more robust and stable predictive ability compared to the other methods.

To understand if our proposed method is qualitatively and quantitatively significant compared to previous approaches we use box plot and perform One-way ANOVA test respectively. This box and whiskers plot visualization offers valuable insights into the distribution and spread of F1-scores, illustrating how the GNNTransformer outperforms other approaches by demonstrating higher consistency and reliability in its predictions. The plot's clear depiction of the data distribution allows for a quick and comprehensive comparison of different methods, supporting the conclusion of GNNTransformer's superiority in this evaluation. We also perform One-way ANOVA test to understand if the performance improvement of GNNTransformer is statistically significant compared to all other approaches. We also perform the same test to investigate the performance gain of GNN Aggregation. We achieve a p value of 0.03 and 0.003 for the GNNTransformer and GNN Aggregation test respectively; (a p value lower than 0.05 indicates that the results are significant) which shows the results produced are statistically significant.

3.6.3 Performance improvement from GNN Aggregation

The research demonstrates the superiority of GNNLSTM+ over the non-graph aggregation method (LSTM+) in both rural and urban scenarios. GNNLSTM+ achieves remarkable F1-scores of 0.85 and 0.70 in rural and urban deployments, respectively, surpassing the scores of 0.71 and 0.65 obtained by LSTM+. Similarly, GNNLSTMAE also exhibits enhancement, with its F1-score rising from 0.60 to 0.64 in rural deployment.

Comparing the LSTM-Autoencoder results with other models, a discrepancy is

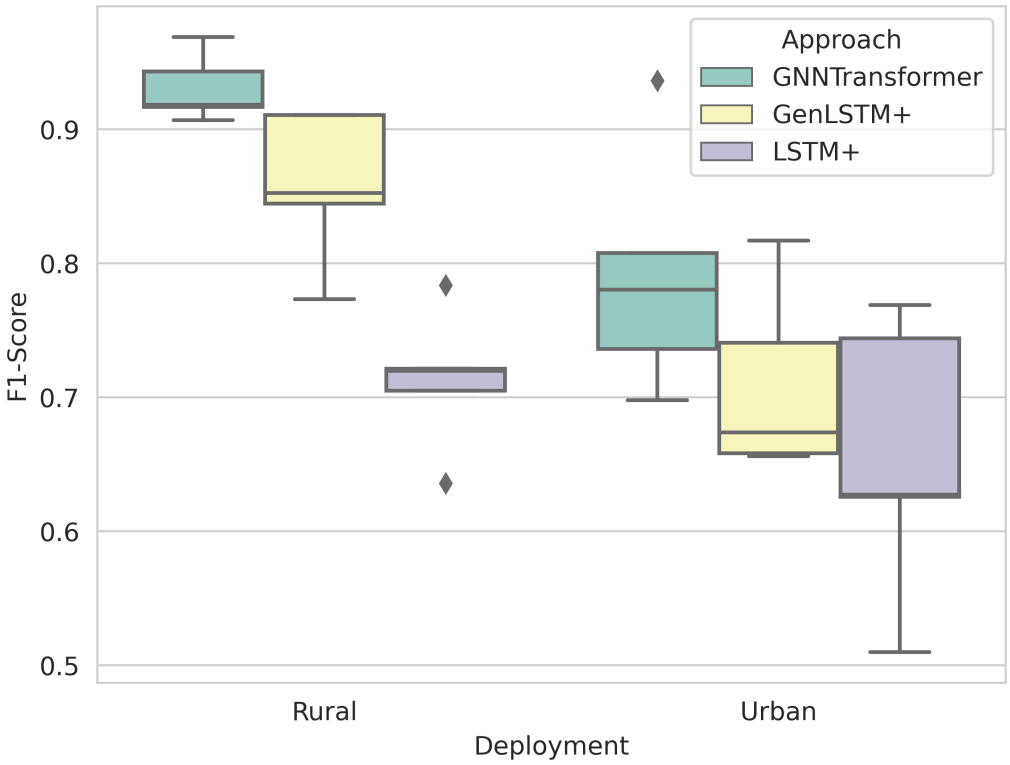


Figure 3.9: Distribution and variability of F1-scores of different approaches.

noticed. Our model's reported F1-score of 0.60 performs worse than that of the previous study [40]. We attribute this difference to our consideration of the scalability score as a numerical feature instead of a categorical one. Furthermore, the dissimilarity in the test dataset could be a contributing factor, as the previous work utilized failure events from approximately 6 months, whereas our test data covered only 2 months. We also observed the model struggled to fit with the complete dataset and so we used random undersampling to train the model. Due to scalability limitations, the rural deployment result for LSTM-Autoencoder is reported, as the model cannot handle larger urban deployments. Nonetheless, the introduction of GNN aggregation remains a potential avenue for enhancing model performance.

The LSTM+ and LSTM-Autoencoder models utilized k nearest weather stations and determined an optimal distance based on a radio link. The reason for their comparatively lower performance lies in their reliance on heuristic-based weather station association methods, which do not allow for dynamic learning from all nearby weather

Table 3.4: Generalization comparison of GNNTransformer and LSTM+ for rural deployment.

Training link fraction	GNNTransformer			LSTM+		
	Precision	Recall	F1score	Precision	Recall	F1score
0.5	0.7157	0.7587	0.7352	0.6840	0.6987	0.6910
0.4	0.8857	0.8396	0.8612	0.6024	0.7365	0.6423
0.3	0.6480	0.7775	0.6927	0.6006	0.7561	0.6436
0.2	0.6068	0.6578	0.6272	0.5655	0.6951	0.5969
0.1	0.5867	0.6178	0.5998	0.5846	0.5981	0.5908

stations. Both the k nearest and optimal distance approaches require constant tuning when topology changes occur, making them susceptible to outliers. Additionally, they treat all associated weather stations equally, disregarding the potential influence of closer stations. We believe that this static nature of weather station association contributes to the observed decline in performance.

In contrast, our proposed approach introduces a GNN aggregation step that enhances existing architectures like LSTM+ and LSTM-Autoencoder by facilitating dynamic learning of relevant weather stations for each link. By leveraging GNN aggregation, the models become capable of generalizing effectively to previously unseen links. This adaptability enables them to focus on the most informative weather stations, leading to improved performance compared to the conventional methods. The ability to dynamically update the station association based on context and proximity enhances the overall effectiveness of the models in predicting radio link performance.

3.6.4 Generalization comparison of GNNTransformer

The primary focus of the study is to assess the generalization capability of GNNTransformer in comparison to LSTM+. To achieve this, both networks are trained on a subset of radio links and then evaluated on the complete deployment, allowing for an evaluation of their ability to learn from a smaller topology and apply that knowledge to a larger one. The fractions of links taken from the rural topology range from 0.1 to 0.5. The results indicate that GNNTransformer consistently outperforms LSTM+ across all fractions, demonstrating an average F1-score of 0.70 and 0.87 for rural (Table 3.4) and urban (Table 3.5) deployment respectively, in comparison with 0.63 and 0.75.

Table 3.5: Generalization comparison of GNNTransformer and LSTM+ for urban deployment.

Training link fraction	GNNTransformer			LSTM+		
	Precision	Recall	F1score	Precision	Recall	F1score
0.5	0.9103	0.9349	0.9222	0.7121	0.8647	0.7681
0.4	0.9051	0.9699	0.9351	0.7104	0.8996	0.7756
0.3	0.9036	0.9399	0.9210	0.7227	0.8897	0.7834
0.2	0.8345	0.9248	0.8743	0.6486	0.9543	0.7237
0.1	0.8827	0.6799	0.7447	0.6687	0.7493	0.7013

For rural scenario, the improvement appears to be more pronounced for larger fractions (0.3, 0.4, 0.5) with an average increase from 0.65 to 0.76, while smaller fractions (0.1, 0.2) exhibit a comparatively smaller improvement, increasing from 0.59 to 0.61 on average. This suggests that GNNTransformer exhibits stronger generalization capabilities, particularly when with a larger fraction of the topology. Similarly, for urban scenario, we observe a similar and consistent improvement over LSTM+, exhibiting a greater improvement for larger fractions. One key distinction with the rural setting, is the decrease in performance is relatively lower; e.g. from 0.5 to 0.2 fractions GNNTransformer F1-score dropped from 0.92 to 0.87 while in the rural it dropped from 0.73 to 0.62. This is most likely due to the denser deployment of ra-dio links for an urban setting, as relatively higher number of links share the same surrounding weather stations.

In previous approaches, the absence of a dedicated architecture component for generalizing to unseen links and effectively utilizing each link's data was evident. The LSTM+ method, for instance, calculates derived features from k nearest weather stations for individual links, limiting its consideration to each link only once.

However, in contrast, our proposed GNNTransformer incorporates a variable weather station input feature within the GNN aggregation method. This allows for data augmentation, enabling the model to consider different numbers of weather stations for the same link during training. This inherent data augmentation technique in GNNTransformer is credited with the observed improvement in generalization performance. By leveraging diverse weather station combinations during training, GNNTransformer can effectively learn from varying contexts and conditions, ultimately enhancing its ability to handle previously unseen links.

Chapter 4

Conclusion and Future Work

4.1 Future Work

In this section we talk about, the limitations of this work from design and evaluation perspective and also discuss how those limitations should be taken care of in future studies.

Comprehensive evaluation. To understand differences in model performance, we utilized rolling-origin [70] evaluation method to compare our proposed model with existing architectures. This evaluation approach has its advantages and disadvantages. The primary advantage is that it matches real world application scenario where model is updated after certain period by fine tuning on new data. So, we are able to evaluate these models as if they were deployed in cellular network data centers. On the other hand, statistically the evaluation technique does not give good confidence as the training set size differs from one fold to the other. It leads to unfair comparison across different data splits because the fold with more data is likely to perform better. So, we are not able to get an average across these folds and provide a statistically sound technique to compare with previous methods. These limitations can be addressed by performing rolling-window evaluation [70] where the training set in each fold is of the same size because we discard samples from the beginning as we offset the window forward in time. Rolling-window evaluation provides the statistical advantages to measure the performance of different models and provide error deviation to understand the reliability of these models.

To have high confidence, it is a common standard to run the model training, validation and testing experiment multiple times for the same dataset and report the average and standard deviation scores. In our evaluation method, we do not perform multiple runs for the same fold of train, validation and test data. For each fold we evaluate the model once. As each fold has different sized training set, this leads to low confidence on the results obtained. So, it also does not provide any information

on the deviation of F1 score. Ideally in each fold, we must run the model for multiple times with different initialization to get the average and standard deviation of F1 scores for each fold. This will provide high confidence in the obtained results.

Other potential models. We considered LSTM+ and LSTM Autoencoder as benchmarks to compare against because these models were directly applied to the same ITU dataset used in this thesis. There are several other competing time series modeling techniques such as Gated Recurrent Unit (GRU) and Ordinary Differential Equations (ODE). Similar to LSTM, GRU can employ gates to select which information should be kept and which should be discarded [17] and it has been successfully used in different domains [79]. It has some benefits over LSTM [77] such as using smaller number of parameters and computational cost. On the other hand, Neu-ral Ordinary Differential Equation (NODE) is able to achieve reasonable result even when data is intermittently sampled [46]. A simpler model may work better for our RLF prediction problem and so GRU and NODE can be a possible avenue for fu-ture exploration. There are also variants of these networks such as GRU and LSTM with attention. Future work can consider comparing other state-of-the-art time se-ries modeling techniques to perform a comprehensive analysis of existing time series modeling techniques for RLF prediction problem.

Extending existing architecture. We show how utilizing a time series Transformer and GNN aggregation can lead to better performance and generalization on unseen links. This work can be further extended by incorporating recent advancements in pretrained transformer and attention GNN because, unsupervised pretraining of transformer has proven to increase performance in time series forecasting [81] and graph attention has improved neighbourhood aggregation in capturing spatial correlations [72]. We can also utilize the GNN aggregation to capture not only weather station effects but also inter-base station effects such as interference. We envision the same architecture principle can be applied to perform purely unsupervised approaches where the input consists of a variable number of weather stations for each radio link.

Synthetic data generation and model explainability. Our datasets have an extremely low minority-to-majority class ratio. Because of that, a small volume of data (minority class) penalizes the model performance. The failure cases present in the dataset may not capture all possible and important cases. We believe a reliable

and truthful generation of synthetic failure cases, with the help of simulations [44] or deep generative models [73], will improve the data quality and thus increase the faithfulness of deep learning models. Another important line of work from the perspective of making these models more faithful, is exploring interpretable and explainable machine learning models. Interpretable machine learning models [19] aim to have such transparency so that a human can understand why the model makes a certain prediction. But these models might not be actually explainable from model's internal workings. Explainable machine learning models [9] are able to point towards internal mechanism that resulted in certain prediction. Using such models will lead to a clear insight into the decision making process of these models and so it would be easier to get them deployed in live networks where network operators can rely on these models.

4.2 Conclusion

The study focuses on addressing the challenges in predicting radio link failures (RLF) in 5G RAN caused by weather changes. A proactive RLF prediction system is crucial for enhancing user experience and optimizing network operator resources. To achieve this, we investigate existing link failure prediction models' limitations and propose a novel time-series transformer-based framework. This framework incorporates GNN aggregation, considering a variable number of surrounding weather stations for each link, resulting in improved prediction accuracy.

The experiments conducted on real-world deployments demonstrate the effectiveness of the proposed framework in accurately predicting next-day RLF. Moreover, the study highlights the potential of applying the GNN aggregation to existing models, enhancing their prediction performance as well. As part of future work, the researchers plan to explore synthetic data generation methods and develop interpretable deep-learning models to further improve RLF prediction capabilities. By addressing these aspects, the research contributes valuable insights and solutions to the vital area of RLF prediction in 5G RANs.

Bibliography

[1] A transformer-based graph neural network aggregation framework for 5g radio link failure prediction. https://anonymous.4open.science/r/2C0F/README.md.

[2] Esmail MM Abuhdima, Ahmed El Qaouaq, Shakendra Alston, Kirk Ambrose, Gurcan Comert, Jian Liu, Chunheng Zhao, Chin-Tser Huang, and Pierluigi Pisu. Impact of weather conditions on 5g communication channel under connected vehicles framework. arXiv preprint arXiv:2111.09418, 2021.

[3] Semih Aktaş, Hande Alemdar, and Salih Ergüt. Towards 5g and beyond radio link diagnosis: Radio link failure prediction by using historical weather, link parameters. Computers and Electrical Engineering, 99:107742, 2022.

[4] Moustafa Alzantot, Supriyo Chakraborty, and Mani Srivastava. Sensegen: A deep learning architecture for synthetic sensor data generation. In 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pages 188–193. IEEE, 2017.

[5] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. 2010.

[6] Yuri Sousa Aurelio, Gustavo Matheus de Almeida, Cristiano Leite de Castro, and Antonio Padua Braga. Learning from imbalanced data sets with weighted cross-entropy function. Neural Processing Letters, 50(2):1937–1949, 2019.

[7] Chloe Bae, Shiwen Yang, Michael Baddeley, Atis Elsts, and Israat Haque. Bluetisch: A multi-phy simulation of low-power 6tisch iot networks. In GLOBECOM 2022-2022 IEEE Global Communications Conference, pages 4280–4285, 2022.

[8] Tahajjat Begum, Israat Haque, and Vlado Keselj. Deep learning models for gesture-controlled drone operation. In 2020 16th International Conference on Network and Service Management (CNSM). IEEE, 2020.

[9] Vaishak Belle and Ioannis Papantonis. Principles and practice of explainable machine learning. Frontiers in big Data, page 39, 2021.

[10] Daniel Berrar et al. Cross-validation., 2019.

[11] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. Advances in neural information processing systems, 31, 2018.

[12] Karim Boutiba, Miloud Bagaa, and Adlen Ksentini. Radio link failure prediction in 5g networks. In 2021 IEEE Global Communications Conference (GLOBECOM), pages 1–6. IEEE, 2021.

[13] Vitor Cerqueira, Luis Torgo, and Igor Mozetič. Evaluating time series forecasting models: An empirical study on performance estimation methods. Machine Learning, 109:1997–2028, 2020.

[14] Ursula Challita, Henrik Ryden, and Hugo Tullberg. When machine learning meets wireless cellular networks: Deployment, challenges, and applications. IEEE Communications Magazine, 58(6):12–18, 2020.

[15] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res., 16(1):321–357, jun 2002.

[16] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. Advances in Neural Information Processing Systems, 33:13260–13271, 2020.

[17] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), pages 1597–1600. IEEE, 2017.

[18] Aaron Yi Ding and Marijn Janssen. Opportunities for applications using 5g networks: Requirements, challenges, and outlook. In Proceedings of the Seventh International Conference on Telecommunications and Remote Sensing, pages 27–34, 2018.

[19] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. Communications of the ACM, 63(1):68–77, 2019.

[20] Joshua Fan, Junwen Bai, Zhiyun Li, Ariel Ortiz-Bobea, and Carla P Gomes. A gnn-rnn approach for harnessing geospatial and temporal information: application to crop yield prediction. In Proceedings of the AAAI conference on artificial intelligence, volume 36, pages 11873–11881, 2022.

[21] Hasna Fourati, Rihab Maaloul, Lamia Chaari, and Mohamed Jmaiel. Comprehensive survey on self-organizing cellular network approaches applied to 5g networks. Computer Networks, 199:108435, 2021.

[22] Yuan Gao, Shohei Miyata, and Yasunori Akashi. Interpretable deep learning models for hourly solar radiation prediction based on graph neural network and attention. Applied Energy, 321:119288, 2022.

[23] Hossein Ghannadrezaii, Jean-François Bousquet, and Israat Haque. Cross-layer design for software-defined underwater acoustic networking. In IEEE OCEANS. IEEE, 2019.

[24] Alexander Greaves-Tunnell and Zaid Harchaoui. A statistical investigation of long memory in language and music. In International Conference on Machine Learning, pages 2394–2403. PMLR, 2019.

[25] Leonardo Guevara and Fernando Auat Cheein. The role of 5g technologies: Challenges in smart cities and intelligent transportation systems. Sustainability, 12(16), 2020.

[26] Nitin Gupta, Shashank Mujumdar, Hima Patel, Satoshi Masuda, Naveen Panwar, Sambaran Bandyopadhyay, Sameep Mehta, Shanmukha Guttula, Shazia Afzal, Ruhi Sharma Mittal, et al. Data quality for machine learning tasks. In Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pages 4040–4041, 2021.

[27] Mohammad Asif Habibi, Meysam Nasimi, Bin Han, and Hans D. Schotten. A comprehensive survey of ran architectures toward 5g mobile communication system. IEEE Access, 7:70371–70421, 2019.

[28] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[29] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing, pages 878–887. Springer, 2005.

[30] I. Haque and N. Abu-Ghazaleh. Wireless software defined networking: a survey and taxonomy. IEEE Communications Surveys and Tutorials, 18(4):2713–2737, May 2016.

[31] I Haque and C Assi. OLEAR: Optimal localized energy aware routing in mobile ad hoc networks. In Proceedings of the 2005 IEEE International Conference on Communications, ICC '05, 2005.

[32] I. Haque, I. Nikolaidis, and P. Gburzynski. On the benefits of nondeterminism in location-based forwarding. In IEEE International Conference on Communications (ICC), 2009.

[33] Israat Haque, Chadi Assi, and William Atwood. Randomized energy-aware routing algorithms in mobile ad hoc networks. In Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, MSWiM '05, 2005.

[34] Israat Haque, Saiful Islam, and Janelle Harms. On selecting a reliable topol-ogy in wireless sensor networks. In Proceedings of the 2015 IEEE International Conference on Communications, ICC '15, 2015.

[35] Israat Haque and M. A. Moyeen. Revive: A reliable software defined data plane failure recovery scheme. In Stefano Salsano, Roberto Riggio, Toufik Ahmed, Taghrid Samak, and Carlos Raniery Paula dos Santos, editors, 14th International Conference on Network and Service Management, CNSM 2018, Rome, Italy, November 5-9, 2018, pages 268–274. IEEE Computer Society, 2018.

[36] Israat Haque, Mohammed Nurujjaman, Janelle Harms, and Nael Abu-ghazaleh. SDSense: An agile and flexible SDN-based framework for wireless sensor networks. The IEEE Transactions on Vehicular Technology, 68(2):1866 – 1876, February 2019.

[37] Israat Haque and Dipon Saha. SoftIoT: A resource-aware sdn/nfv-based iot network. The Elsevier Journal of Network and Computer Applications, 193, Nov 2021.

[38] Israat T Haque and Chadi Assi. Localized energy eficient routing in mobile ad hoc networks. The Willey Journal of Wireless and Mobile Computing, 7(6):781–793, August 2007.

[39] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pages 448–456. pmlr, 2015.

[40] Mohammad Ariful Islam, Hisham Siddique, Wenbin Zhang, and Israat Haque. A deep neural network-based communication failure prediction scheme in 5g ran. IEEE Transactions on Network and Service Management, pages 1–1, 2022.

[41] Anil Jadhav, Dhanya Pramod, and Krishnan Ramanathan. Comparison of performance of data imputation methods for numeric dataset. Applied Artificial Intelligence, 33(10):913–933, 2019.

[42] Zhenhao Jiang, Tingting Pan, Chao Zhang, and Jie Yang. A new oversampling method based on the classification contribution degree. Symmetry, 13(2), 2021.

[43] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. Journal of Big Data, 6(1):27, 2019.

[44] Theofanis Karamplias, Sotirios T Spantideas, Anastasios E Giannopoulos, Panagiotis Gkonis, Nikolaos Kapsalis, and Panagiotis Trakadas. Towards closed-loop automation in 5g open ran: Coupling an open-source simulator with xapps. In 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), pages 232–237. IEEE, 2022.

[45] Shubham Khunteta and Ashok Kumar Reddy Chavva. Deep learning based link failure mitigation. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 806–811. IEEE, 2017.

[46] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. Advances in Neural Information Processing Systems, 33:6696–6707, 2020.

[47] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. Mechanical systems and signal processing, 151:107398, 2021.

[48] Vinay Kolar, Israat T. Haque, Vikram P. Munishwar, and Nael B. Abu-Ghazaleh. Ctcv: Coordinated transport of correlated videos in smart camera networks. In 24th International Conference on Network Protocols (ICNP). IEEE, 2016.

[49] David M Kreindler and Charles J Lumsden. The effects of the irregular sample and missing data in time series analysis. In Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data, pages 149–172. CRC Press, 2016.

[50] Shyamal Krishna Agarwal, Somesh Banerjee, and Rajarshi Mahapatra. Prediction and recovery of radio link failure caused by environmental factors. In 2022 IEEE 19th India Council International Conference (INDICON), pages 1–6, 2022.

[51] M. Kulkarni, M. Baddeley, and I. Haque. Embedded vs. external controllers in software-defined iot networks. In 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), 2021.

[52] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10):1995, 1995.

[53] Udaya Lekhala and Israat Haque. Piqos: A programmable and intelligent qos framework. In IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2019, Paris, France, April 29 - May 2, 2019, pages 234–239. IEEE, 2019.

[54] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. Philosophical Transactions of the Royal Society A, 379(2194):20200209, 2021.

[55] Yi-Wei Mal, Jiann-Liang Chen, and Hao-Kai Lin. Mobility robustness optimization based on radio link failure prediction. In 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), pages 454–457. IEEE, 2018.

[56] Ibomoiye Domor Mienye and Yanxia Sun. Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. Informatics in Medicine Unlocked, 25:100690, 2021.

[57] Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In 2020 11th International Conference on Information and Communication Systems (ICICS), pages 243–248, 2020.

[58] M. A. Moyeen, Fangye Tang, Dipon Saha, and Israat Haque. SD-FAST: A packet rerouting architecture in SDN. In 15th International Conference on Network and Service Management, CNSM 2019, Halifax, NS, Canada, October 21-25, 2019, pages 1–7. IEEE, 2019.

[59] David Mulvey, Chuan Heng Foh, Muhammad Ali Imran, and Rahim Tafazolli. Cell fault management using machine learning techniques. IEEE Access, 7:124514–124539, 2019.

[60] Irfan Pratama, Adhistya Erna Permanasari, Igi Ardiyanto, and Rini Indrayani. A review of missing values handling methods on time-series data. In 2016 international conference on information technology systems and innovation (ICITSI), pages 1–6. IEEE, 2016.

[61] Mohammad Reza Rezaei-Dastjerdehei, Amirmohammad Mijani, and Emad Fatemizadeh. Addressing imbalance in multi-label classification using weighted cross entropy loss function. In 2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME), pages 333–338, 2020.

[62] Hawraa J Saadoon, Thamer M Jamel, and Hasan F Khazal. An overview for the effects of different weather conditions on 5g millimeter waves propagations. CPGR, pages 19–20, 2021.

[63] Dipon Saha, Meysam Shojaee, Michael Baddeley, and Israat Haque. An Energy-Aware SDN/NFV architecture for the internet of things. In IFIP Networking 2020 Conference (IFIP Networking 2020), Paris, France, June 2020.

[64] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. Network anomaly detection using lstm based autoencoder. In Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, pages 37–45, 2020.

[65] Meysam Shojaee, Miguel C. Neves, and Israat Haque. Safeguard: Congestion and memory-aware failure recovery in SD-WAN. In 16th International Conference on Network and Service Management, CNSM 2020, Izmir, Turkey, November 2-6, 2020, pages 1–7. IEEE, 2020.

[66] Paria Soltanzadeh and Mahdi Hashemzadeh. Rcsmote: Range-controlled synthetic minority over-sampling technique for handling the class imbalance problem. Information Sciences, 542:92–111, 2021.

[67] Fangye Tang and Israat Haque. Remon: A resilient flow monitoring framework. In Network Trafic Measurement and Analysis Conference, TMA 2019, Paris, France, June 19-21, 2019, pages 137–144. IEEE, 2019.

[68] Fangye Tang, Meysam Shojaee, and Israat Haque. Ace: an accurate and cost-effective measurement system in sdn, 2021.

[69] Antonio Tarrias, Sergio Fortes, and Raquel Barco. Failure management in 5g ran: Challenges and open research lines. IEEE Network, 2023.

[70] Leonard J Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. International journal of forecasting, 16(4):437–450, 2000.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[72] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. arXiv preprint arXiv:1710.10903, 2017.

[73] Kun Wang and Chenxu Sheng. Application of gan in 5g technology. In Journal of Physics: Conference Series, volume 1699, page 012004. IOP Publishing, 2020.

[74] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. arXiv preprint arXiv:2202.07125, 2022.

[75] Yajing Wu, Xuebing Yang, Yongqiang Tang, Chenyang Zhang, Guoping Zhang, and Wensheng Zhang. Inductive spatiotemporal graph convolutional networks for short-term quantitative precipitation forecasting. IEEE Transactions on Geoscience and Remote Sensing, 60:1–18, 2022.

[76] Kejie Xu, Hong Huang, Peifang Deng, and Yuan Li. Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing. IEEE Transactions on Neural Networks and Learning Systems, 33(10):5751–5765, 2021.

[77] Peter T Yamak, Li Yujian, and Pius K Gadosey. A comparison between arima, lstm, and gru for time series forecasting. In Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence, pages 49–55, 2019.

[78] Zhen Yang, Ming Ding, Bin Xu, Hongxia Yang, and Jie Tang. Stam: A spatiotemporal aggregation method for graph neural network-based recommendation. In Proceedings of the ACM Web Conference 2022, pages 3217–3228, 2022.

[79] Z Zainuddin, P Akhir EA, and MH Hasan. Predicting machine failure using recurrent neural network-gated recurrent unit (rnn-gru) through time series data. Bulletin of Electrical Engineering and Informatics, 10(2):870–878, 2021.

[80] Abdelbasset Bedda Zekri, Riadh Ajgou, Ali Chemsa, and Said Ghendir. Analysis of outdoor to indoor penetration loss for mmwave channels. In 2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP), pages 74–79, 2020.

[81] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pages 2114–2124, 2021.

[82] Albert Zeyer, Parnia Bahar, Kazuki Irie, Ralf Schlüter, and Hermann Ney. A comparison of transformer and lstm encoder decoder models for asr. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pages 8–15. IEEE, 2019.

[83] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. Do rnn and lstm have long memory? In International Conference on Machine Learning, pages 11365–11375. PMLR, 2020.