Time Series Forecasting for Rural Fixed-Wireless Communication Network Monitoring

by

Alexander Gordon Bruce Colpitts

B.Sc.E. (EE), University of New Brunswick, 2018
M.Sc.E. (EE), University of New Brunswick, 2020

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

In the Graduate Academic Unit of Electrical and Computer Engineering

Supervisor:	Brent R. Petersen, Ph.D.,
	Electrical and Computer Engineering
Examining Board:	Erik J. Scheme, Ph.D.,
	Electrical and Computer Engineering
	Julian Meng, Ph.D.,
	Electrical and Computer Engineering
	Kenneth B. Kent, Ph.D.,
	Faculty of Computer Science
External Examiner:	Halim Yanikomeroglu, Ph.D.,
	Department of Systems and Computer Engineering,
	Carleton University

This dissertation is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK
December, 2023

Abstract

LTE and 5G cellular networks are evolving at a rapid pace to accommodate more users and higher traffic. Existing studies have largely focused on urban mobile networks, leaving their rural fixed-wireless counterparts largely ignored. This work investigates the performance of a rural Canadian fixed wireless network on several time scales. Short- and long-term performance properties are considered. It is well known that rural propagation environments behave differently from urban ones. Long-term temporal changes in the propagation environment, such as foliage and snow, were shown to have a small impact on the performance of the network. From a forecasting point of view, it was shown that including environmental features and increasing the time horizon of the forecasts will increase the accuracy of the forecast. In contrast, it was shown that including environmental features did not provide any benefit to short-term forecast accuracy; however, longer input sequence lengths were demonstrated to be beneficial for short-term forecasts. Finally, an unsupervised anomaly detection algorithm, RAINFOREST, is presented which leverages the temporal context obtained from the forecasts alongside density-based clustering analysis to outperform all the baselines tested.

To Brittany

Acknowledgements

The pursuit of a PhD is not a simple task. Without the extensive support of my wife, Brittany, I would not have been able to complete it. Her endless support and care have made this endeavor possible. I also thank my supervisor, Brent Petersen. He started as a client on my undergraduate capstone project, and eventually became both my MScE and PhD thesis supervisor. Throughout all of these projects, he has been supportive and an excellent enabler. Supervision of such high quality is hard to come by. This research was funded by MITACS, NBIF, and the University of New Brunswick. A big thanks is given to Xplore Inc., my MITACS partner, for supporting me financially and for providing me with the resources I needed to succeed. A special thanks to Anthony Brown, Dave Leger, and Harshal Soni for their guidance and allowing me to research solutions to real world problems. Finally, I would like to thank the technical and office staff of the electrical and computer engineering department. Special thanks are given to Troy Lavigne, Shelley Cormier, Denise Burke, Dawne McSorley, Bruce Miller, Ryan Jennings, and Mike Morton.

Table of Contents

Vita

List of Tables

List of Figures

List of Symbols, Nomenclature or Abbreviations

5G	Fifth-generation
ACF	Autocorrelation Function
AD	Anomaly Detection
ADF	Augmented Dickey-Fuller
ARIMA	Autoregressive Integrated Moving Average
AUE	Active Users
AutoML	Auto Machine Learning
DBSCAN	Density-based Spatial Clustering of Application with Noise
DL	Deep Learning
DMS	Direct Multistep
DSL	Digital Subscriber Lines
EM	Excess Mass
ETS	Expontential Smoothing
FC	Fully-connected

FWA	Fixed-wireless Access
GPU	Graphics Processing Unit
GRU	Gated-recurrent Unit
IMS	Iterative Multistep
INTRF	Interference
KPI	Key Performance Indicator
KPSS	Kwiatkowski-Phillips-Schmidt-Shin
LSTM	Long Short-Term Memory
LTE	Long-term Evolution
MI	Mutual Information
MIMO	Multi-Input Multi-Output
ML	Machine Learning
MSTL	Multiple Seasonal-Trend Decomposition using LOESS
MV	Mass Volume
N-BEATS	Neural Basis Expansion Analysis for Interpretable Time Series
N-HiTS	Neural Hierarchical Interpolation for Time Series
NN	Neural Network
PACF	Partial Autocorrelation Function
PC	Pearson Correlation
PR	Precision-Recall
PRB	Physical Resource Block
ROC	Receiver-Operating Characteristic
SARIMA	Seasonal ARIMA

SE	Spectral Efficiency
STL	Seasonal-Trend Decomposition using LOESS
Seq2Seq	Sequence-to-Sequence
TFT	Temporal Fusion Transformer
THRP	Throughput
k-NN	K Nearest Neighbors

Chapter 1

Introduction

Access to the internet has provided the world with an unprecedented level of connectivity. An internet user can purchase groceries online, have a video conference call with their co-workers, or stream their favorite musical artist with ease. The dependency on the internet for work and entertainment has solidified it as a requirement for daily life in the 21st century.

The consumption of data through mobile and wireless devices has increased at a staggering rate [4]. LTE and 5G services are often deployed to provide service to urban and rural internet users. To accommodate the growth of the traffic consumed by each subscriber, network operators are rapidly scaling the capacity of their network. In urban environments, it can be simple to add more capacity; however, in rural areas, it is often difficult to provide high-speed broadband. Long-range channels and difficult terrain pose a significant problem for wireless communication using cellular technology.

Monitoring a rapidly scaling number of cells in a national rural fixed wireless network is no simple task. How can a network operator monitor problems, dynamically handle resources, and make long-term planning decisions when the scale has grown beyond what is possible for large teams of engineers to manage? Machine learning and automation have been suggested as keys to enabling the advancement of wireless networks [5].

Wireless communication networks produce enormous amounts of data that describe their performance and configuration. Performance and to some extent configuration are intrinsically temporal signals. Internet traffic is generally higher during the day when users are online and lower around 2 am when most users are asleep. Analysis of these time series signals is critical for network operators to understand how their network is performing and for long-term planning. For example, a sudden sharp decrease in traffic could indicate that a network fault occurred and a long-term upward trend in traffic could suggest that upgrades are needed to meet growing capacity demands.

The focus of literature has been predominantly on urban mobile networks. This is not surprising since most subscribers to cellular network services reside in urban areas [4]. One of the key aspects of this work is the consideration of rural fixed wireless networks. This requires a paradigm shift in how a network can be designed and analyzed. Users are placed in fixed positions, meaning that they do not roam from cell to cell. The distance between users and the base station is no longer hundreds of meters as in urban networks, it is on the order of tens of kilometers. The effects of terrain will also be dramatically different; wireless propagation in one urban center versus another is likely very similar, while propagation in a forest is likely very different than in coastal plains. In addition to environmental factors, the populations of rural areas themselves are known to behave differently from their urban counterparts, which can further change the temporal dynamics of a rural network [6].

This work will aim to investigate three aspects of rural fixed wireless networks and compare them with similar urban studies. First, the long-term seasonal variation of the network will be considered. This will aim to characterize the implications of seasonal changes in the propagation environment and how they can be used to generate long-term forecasts for tasks such as network and capacity planning. Second, short-term forecasting will be investigated for use in dynamic resource allocation. Similar urban studies have not provided a robust analysis or statistical approach to the short-term temporal behavior of a wireless network, and rural networks have received little to no investigation. Finally, a novel unsupervised anomaly detection algorithm is proposed for LTE KPI fault detection that outperforms baseline models by using the temporal context obtained from time series forecasts.

1.1 Overview and Research Objectives

Modern telecommunication networks are a monumental source of data. The volume and diversity of the data sets are significant. For a wireless network, there will be user equipment, cells, sites, and network cores. Each of these components has hundreds or thousands of metrics being logged. In many cases, there are multiple unique sources of data. A user could have a set of data related to networking features, such as throughput, or electromagnetic features, such as interference levels.

The unprecedented scale of data being generated from telecommunication networks provides an impressive level of visibility into a network's performance. Each generational increase in cellular technology introduces an additional layer of complexity in terms of management and operation. This increase in complexity directly leads to more difficulty in monitoring and maintaining a network. Furthermore, as the number of cells and subscribers in the networks increases, the complexity will increase further. Automation, *Big Data* analytics, and machine learning are rapidly becoming the only way to address network-scale monitoring without significantly increasing the number of telecom engineers. It would not be surprising if the number of telecom engineers was not growing rapidly enough for traditional monitoring and analytics to be viable in the near future.

Although all components of a network are important to consider, in fixed wireless networks, the primary focus is typically at a cell level. Cells are the building block of the network itself. The stationary nature of users means that their cell connection options are small. The sites are carefully chosen in the best coverage location available; however, it is not practical to move the location of a given installation. This leaves cells as the largest optimization surface in the network, which is why this work focuses exclusively on the cell-level aspects of the network.

The three main research objectives of this work relate to characterizing the temporal and environmental properties of a rural fixed wireless network, the forecast implications of these relationships, and how these forecasts can be used to detect anomalous behavior at the network scale.

The temporal properties of communication networks are critical for business decisions. Weekly and daily seasonality can be used to understand network congestion. Monthly or yearly seasonality can be used to understand long-term trends, such as subscriber growth or performance trends. Seasonality is critically important for network monitoring. Urban seasonality has been highlighted in open datasets such as Big Data Italia [7] and several similar studies; however, no investigation of seasonality in rural fixed wireless networks has been conducted. One of the objectives of this work is to characterize the long-term behavior of rural fixed wireless LTE network KPIs. Additionally, mechanisms that could be causing seasonal changes on the network scale, such as temperature and snow, will be analyzed.

Existing urban mobile studies have highlighted that weather and environmental covariates can be used to greatly improve the accuracy of network performance forecasting [7]. Using the seasonality and environmental relationships identified, forecasting models with different time horizons were created to investigate the merits of covariate signals in the LTE KPI forecasting model. Short-term forecasts with horizons on the order of a day or less are expected to see minimal benefit from including environmental features due to the short-time dependencies of the network KPIs; however, long-term forecasts with horizons extending weeks or more are expected to benefit from these covariates.

Using LTE forecasting models and combining several AD methods, a novel hybrid AD method is proposed. AD is a major problem in LTE due to the rudimentary nature of built-in LTE monitoring systems. Managing the tuning of threshold-based alarms on a network scale is not practical or economical because of the time that network experts would spend for a minor improvement in monitoring. Anomalous behavior has a wide array of manifestations; however, common examples are users disconnecting due to power outages or interference issues reducing spectral efficiency. The proposed methodology combines statistical, clustering, and forecasting AD techniques to alleviate the drawbacks of each individual method while achieving robust and explainable results. Furthermore, the proposed method is unsupervised and can be tuned based on a small number of intuitive parameters.

1.2 Seasonal Properties of Rural Fixed Wireless LTE Networks

Changes in an environment over time are a common experience for the majority of the world's population. Some climates have wet and dry seasons, while others experience more seasons. This work investigates the impact of seasonal changes in rural Canada on a fixed wireless communication network.

The Canadian climate is harsh: the winter months can have temperatures below $-40°C$ while summers can exceed $40°C$. The transitions between these months are in autumn and spring, where the temperature changes are not drastic, but introduce changes in the foliage that can dramatically alter propagation characteristics and user behavior.

Existing studies have considered a small number of temporal samples and measurement locations [8, 9, 10]. The industry partnership that supported this research provided access to an unprecedented temporal and spatial resolution: data was collected from a rural Canada fixed wireless LTE network with a 15-minute sampling period. The unique nature of this data set is such that direct comparable studies do not exist.

A key objective of this work is to analyze the relationship between environmental factors and changes in network performance over time. Studies by Chee et al. have identified that there are changes that impact propagation characteristics; however, this work aims to determine if there are measureable changes in network performance. These changes may be caused by the physical environment, user behavior, or some combination of both.

Using the relationships identified between environmental changes and network performance, forecasting models were created to test how these relationships could be used to improve modeling. The objective of these models was to help with capacity planning and management, which meant that the time horizons were on the order of months. A variety of models and forecasting horizons were tested; however, it was ultimately shown that the amount of information added by environmental signals was minimal and resulted in minimal changes in forecast accuracy.

1.3 Short-Term Multivariate Forecasting

Approaches to automated network monitoring and dynamic resource allocation often rely on short-term forecasting. Being able to accurately estimate the future demand and performance of a network will allow anomalies to be detected or resources to be reallocated as needed.

The general premise of forecasting is to consider some historical data and use them to predict the future. Depending on the forecasting task, the history could be a few hours, a few days, or longer if needed. Short-term forecasting has been considered in urban LTE networks [7, 11, 12]; however, rural networks remain largely ignored. Furthermore, the forecasting architecture approaches are often rudimentary and do not consider newer time series forecasting neural network architectures.

In existing studies, there are generally limited or no statistical analysis performed. Although many neural network approaches do not require significant investigation into the properties of the signals being forecasted, it is noteworthy that early analysis can help reduce the hyperparameter tuning search space.

This work considers multivariate forecasting of rural fixed wireless

LTE network KPIs using a variety of neural network architectures. The primary focus will be on characterizing the time history requirements needed to achieve accurate single-step forecasts. One of the contributions of this work will be to evaluate the time-history requirement of a variety of models. *Generalized* or *theoretical* neural network architecture research, such as [1] or [2], will focus on the performance of a neural network on a variety of benchmarks. It is important to note that this work is application-based, where the objective is to maximize performance for a specific application. In other words, this work aims to investigate how to optimize single-step multivariate LTE KPI forecasts rather than optimize the architecture to perform well on a variety of benchmarks.

1.4 Robust Anomaly Detection

Quickly detecting anomalies is the key to ensuring a good end-user experience in any commercial network. It is inevitable that problems will occur, given the scale of equipment required to provide services; however, detecting and mitigating problems when they occur will help reduce their impact on end users.

Anomaly detection for LTE networks has been investigated; however, the approaches are problematic and make erroneous problem formulation assumptions. Namely, algorithms generally adopt approaches that force a specific percentage of samples to be considered anomalies regardless of their true status [13, 14, 15].

This work investigated an alternative approach that does not enforce or assume properties of the sample data. Using forecasting, clustering, and statistical analysis, an anomaly detection algorithm was proposed to offer a more robust anomaly detection process. The proposed algorithm was tested over a rural fixed wireless network and was found to outperform the baseline methods. Furthermore, a case study of a known outage is presented to demonstrate the operation and interpretability of the algorithm itself.

Chapter 2

Background and Literature Review

2.1 Big Data and Open Source Software

Big Data is often used as a nebulous term to describe the existence of a large amount of data. What is a large amount of data? There is no threshold for what is considered big data; however, many researchers, scientists, and engineers begin to encounter big data problems when Excel spreadsheets are no longer able to open files and perform analysis in a timely manner. In the context of this work, big data is loosely defined as data that is larger than the memory of a single computer. There is also often the connotation that data must be poorly structured and require extensive cleaning and processing; however, *Big Data* is more of a problem of scale than of usability, since data can be bad in any volume. As the scale of available data grows, simple questions like *how many users were on the network today by hour?* begin to require analyzing gigabytes or terabytes of data.

Data collection for this work began in September 2020. The primary source of data was LTE and 5G cells distributed throughout Canada in rural areas. Every 15 minutes, thousands of cells collect thousands of metrics related to performance and configuration; the scale of the raw data is easily on the order of terabytes. The cleaned and distilled data remain on the order of terabytes per year. Data on this scale pose two significant problems: how can data be both stored and analyzed efficiently?

Storage of data on the terabyte and petabyte scale is not a new problem. The data collected for this research are primarily in tabular form, so it is natural to use a database. Various database options are available; however, Postgres [16] is an incredibly popular choice, as it has a large community and is open source. Furthermore, there are extensions of Postgres that allow for more efficient handling of geospatial and time series data. Furthermore, Postgres natively supports scalability features such as high availability through replication.

Although large-scale storage is straightforward to implement, computing resources can be problematic. Since no single computer is capable of handling the sheer volume of data, distributed computing is an inherent requirement. Open source software has grown exponentially to fill this void in large-scale analytical work. The creation of open source distributed computing frameworks such as Dask [17] and Spark [18] serves to scale massive workloads to arbitrarily large computing clusters. The independent nature of LTE and 5G cells easily allows for parallelization. In other words, most operations and analysis are segmented by cell, allowing workloads to be distributed across as many CPU cores as there are cells.

In many large-scale computational problems, analysis deadlines can be relatively short. For example, a business could look to analyze quarterly sales volumes or predict what advertisements would perform best with a specific user. Both of these operations can take days or weeks to process, without the results becoming stale and outdated. In the context of this work, there are short and long time horizons that can result in a variety of deadline requirements. Most of the analysis is short-term, where the timescale is on the order of minutes, while additional analysis is done on a much longer timescale where yearly trends are analyzed. Optimization of performance and scaling a system for short-term deadlines will inherently enable long-term analysis deadlines to be met. The focus of this work is not distributed computing or data warehouse designs; however, understanding the scale and application of this work provides context for why it is important.

2.2 Wireless Communication

Internet access methods vary widely depending on location and cost. In simple terms, urban internet speeds are typically very fast and cheap, whereas rural Internet speeds are typically slow and more expensive. The density of urban areas makes it cost-effective for modern internet infrastructure to rely largely on fiber optic connections. Wireless local area networks (WLAN), long-term evolution (LTE), and 5^{th} generation (5G) cellular networks provide mobility and edge connection to users. Rural internet users do not have the same quality of internet access as urban counterparts; the low population density of rural areas forces many users to rely on satellite, fixed wireless, and antiquated DSL connections. Governments in countries like Canada have been pushing to improve rural internet access [19]; however, finding cost-effective deployment methods is a constant concern.

In Canada, most rural internet coverage is provided through fixed wireless or satellite platforms. Fiber and DSL options exist; however, their availability is limited by region and population density. Fixed wireless implementations of LTE and 5G are a popular access method in rural areas. Ericsson projects that there will be more than 300 million fixed wireless access (FWA) subscriptions worldwide by the end of 2028 [4]. Although this is a significant number of subscriptions, it is nearly two orders of magnitude smaller than the projected figure of 9.2 billion mobile subscriptions worldwide by 2028. More subscribers are forcing service providers to rapidly expand capacity; however, the compound annual growth rate of traffic per device is expected to grow 21% year-over-year meaning that they need to accommodate not only subscriber growth, but also traffic growth.

More users who consume exponentially increasing amounts of data pose a significant problem for network operators. The cellular concept allows efficient reuse of spectrum resources by segmenting coverage by area; however, Shannon's capacity theorem ultimately limits what is possible with a single block of spectrum [20]. 5G is promising as it is able to achieve higher spectral efficiencies and allows broader spectrum allocations; however, equipment upgrades are not instantaneous, and users will surely grow to exceed 5G's traffic capacity in the long term.

2.2.1 Urban Versus Rural Networks

In an urban environment, the rapid deployment of the wireless network infrastructure is almost trivial. Network hotspots can be identified and coverage can be added as needed. The time it takes an infrastructure team to set up a new cell or upgrade an existing one is relatively short. Existing buildings and infrastructure can easily have networking hardware mounted on them, which serves to further accelerate the deployment. For rural fixed wireless networks, this is not the case for several reasons. First, the infrastructure of a rural wireless network is inherently remote. Sites can be deployed hundreds of kilometers from major cities, making physical access difficult. Second, the deployment conditions are also physically different and do not allow for the same amount of coverage redundancy. It is generally not practical or difficult to provide coverage to a single user using multiple cells. Furthermore, while networking hardware can be mounted on structures such as silos and barns, it is not always practical or possible to do so.

Urban networks will leverage several different cell sizes. Cells with large radii, often called umbrella or macro cells, can be used to provide coverage in high-mobility situations, such as on roads. In areas with higher density and lower mobility, it is typical for more cells to be deployed with much smaller radii to allow better spatial spectrum reuse and capacity. Rural networks, especially those that are fixed wireless, do not have the same design paradigm. Due to the remote nature and low population density in rural areas, the size of cells becomes much larger. Despite the fact that cells have radii on the order of tens of kilometers, it is often unlikely that a user can connect to more than one or two cells. In cases where a user can connect to multiple cells, it is often due to a shared coverage footprint shared between several cells for load-balancing purposes.

The objective of any commercial wireless communication network is to provide quality service as efficiently as possible. In this sense, the efficiency of wireless communication is paramount. For many network operators, spectrum licensing is a large part of their operating costs. A recent spectrum auction in Canada for allocations around 3.5 GHz had a final cost of almost $9 billion CAD [21]. These exorbitant costs lead to a desire to ensure that spectrum is used efficiently. Spectral efficiency is therefore a key performance indicator (KPI) of special interest and is loosely defined as the throughput of a cell divided by the bandwidth. The theoretical limit is bounded by the channel capacity C, which is defined as

$$C = B \ \log_2 (1 + \gamma), \tag{2.1}$$

where B is the channel bandwidth and γ is the signal-to-noise ratio (SNR) [20]. This definition of capacity is only valid under conditions where the channel experiences frequency flat fading and single-input single-output (SISO) communication is used. For broadband LTE networks, these assumptions do not hold as several MHz of bandwidth with multi-input multi-output (MIMO) will inherently experience frequency-selective fading.

Many wireless communication systems, such as LTE and 5G, use MIMO to increase spectral efficiency. Using multiple antennas allows multiple signals to be transmitted simultaneously or a single symbol to be transmitted multiple times concurrently to decrease the probability of error. The former is spatial multiplexing, while the latter is known as transmit diversity. Eqn. 2.1 can be generalized for frequency-selective fading MIMO channels by accounting for the number of antennas, subdividing the channel bandwidth into frequency-flat fading components and characterizing the channel impulse response. The normalized bandwidth form can be expressed as

$$C = \frac{1}{K} \sum_{i=0}^{K-1} \log_2 \left(\det \left[I_N + \frac{\gamma}{M} H_i^H H_i \right] \right), \tag{2.2}$$

where K is the number of discrete uniform frequency bins such that the subchannels experience frequency-flat fading, M and N are the number of transmitters and receivers, respectively, I is the identity matrix, γ is the SNR, H is the channel impulse response matrix of dimensions M by N, and $(\cdot)^H$ is the Hermitian transpose [22].

Both Eqn. 2.1 and Eqn. 2.2 demonstrate a strong dependence on SNR. Although Eqn. 2.1 indicates that capacity grows linearly with bandwidth and logarithmically with SNR, there are practical limitations to both. As previously mentioned, spectrum licenses are prohibitively expensive. Network operators can and do leverage unlicensed spectrum such as ISM bands around 2.4 and 5.7 GHz; however, ensuring quality service for end users can become difficult, as they ultimately have no control over who is using that shared spectrum. Bluetooth, Wi-Fi, and microwave ovens are a few examples of spectrum uses that would compete with LTE and 5G for the use of unlicensed spectrum.

Increasing capacity using SNR improvements and higher ranks of MIMO has become a popular research topic. Higher ranks of MIMO are achieved by maximizing the diagonal components of $H_i^H H_i$ while minimizing the non-diagonal terms. In other words, the objective is to minimize interference between non-diagonal antenna pairs when multiple antennas are used. In the context of rural fixed wireless LTE networks, it has been shown that the properties of H can be modified by carefully selecting antenna spacing [23]; however, network operators often forego this in favor of standard vendor solutions such as massive MIMO.

Massive MIMO has become a key technology for the deployment of high-performance wireless networks [24]. Historically, networks have been deployed with an arbitrary coverage footprint and users are distributed throughout the coverage area. Antenna beamforming was possible; however, the number of antennas and control over beamforming were rudimentary. Massive MIMO is aptly named and uses an enormous number of antennas to significantly increase the capabilities and control of beamforming. Massive MIMO gains come largely from beamforming. Users that are spatially separated can share spectrum while still achieving full speed communication. Furthermore, beamforming allows users to have antenna beams electrically steered, further increasing their SNR. In other words, massive MIMO allows a single cell to create dynamic pseudocells within it by beamforming.

One of the common example use cases of massive MIMO is in urban environments where a massive MIMO cell provides coverage to a nearby multi-story building. Azimuth and elevation beamforming can be used to provide excellent performance to users with sufficient displacement between them relative to the incident wave from the basestation. Although this is a great real-world example of a massive MIMO use case, what happens in a rural propagation environment? Users are often so far away from the basestation that elevational beamforming is no longer meaningful. Often times, it may even be difficult for some users to benefit from azimuthal beamforming; however, it can still be beneficial in cases with sufficient azimuth spacing.

Despite best efforts to improve broadband spectral efficiency, free space pathloss will ultimately introduce problems for providing high SNRs to remote users. Friis' equation dictates the power level at a receiver to be

$$P_r = \frac{G_t G_r \lambda^2}{(4\pi R)^2} P_t, \tag{2.3}$$

where P_r is the power at the receiver, P_t is the power at the transmitter, G_r is the gain of the receiver, G_t is the gain of the transmit antenna, λ is the carrier wavelength, and R is the distance between the transmitter and receiver [25]. In simple terms, the wave spreads spherically as it propagates from the transmitter to the receiver. Eqn. 2.3 assumes an ideal environment with no effects from the channel except for path loss. A generalization of this is to exchange the exponent of 2 for an arbitrary exponent, n, which is related to the specific propagation environment. Using logarithms, Eq. 2.3 can be rewritten more generally as

$$P_{t,dB} - P_{r,dB} = G_{t,dB} + G_{r,dB} + 10\log_{10}\left(\frac{\lambda}{4\pi R}\right)^n \tag{2.4}$$

where n is the path loss exponent. The free-space pathloss is $n = 2$. Indoor line-of-sight paths are often $n < 2$ due to propagation phenomena such as waveguiding and reflections along hallways and in rooms. In urban environments, n varies significantly more than the indoor line of sight, but is generally less than 5 [26]. Rural environments can vary much more significantly than urban ones in terms of pathloss.

In an urban cell, the distance between users and a given tower is likely to be on the order of a few hundred meters. Higher mobility urban cells may push that distance to a few kilometers. In contrast, rural cells will have propagation distances that are almost always longer than one kilometer and can stretch up to 30 kilometers. Terrain is often mixed in urban environments; hills, buildings, cars, or overpasses could block or obstruct signals. Rural propagation will experience obstruction from hills, mountains, and other problematic terrain characteristics. Ultimately, they represent two sets of propagation environments with suburban mixing of both. For example, the path loss difference between an urban and rural cell can easily exceed 20 dB at 3.5 GHz. Furthermore, this does not account for any changes in the path loss exponent, which typically further penalizes rural channels. Assuming that bandwidths and, thereby, thermal noise are kept constant, a rural cell will require significantly more power to achieve levels equivalent to those of an urban cell. Practical and regulatory [27] limits restrict the maximum power that can be used, further putting rural users at a disadvantage in terms of spectral efficiency.

2.2.2 Propagation Factors

For any modeling task, it is important to understand the fundamental properties of the system being modeled. In the context of this work, the models will characterize and describe the performance of a rural fixed wireless network. Temporal user behavior is a major driver of network traffic, that is, high use in the evenings and low use overnight; however, the time of day may not account for all seasonal changes in a given cell. Beyond the daily schedule of users, there is also a need to consider the long-term changes that can occur in the propagation environment.

Canada is well known for its large seasonal changes. Summer months are warm and can exceed 30° C while winter months can dip below -40°C. These seasonal changes are also highly location dependent. This can lead to a few different phenomena which will be discussed in later sections; however, water freezing, snowfall, and foliage are just a few examples.

In any environment, it is important to understand permittivity and how it can affect propagation. One of the main ways permittivity will affect propagation is through reflection and transmission. When a plane wave propagates through a lossless medium with permittivity ϵ_1 and collides with another medium with permittivity ϵ_2, the wave will be reflected and transmitted. The reflected wave will remain inside the first medium, while the transmitted wave will pass into the second.

To understand the magnitudes of the transmitted and reflected signals, the angles of incidence and reflection, as well as the permittivity of both mediums, must be known. Each medium will have an impedance defined as

$$\eta = \frac{\mu}{\epsilon}, \tag{2.5}$$

where η is the impedance, μ is the permeability and ϵ is the dielectric constant of the medium.

The reflection coefficient Γ and the transmission coefficient τ depend on the incident angle and the impedance of the mediums. In simple terms, the transmission and reflection coefficients describe the phase and magnitude of the signal passing into the second medium and the signal staying in the first medium.

When the Poynting vector of the wave is parallel to the normal vector of the plane formed by the boundary of the two mediums, the reflection and transmission coefficients can be written as [28]

$$\Gamma = \frac{\eta_2 - \eta_1}{\eta_2 + \eta_1}, \tag{2.6}$$

$$\tau = \frac{2\eta_2}{\eta_2 + \eta_1}. \tag{2.7}$$

Although there are many cases where a wave enters a new medium at an incident angle, it is more likely that the wave enters at some angle. This angle will cause the transmission and reflection coefficients to change as the angles change. Moreover, in most cases, the relative permeability of both mediums can generally be assumed to be unity, $\mu_1 = \mu_2 = \mu_0$, when ferromagnetic materials are excluded. With this assumption, Eqn. 2.7 and 2.6 can be written more generally as

$$\Gamma = \frac{\cos\theta_i - \sqrt{\frac{\epsilon_2}{\epsilon_1}}\sqrt{1 - \left(\frac{\epsilon_1}{\epsilon_2}\right)\sin^2\theta_i}}{\cos\theta_i + \sqrt{\frac{\epsilon_2}{\epsilon_1}}\sqrt{1 - \left(\frac{\epsilon_1}{\epsilon_2}\right)\sin^2\theta_i}} \tag{2.8}$$

$$\tau = \frac{2\cos\theta_i}{\cos\theta_i + \sqrt{\frac{\epsilon_2}{\epsilon_1}}\sqrt{1 - \left(\frac{\epsilon_1}{\epsilon_2}\right)\sin^2\theta_i}} \tag{2.9}$$

where θ_i is the incident angle relative to the normal vector of the medium boundary plane [29]. A visualization of these two curves for several different media changes can be seen in Fig. 2.1. Furthermore, a diagram of the physical configuration described by these equations can be seen in Fig. 2.2.

The values of θ_i and θ_r for greenfield channels in a rural environment will have values close to $\pi/2$. From Fig. 2.1 it can be seen that as the incident angle approaches $\pi/2$ that the reflection coefficient approaches unity regardless of the boundary media permittivities.

It is typical for long-range fixed wireless rural networks to be configured in such a way that users experience line-of-sight (LOS) or near LOS. This is important, since it will allow users to mitigate the effects of propagating through trees or diffracting over terrain. Despite this attempt to mitigate additional path loss, there will typically still be signal paths that go through trees and around terrain. This will mean that incident angles are much smaller, meaning that transmission and reflection can occur, as opposed to primarily reflections for grazing angles.

Scattering is generally modeled as isotropic with some retransmission efficiency or attenuation introduced by the object. Scattering could be caused by foliage, trees, buildings, etc. In rural environments, it is common to encounter double scattering. One set of scatterers is near the transmitter

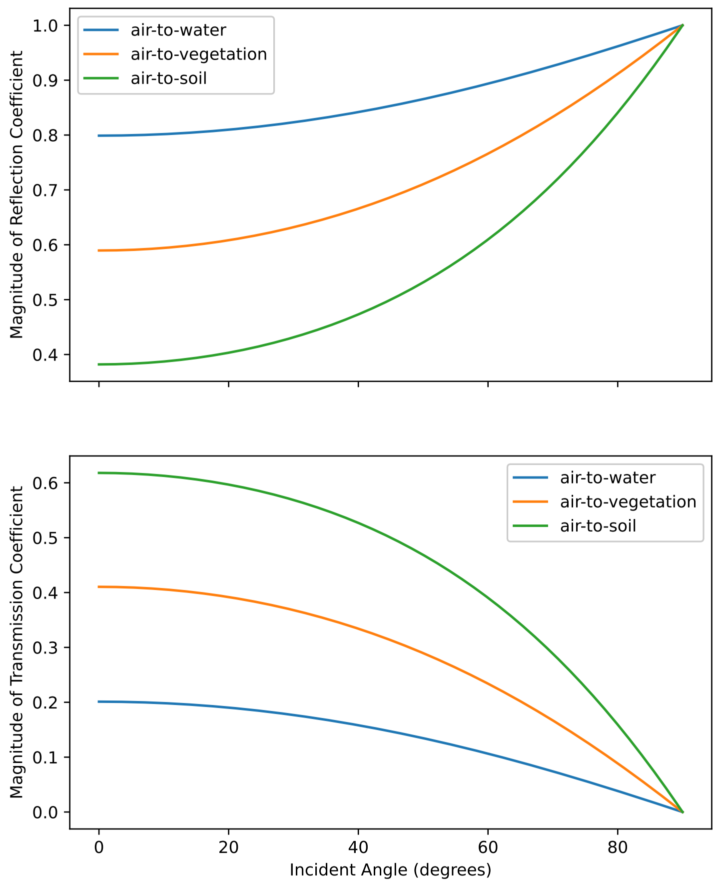

Figure 2.1: Reflection and transmission coefficient magnitude for several different boundary conditions for $\theta_i \in [0, 90]$ degrees. $\epsilon_{r,air} \approx 1$, $\epsilon_{r,water} \approx 80$, $\epsilon_{r,vegetation} \approx 15$, $\epsilon_{r,soil} \approx 5$ [3].

and another near the receiver. When the distance between the transmitter and receiver is long, double scattering channels will form keyhole channels, which can be problematic for channel capacity [30].

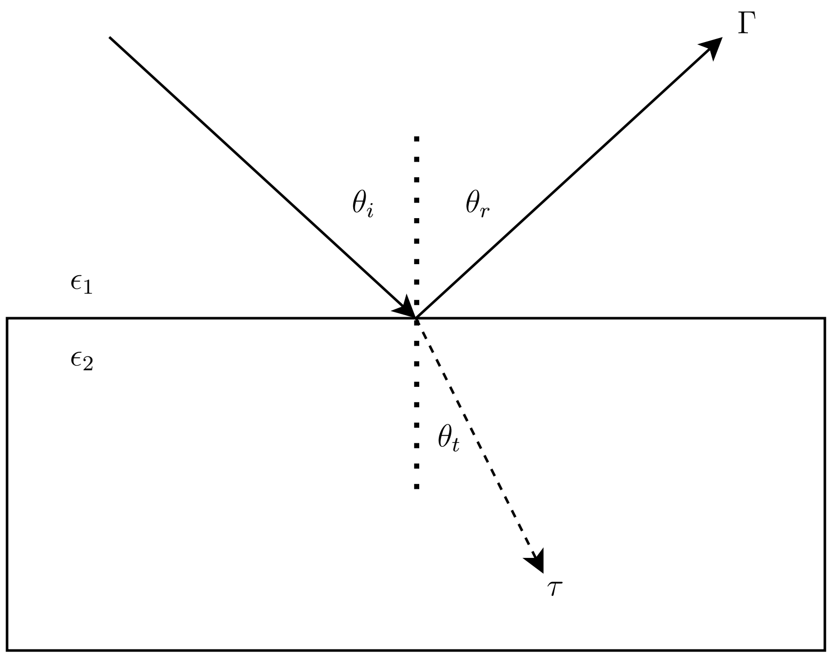

Figure 2.2: Example of boundary conditions for incident, reflected, and transmitted wave.

Local scattering will result in a much larger number of strong signals reflecting at nongrazing angles. Because of this, an environment with changing permittivity may see changes in performance. Snow, rain, foliage, and other transient changes in a propagation environment could lead to performance changes.

2.3 Time Series Forecasting

The ability to accurately predict the future values of a time series signal is of monumental interest in many industries. Weather [31], stock prices [32, 33], electricity demand [34], sales [35], and traffic [36] are just a few examples of areas that have gained significant traction in the time series forecasting space. At its core, time series forecasting is a regression problem that takes historical data for a set of input variables and makes predictions in the future.

Forecasting tasks can be used to address a variety of problems. Different tasks require different models and techniques depending on the problem being considered. A few common considerations are whether the problem is univariate or multivariate, single- or multi-step forecasting target, stationary or nonstationary time series, and finally the inclusion of extra signals. Time series forecasting research generally takes one of two approaches; studies typically either optimize a model across a number of benchmark datasets [37] or optimize a set of models on a given forecasting task [1]. This roughly divides the lines between 'pure' forecasting modeling research and machine learning application engineering research. Both approaches are crucial in that the theoretical approaches provide tools for application-based research.

2.3.1 Problem Formulation

Univariate single-step forecasts are the simplest form of time series forecasting. Several points from the past are fed into a model, which then predicts the next value in the series. The signal of interest is often called the target variable, or target signal. Although univariate forecasts can be useful in simple systems, it is often insufficient for complex forecasting models to limit the number of target signals and the number of forecast steps to one. To address this desired increase in model complexity, it is often required that forecasts become multivariate and multistep in nature. A simple multivariate forecasting problem can be written as

$$\hat{y}_{i,t:t+n} = f(y_{i,t-m:t-1}) \tag{2.10}$$

where y_i is the i^{th} target signal, $\hat{\bullet}$ indicates the estimate, t is the discrete time step, m is the number of historical steps, n is the number of future time steps, and $f(\bullet)$ is the forecast process. Historical and future time steps are often used interchangeably with the input sequence length and the output sequence length, respectively.

In addition to the desire to forecast several target signals simultaneously, it is often desirable to consider additional signals. For example, when predicting the number of people buying umbrellas, it could be beneficial to know what time of year it is and the current weather. It should be noted that the forecast proposed in Eqn. 2.10 does not account for the inclusion of covariate signals. Although most of the information in a signal is typically contained in the target signals, external signals commonly referred to as exogenous features or covariates, can be used to improve forecast accuracy. There are several types of covariate that can be used: past, future, and static covariates. Past covariates are signals with known historical values. Future covariates are signals with known historical and future values; time is the most commonly used future covariate. In essence, static covariates are additional data on the entity being considered. In the context of LTE cell performance forecasting, this could be information related to location, equipment, bandwidth, etc. This allows Eqn. 2.10 to be rewritten more generally as

$$\hat{y}_{i,t:t+n} = f(y_{i,t-m:t}, x^p_{i,t-m:t}, x^f_{i,t-m:t+n}, x^s_{i,t-m:t}), \tag{2.11}$$

where x^p are past covariates, x^f are future covariates, x^s are static covariates, i is the input signal index, t is the discrete time step, m is the number of input steps, and n is the number of target steps. It is important to note that future covariates have a known history but also have a definition beyond the current time, t.

2.3.2 Loss Metrics

Forecasting ultimately seeks to create a model that minimizes the selected error metric. The basis for many of these metrics is the error described as

$$e = y_{i,t:t+n} - \hat{y}_{i,t:t+n}. \tag{2.12}$$

The mean squared error (MSE) and the root mean squared error (RMSE) are two of the most commonly used error metrics. For a forecast with N targets and T time steps, the MSE is defined as

$$e_{mse} = \frac{1}{NT} \sum_{i=0}^{N-1} \sum_{t=1}^{T} e_{i,t}^2, \tag{2.13}$$

where the RMSE is simply the square root of Eqn. 2.13.

It is often typical for the mean absolute error (MAE) to be considered alongside either the MSE or RMSE. It is defined as

$$e_{mse} = \frac{1}{NT} \sum_{i=0}^{N-1} \sum_{t=1}^{T} \left| e_{i,t} \right|, \tag{2.14}$$

where N is the number of target signals and T is the number of forecast steps.

Another metric that will be considered in this work is the coefficient of determination R^2. It is the square of the correlation between the actual and forecasted signals. It can also be written as

$$R^2 = \frac{\sum \left(\hat{y} - \bar{y}\right)^2}{\sum \left(y - \bar{y}\right)^2}, \tag{2.15}$$

where $\bar{y}$ is the mean of the target variable throughout the time horizon of interest [38].

Using error metrics, such as those mentioned, helps drive the overarching objective of finding objectively good models. It is important to select good and robust error metrics to allow comparison between different models and configurations. Many of the older statistical forecasting models are still relevant and are used as baselines, such as the Holt-Winters models [39]. Over time, these models have evolved from autoregressive techniques like autoregressive integrated moving average (ARIMA) and seasonal ARIMA (SARIMA) to advanced neural network-based architectures such as the temporal fusion transformer (TFT) [40] and neural hierarchical interpolation for time series (N-HiTS) [1]. Improvement in forecast accuracy is often demonstrated by comparing the performance of existing models with new forecast architectures, making it critical that metrics be standardized for repeatability.

2.3.3 Time Series Decomposition

A key concept that appears in many different types of forecasting models is the decomposition of time series. The objective of time series decomposition is to break down the signals of interest into different components that describe its trend, seasonality, and residuals. Time series decomposition is not unlike segmenting a signal into its high, mid, and low band frequency components. Many time series data sets are very noisy, making it difficult to extract meaningful information from the raw signals. Decomposition does not explicitly remove noise; however, strong periodic components and trends can be extracted to improve analytical capabilities.

In broad terms, there are two approaches to decomposition: additive and multiplicative. Additive models can be expressed as

$$y_t = T_t + S_t + R_t, \tag{2.16}$$

while multiplicative models can be expressed as

$$y_t = T_t \times S_t \times R_t, \tag{2.17}$$

where y_t is the signal that is decomposed at time t. T, S and R are the trend, seasonal, and residual components, respectively [38].

Time series decomposition is a valuable technique when considering signals that exhibit changes in both trend or seasonality. Many statistical forecasting models struggle with nonstationary signals. If a signal's mean or variance changes, it can be difficult for some models to make meaningful predictions. The decomposition allows the trend and seasonality components to be separated while using the residuals to account for any additional variations in the signal. An example of an additive time series decomposition can be found in Fig. 2.3. The raw signal is very noisy, but visual inspection indicates that there is some periodicity to the signal. Although it depends on the exact decomposition algorithm selected, the trend line can be roughly approximated as a rolling average of the signal. The window size or expected frequency to be used can generally be specified; in this case, the periodicity was set to one day. Compared to the original signal, the periodicity of the seasonal component is much more apparent. Some of the flat spots caused by peak hours around November 20-21 have been removed, and the shape of the daily signal is clearer. Finally, we can observe that this is a rather noisy signal since the remainder contains a moderate amount of power; however, the assumption that the remainder signal is entirely noise is a risky assumption. Sometimes there can be periodic components that were simply not captured by the trend or seasonal components, especially in the event of a signal with multiple seasonalities.

Several different time series decomposition techniques have been developed over the years. The most popular method appears to be seasonal trend decomposition with local regression (STL) [41]; however, other methods have been developed for specific time horizons or applications. Some of these methods include seasonal extraction in ARIMA time series (SEATS) and X-11. STL is desirable in that it can handle arbitrary seasonalities, changes in seasonality, and robustness to outliers [38]. Furthermore, STL was developed to operate rapidly in long time series with large amounts of smoothing.

Another powerful feature of STL is that it allows the input and decomposed sequences to have the same length; this is not the case for all decomposition methods. Many decomposition techniques use sliding-window averaging to extract the trend component. This forces the endpoints of the signal to be dropped unless the analysis is willing to accept partial moving average windows. STL extrapolates these values, which can be problematic depending on the use case; however, in general the performance is

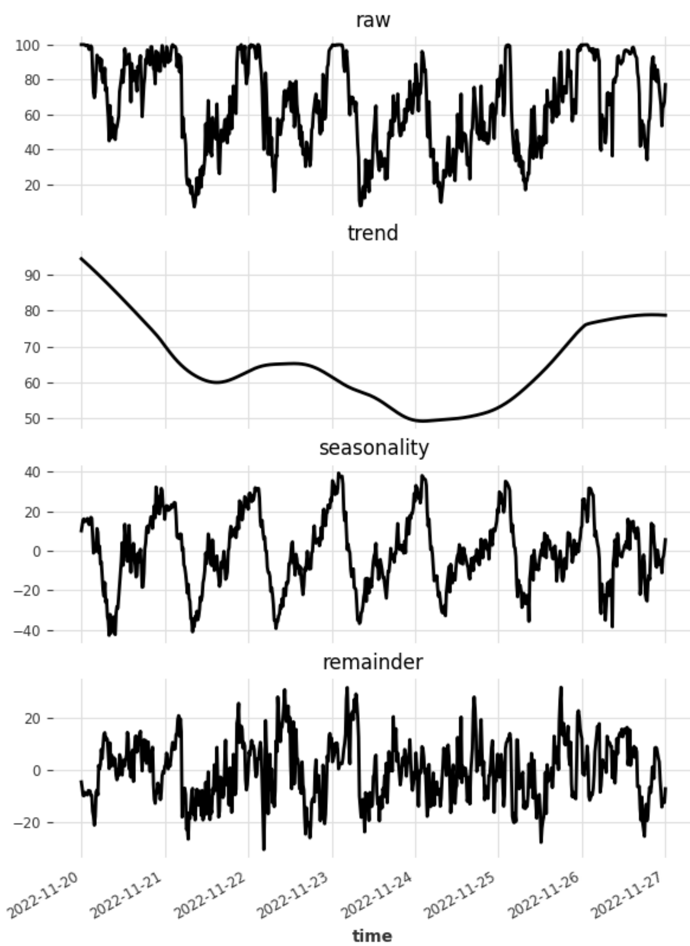

Figure 2.3: STL decomposition of downlink physical resource block use.

adequate [41].

One of the deficiencies of most decomposition techniques is that they do not have the ability to capture multiple seasonalities. For example, it is safe to assume that the LTE KPI data will have a daily, weekly, and yearly seasonality. Depending on the time horizon used for the analysis, the effects of different seasonalities will be present. STL has been adapted to handle multiple seasonalities to address this specific edge case [42]. This will allow all of the unique seasonalities and trends to be extracted to provide better context for what is happening in an arbitrary window of time.

2.3.4 Statistical Relationships and Properties

The rise of deep learning models for time series forecasting has led to a significant downturn in the robustness of feature analysis. The architectural research of neural networks focuses on optimizing performance in a large number of benchmark datasets; however, feature analysis is foregone in an attempt to provide a better comparison of the raw performance of the architecture [2, 40, 1, 43, 44, 45]. In a sense, the architectural research approach aims to minimize feature engineering and preprocessing. This approach is not without merit as it can accelerate the time for industrial applications to go from building to deploying models for tangible benefits.

Blindly feeding data into a neural network architecture is a problematic approach which has become commonplace in the current literature. In many cases, it is possible to achieve near- or actual-state-of-the-art performance with much simpler models and hyperparemeter tuning. Many forecasting applications are not time sensitive in their forecast generation. Predicting monthly or quarterly sales projections or weather for the next week have large time buffers with lax deadlines. For short-term LTE performance forecasting, this is not the case. Forecasts must be generated in a timely manner so that dynamic resource allocation and anomaly detection can be performed rapidly. Dynamically switching users between cells is not meaningful if the forecast is created after the next cell performance data collection period occurs.

Although many neural network architectures can ignore the statistical properties of the data in favor of simple preprocessing and sufficient training time, using smaller models allows for looser deadlines on forecast generation and further downstream analysis. In other words, for long-term analysis arbitrarily large models with minimal hyperparameter tuning can be adequate; however, for tight deadlines, tuning and architectural optimization must be considered.

There are several statistical analysis techniques that can provide significant information in preliminary hyperparameter selection for both statistical and machine learning models. For statistical models, it is often a requirement that they are fed stationary input signals; this will be analyzed using stationarity tests such as the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) and the Augmented Dickey Fuller (ADF) tests. The self-contained information in the signals will be analyzed using the autocorrelation function (ACF) and the partial autocorrelation function (PACF). Finally, Pearson's correlation (PC) and mutual information (MI) will be considered to evaluate the strength of the relationships between signals.

Time series signals change inherently over time. A constant time series is trivial to forecast an arbitrary number of steps ahead because it is a constant. Signals that are strongly periodic are also easy to forecast, as the properties are consistent over time. Stationary signals are those that have constant mean and variance. If a signal has a significant linear component that results in long-term growth, it is not stationary but trend stationary. Similarly, a person's time to complete a task could be highly variable, but as they become more proficient, the variability in their time to complete the task would decrease; this also results in a nonstationary signal.

Two statistical tests are commonly used to identify the stationarity properties of a signal; they are the KPSS and ADF tests. KPSS evaluates whether the data is stationary around a level or a linear trend, while ADF determines the presence of a unit root. Although both tests indicate the stationary properties of a signal, they test fundamentally different properties. The difference in properties allows more information about the signal to be determined.

For KPSS, the null hypothesis is that the signal is trend-stationary, while the alternate hypothesis is that it is not stationary. For ADF, the null hypothesis is that the signal has a unit root, while the alternate hypothesis is that there is no unit root. When using KPSS and ADF in conjunction for stationary testing, there are four possible results. The first occurs when both tests agree that a signal is stationary. The second occurs when the KPSS indicates stationary, while ADF does not; in this case, the signal is trend stationary. The third case is where KPSS indicates nonstationary while ADF indicates stationary; in this case, the signal is difference stationary. Finally, both tests can indicate that the signal is nonstationary [46]. In general, stationary testing is required for statistical models but is not required for neural network-based approaches. This is discussed further in Section 2.3.6.

Stationary testing is used to help identify a signal's temporal statistical relationship with itself. Although this provides insight into what types of model are usable for forecasting, it does not provide information about properties, such as how long a history is required or what covariates should be included. Self- and inter-signal relationships can be characterized using metrics like Pearson's correlation (PC), autocorrelation, partial autocorrelation, and mutual information.

PC is a very popular statistical metric for measuring the linear dependence of one signal on another. PC is defined as

$$R_{xy} = \frac{\sum_{i=0}^{n-1}\left(x_i - \mu_x\right)\left(y_i - \mu_y\right)}{\sqrt{\sum_{i=0}^{n-1}\left(x_i - \mu_x\right)^2}\sqrt{\sum_{i=0}^{n-1}\left(y_i - \mu_y\right)^2}}, \tag{2.18}$$

where R is the correlation, x and y are the signals being considered, n is the number of samples, and μ is the mean of the subscripted variable. From the Cauchy-Bunyakovsky-Schwarz inequality, it can be said that the magnitude of the numerator will always be less than or equal to that of the denominator [47]; this assertion leads to $R \in [-1, 1]$. The magnitude of PC indicates the strength of the relationship, while the sign indicates the direction of monotonic increase. For preliminary signal investigation, PC is a very powerful metric; however, it becomes problematic in situations where the relationships between variables are nonlinear.

For two independent variables, the PC will theoretically be zero. In practical applications, this may not be the case, but it will ultimately be of low value. When considering the joint distribution of these two variables, circular patterns will be observed in a two-dimensional scatter plot. Adding in a linear dependence between the variables will transform the circular patterns into ellipses. In simple terms, the PC will allow the linear dependence to be identified. Although PC is excellent at detecting linear dependence, it begins to breakdown for nonlinear relationships or heteroskedasticity [48]. An example of these three cases can be seen in Fig. 2.4. The provided example of a nonlinear relationship is a quadratic relationship between the two variables. The value of PC was small, indicating that the relationship is insignificant; however, visual inspection indicates that there is clearly a nonlinear relationship.

To address PC's inability to capture nonlinear relationships, MI can be considered. MI represents the amount of information shared between two signals. In other words, how much is known about the signal x based on the signal y? It is not unlike the channel capacity described by Eqn. 2.1. More formally, MI can be written as

$$I\left(X,Y\right)=H\left(X\right)+H\left(Y\right)-H\left(X,Y\right), \tag{2.19}$$

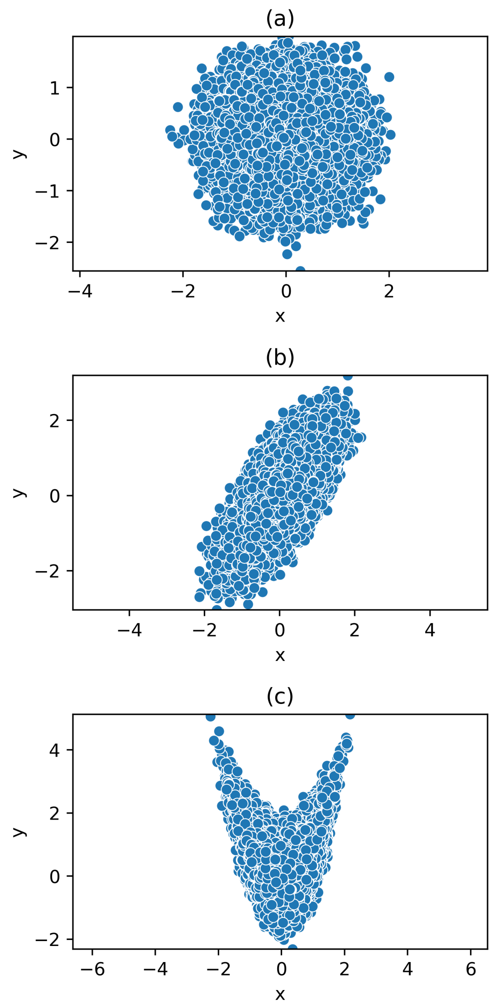

Figure 2.4: Joint distributions of datasets for feature relationship discussion. Subplot (a) is two Gaussian variables, subplot (b) is two correlated Gaussian variables, subplot (c) is a squared relationship between two Gaussian variables.

where I is the mutual information, X and Y are two unique processes or sets of signals, and H is the entropy. Furthermore, entropy can be expressed as

$$H\left(f\right) = -\sum_{a \in A} p_f\left(a\right) \log_2 p\left(a\right), \tag{2.20}$$

where f is a discrete random process with a set of values, A, and a is a value in the set A, and $p\left(a\right)$ is the probability of the result a being achieved [49].

PC is a simple metric to compute and is easy to calculate on large datasets; however, the same is not true for MI. From Eqn. 2.19 and Eqn. 2.20, it can be seen that the probability density function of the random process must be known to estimate the MI. Using k-nearest neighbor distances has been proposed as a way to minimize computational complexity while still achieving accurate results [50].

Although it is able to capture nonlinear relationships and is only zero in the case when the two signals being considered are truly independent, MI can be problematic in that it is partially bounded. PC is restricted to $R \in [-1, 1]$; however, MI is restricted to $I \in [0, \infty)$. Signals with lower MI are easier to forecast. For example, a signal that is constant will have $I = 0$. As the signal becomes more random, it contains more information and becomes harder to forecast. Unfortunately, low-information signals are rarely of interest for forecasting applications.

Adding additional signals to a forecasting problem can improve performance, but can also degrade it. A large part of this assumption hinges on the assumption that adding features will increase the information known about the target signals. The unbounded nature of MI can introduce some difficulty in the matter. If a signal had 10 bits of information and a covariate signal added 1 bit of certainty about the target signal, does it merit being involved?

Using the normalized mutual information is the easiest way to facilitate a 'quality' estimation for covariate signals. Similarly to a correlation matrix that describes the linear relationship between signals, the equivalent can be constructed using MI instead. In this case, the MI is similar to the covariance, and the desired result is the correlation; the objective is to constrain the scale of the variables. For an MI matrix, this can be done by normalizing a given row or column by the nearest diagonal term. Although MI is unbounded, it is limited by the information contained in either signal of interest. This allows for the assertion that $I(X,Y) \in [0, \min\{I(X,X), I(Y,Y)\}]$. By normalizing by the nearest diagonal term in the row, the relative amount of information about a signal is known using a bounded metric. This better facilitates the comparison of targets and covariates.

PC and MI are excellent metrics to identify when signals share relationships at a given sample time t_n, where n is an arbitrary time step. These metrics highlight the relationship between the features at t_n; however, this is ultimately not what is required for a forecasting problem. The objective of forecasting is to predict the future based on the present value, which means that the relationship between t_n and t_{n+m} is of great interest. Two common techniques for evaluating the relationship of signals over time are the autocorrelation function (ACF) and the partial autocorrelation function (PACF).

The ACF and PACF are very similar; however, they serve slightly different purposes. ACFs are used to identify the relationship between a signal x_t and its lags x_{t-k} for some integer k. The identified relationship is the strength of the correlation between the two signals. A key detail of this metric is that it evaluates the correlation between all shifted points, i.e. t_n against t_{n-k} for all valid n; however, it does not allow subcomponents of the signal to be extracted and analyzed. Although the ACF can provide insight into the signal, it is often necessary to consider the PACF as well. The ACF considers the full range of the signals for some lag K, while the PACF considers the relationships for a given lag without being affected by earlier lags [51].

Despite the ACF and PACF describing temporal relationships in the data, they will still continue to suffer from the same problems as PC. ACF and PACF are based on linear relationships between lags that will eventually break down for nonlinear signals.

2.3.5 Preprocessing

Preprocessing data for machine learning applications is often the key to ensuring quality learning outcomes. Although not applicable to all types of model, data preprocessing will help precondition the data in such a way as to improve the learning capacity of the model. For neural network-based approaches, this generally takes the form of input scaling, while statistical forecasting methods will require more extensive preprocessing and optimization. The focus of this work is on the first one, and the reason behind this decision is further discussed in Section 2.3.6.

When creating a global model that learns across many different time series, there are several ways to handle each individual time series. Most likely, each series was created by a unique entity, such as an LTE cell, a store, or a stock. If the assumption is made that the set of signals contain information that can be learned in a global context, the problem becomes deciding how to handle the input scaling. There are two approaches that are commonly used: entity-based and globally-based scaling.

The decision to use global or entity scaling can yield different results depending on the application. Some models, such as N-BEATS [43] go so far as to abandon input scaling, preprocessing, or feature engineering altogether. Global scaling for bounded signals is intuitive. For example, if the target signal was a percentage, it would be natural to use the global context of all signals, since they are bounded. Entity-based scaling is often used in applications where the scale of the signals is not related or constrained in the same way. A commonly used benchmark data set are stock exchange prices, where the price of a single stock could vary; however, global trends may be similar.

For univariate forecasting, it is intuitive that all unique entities are scaled using the same scaler properties. For example, MinMax scaling would have a global maximum and minimum used to scale all entities. Although it is intuitive for univariate forecasting problems, for multivariate forecasting problems, it is important to note that each unique feature will share the same scaler. Consider a set of LTE cells: features such as cell throughput will share a scaler, while the number of connected users will share another. Global scaling in multivariate applications does not share scaler parameters across features such as cell throughput and connected users.

Research studies often have the luxury of avoiding practical deployment conditions. For commercial production environment forecasting models, it is important to understand the implications of selecting a specific scaling and entity handling process. In the context of LTE KPI forecasting, this leads to the question of how can new cells being added and end-of-life cells being removed be handled without having to retrain a model and recreate scaler objects? The easiest way to address this issue is to use global scaling. Many KPIs are constrained to specific intervals or effectively so. For example, interference levels may have undefined limits; however, at a certain point, the cell will not function due to interference. If the selected set of entities for model training accurately represents the global limits of the KPIs then the scaler should be good for the addition or removal of entities from the deployment conditions.

Standardization is a popular scaling method for time series forecasting [2, 1]; however, it can be difficult to handle the addition or removal of entities from the set. For example, if a new entity were added with a different distribution, it could result in poor forecast accuracy. Moreover, global MinMax scaling will not be as sensitive to temporal fluctuations. Seasonal changes have been observed in LTE network KPIs and are discussed in further detail in Section 3. These seasonal fluctuations could lead to a change in mean and variance that changes the input range of features being fed to a model. Despite concerns about changes in mean and variance over time, it has been shown that using a sufficiently large training set and global scaling will ultimately have standardization outperform MinMax scaling [37].

2.3.6 Forecasting Models

Creating a forecast based on an input sequence can be done in a variety of ways. Ultimately, the input will be some set of signals, while the output will be the forecast. The generation of output sequences can be broken down into two categories: autoregressive and direct. Autoregressive models will iteratively build the output step-by-step, while direct methods will predict all output steps concurrently. These two methods are often referred to as direct multistep (DMS) and iterative multistep (IMS) [2].

The selection of a model is largely application-dependent. One type of model may outperform others for a given task. Additionally, there may be practical performance considerations that need to be taken into account when selecting a forecasting model. For a multi-step forecast, an autoregressive model will iteratively build a forecast one step at a time. If there are many steps to forecast, this becomes inefficient compared to direct models, which could generate the forecast in one shot. This does come with the caveat that the time complexity of the direct model must be less than the accumulative autogressive model's complexity to make this beneficial. This is an open topic in multistep long-term forecasting [1].

Beyond the forecasting output sequence generation, the inner workings of a forecasting model can be broadly classified as soft-computing or statistical methods. This segmentation of model architectures is not clear cut, but serves to segment the problem space into two common approaches. Statistical models often require an extensive analysis and processing of target signals to carefully select model parameters [11, 52]. On the contrary, many soft-compute methods only need preprocessing in the form of scaling the input signal. Some soft-compute models, such as N-BEATS [43] go so far as to abandon input scaling, processing, or feature engineering altogether. Furthermore, soft-compute models generally do not care about properties such as the signal being stationary; however, this is a double-edged sword in that blindly feeding data into a model can achieve poor results.

With the rapid development of soft-compute forecasting models, it is important to be able to accurately characterize the performance. As such, many published works leverage open datasets. Although there are inconsistencies as to which datasets are included in the published results, in general they include a variety of tasks and an adequate number of comparison models. An important observation about these datasets is that they are all relatively small in size compared to the datasets used in this work. Table 2.1 contains a summary of some commonly used datasets. Traffic is the largest benchmark data set with more than 15-million data points. Deep learning models perform better with larger datasets, whereas statistical models may perform better relative deep learning models on sparser data sets. Data set size requirements for *good* time series forecasting remain to be studied; however, it can be inferred from the results of existing time series forecasting studies that more data is generally better.

Dataset	Number of Observations
ETTh1*	17,420
ETTh2*	17,420
ETTm1*	69,680
ETTm2*	69,680
Electricity*	26,304
Exchange-Rate†	60,704
Weather†	1,106,595
ILI†	6,762
TrafficL†	15,122,928
ECL†	8,443,584

Table 2.1: Common benchmark time series datasets. ∗ indicates results taken from [1] while † indicates results taken from [2].

2.3.6.1 Forecast Generation

Before introducing and discussing different forecasting models, it is important to understand the ways in which forecasts are generated. In general, there are two ways to generate a forecast. The typical forecast generation is either DMS or IMS. In simple terms, is a multistep forecast built one step at a time or is it all created in one iteration? There are benefits and drawbacks to either forecasting method. The recursive nature of IMS forecasts may lead to a higher time complexity.

When using autoregressive models, which are further discussed in Section 2.3.6.2, it is important to correctly configure the model. Incorrect specification can lead to bias when using IMS forecasting techniques. If the optimal model specification is unknown, a DMS forecast will always achieve better performance than IMS; however, if the model is correctly specified, then the performance is equal. For AR models, it has ultimately been suggested to test both options and select the better performing strategy [53]. Comparing the two methods it can be intuited that the direct strategy will avoid accumulation of error from recursion; however, the direct strategy may not learn how the prior forecasted steps are related to the current target step.

2.3.6.2 ARIMA

A family of statistical models that has stood the test of time is ARIMA. This family of models is a generalized form of three other types of model from which it is named: autoregressive (AR), integrated (I), and moving average (MA) models.

When explaining the configuration of an ARIMA model, it is common to refer to it by the order of the individual components. This is commonly written as

$$\text{ARIMA}(p, d, q) \tag{2.21}$$

where p is the number of AR terms, d is the difference order, and q is the number of MA terms.

The AR component of this family of models is a linear combination of the lags of the target variable. This component can be expressed as

$$\text{ARIMA}(p, 0, 0) = \sum_{i=1}^{p} \phi_i \; y_{t-i} + b_0 + \eta_t, \tag{2.22}$$

where ϕ_i is the weight of the regressor for the i^{th} time delay, b_0 is a constant and $\eta_t \sim \mathcal{N}(0, 1)$.

The MA component takes a similar form as the AR components; however, the MA terms involve the noise terms rather than the previous values of y. It can be written as

$$\text{ARIMA}(0, 0, q) = \sum_{i=1}^{q} \theta_i \; \eta_{t-i} + b_0 + \eta_t, \tag{2.23}$$

where θ_i is the weight of the regressor for the i^{th} delayed noise term.

It is common to find AR and MA terms considered without differencing. It is a category of models called ARMA. It is ultimately a combination of Eqn. 2.22 and 2.23 without duplicating the noise and constant terms. ARMA is expressed as

$$\text{ARIMA}(p, 0, q) = \left(\sum_{i=1}^{p} \phi_i \; y_{t-i} \right) + \left(\sum_{i=1}^{q} \theta_i \eta_{t-i} \right) + b_0 + \eta_t. \tag{2.24}$$

From the definition of an ARMA model, it is trivial to arrive at an

ARIMA model. Before fitting the model described in Eqn. 2.24 the input sequence will undergo d discrete derivatives. This results in one data point being lost due to the fence post problem; however, it is not ultimately a problem in a practical sense. Taking the derivative of the input signal can help convert nonstationary signals into stationary ones.

References for AR, MA, ARMA and ARIMA models are loosely based on [51, 38] with adjustments made for notation.

There exist further extensions of ARIMA, such as SARIMA, ARIMAX (ARIMA with eXplanatory variables), and SARIMAX; however, they are ultimately extensions of the basic models outlined in Eqn. 2.24. SARIMA extends ARIMA by adding lagged terms with some specified look-back period. Instead of lags taking the i^{th} delay for the AR and MA terms outlined in Eqn. 2.24, the lags of the AR component will become $t - iS \in [0, P - 1]$ and $t - iS \in [0, Q - 1]$ where P and Q are the SARIMA AR and MR orders, and S is the seasonality period.

2.3.6.3 Tree-based Regressors

XGBoost [54], random forest (RF), and other tree-based methods have gained popularity in many tasks such as classification, regression, and time series forecasting. Tree-based techniques are known to outperform many neural network approaches in tabular data. This is largely attributed to their ability to extract important features, while not smoothing out the output space [55]. Although these findings strongly support the use of tree-based methods over neural networks for tabular data, time series forecasting in the context of short-term LTE forecasting has found contradictory results in which RF was outperformed by all evaluated neural network models [37].

Although not discussed further in this work, tree-based methods achieve poor time complexity relative to many of the neural network-based methods discussed in later sections. The time complexity of XGBoost is known to be $\mathcal{O}\left(Kdx + x\ \log\left(B\right)\right)$ where x is the number of inputs, d is the maximum tree depth, K is the total number of trees, and B is the number of rows in each block. This is objectively worse than models such as NLinear and DLinear which will be introduced in Section 2.3.6.6. Although time complexity is not the be all end all of model efficiency, the scaling and performance of tree-based methods for time series forecasting quickly removes them from contention in high-throughput forecasting scenarios such as short-term forecasting.

2.3.6.4 NN, RNN, LSTM, and GRU

Neural networks (NN) have become a topic of great interest. Although the concept of neural networks has been researched for many years, their aggressive adoption for ML tasks can be largely attributed to the availability of compute power. Generally, the NN must grow to be larger in size to perform more complex operations. Small networks have small learning capacities, while larger networks have larger learning capacities. More compute power in the form of graphics processing units (GPUs) was one of the keys to the wide-scale adoption of NNs, since they allowed networks to grow in complexity to address more complicated problems. More computing power with infinitely scaling network sizes is one way to address complex problems; however, it is not always efficient or the best approach to do so.

At its core, an NN is a set of neurons connected in predefined patterns with an activation function at the output of each neuron. The structure of a neuron is similar to that of its biological counterpart. An example of a single neuron with three inputs and one output can be found in Fig. 2.5.

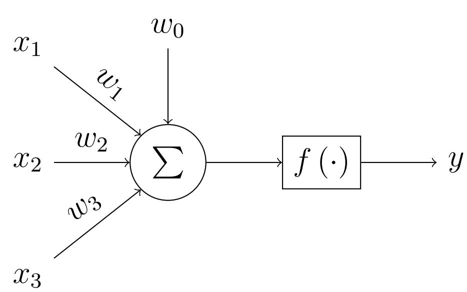

Figure 2.5: Example of a single neuron from a NN. Inputs x are multiplied by weights w and summed. Following the sum they are passed through an activation function, $f(\cdot)$, which produces the output of the neuron.

The equation to describe the operation of the neuron shown in Fig. 2.5 can be written as

$$y = f\left(wx^T + w_0\right), \tag{2.25}$$

where x is the $a \times b$ input vector, w is the $a \times b$ weight vector, $f(\cdot)$ is the activation function, w_0 is the bias weight, and y is the neuron output. There are many different approaches to selecting an activation function; however, for now it is abstracted in favor of only requiring the function itself to be nonlinear.

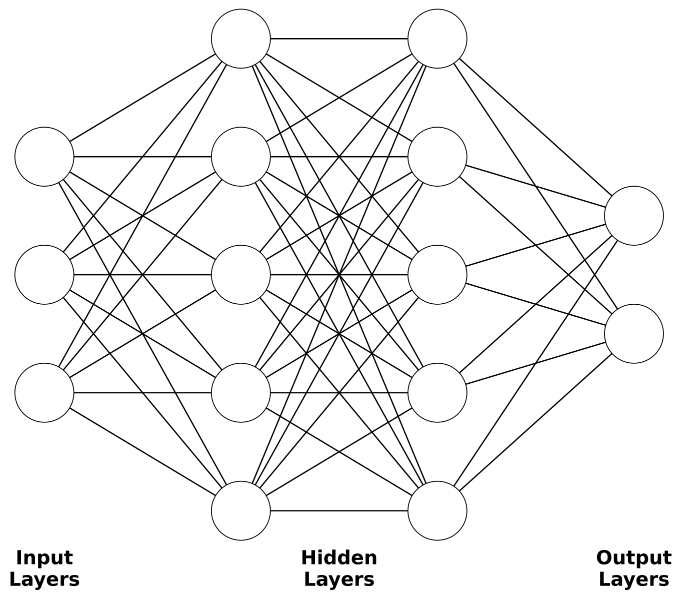

Figure 2.6: Inter-cell connections for a fully connected NN with 4 layers.

To increase the complexity and learning capacity of a NN, the aforementioned cells are stacked to create depth and breadth for a network. By abstracting mathematical notation in favor of structure, a fully connected (FC) NN can be seen in Fig. 2.6. For an FC NN, information is passed in, propagated through each neuron pathway, to obtain some output value. Although FC NN can learn to approximate functions of arbitrary complexity, they do not have any awareness of the structure of the input signals and consider all inputs to the network concurrently.

Recurrent neural networks (RNNs) have been proposed as a way to capture temporal dynamics within NNs. By adding feedback mechanisms to a FC NN the network becomes able to learn sequences of data with temporal dependencies. Furthermore, feedback connections allow RNNs to be smaller than an equivalent FC NN [56]. In the current literature, *RNN* is often used interchangeably with Elman RNNs (ERNNs) [57]. One of the most popular NN libraries, PyTorch [58], goes so far as to name its implementation of ERNNs *RNN*. ERNNs, hereby referred to as RNNs, gain recurrence by copying their hidden state and feeding a delayed version of it back into the neuron.

Although RNNs can provide a more compact form with better temporal learning capabilities than FC NNs, their feedback mechanism does introduce problematic behaviors while training the network: exploding and vanishing gradients. RNNs share similarities with infinite impulse response (IIR) filters; however, nonlinear activation functions ultimately cause a divergence in behavior [59]. Training RNNs on long time dependencies is where vanishing and exploding gradients typically occur. Exploding gradients are caused by the RNN equivalent of a resonating IIR filter, whereas vanishing gradients are the result of exponential decay of the signal over time [60, 61]. RNNs provide a way to learn temporal dependencies, but introduce many problems with their recurrence.

Long short-term memory (LSTM) networks were proposed as a solution to exploding and vanishing gradients [62]. LSTMs are an extension of

RNNs; however, they add learnable gating mechanisms that allow for constant error flow during weight updates. These take the form of input, output, and forget gates. The input to the cell and the previous hidden state are fed to each of these gates. Each gate applies its own set of weights and biases before applying the sigmoid function. A key difference between LSTMs and RNNs is that they contain internal memory, which is used to ensure uniform gradients across all timesteps. There is also a gating mechanism used to control how information is allowed to impact the memory cell, the forget gate is used to control how much the current memory cell will be carried into the future, while the output gate controls the flow of information out of the cell.

More formally, the operation of the LSTM can be described as

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}), \tag{2.26}$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}), \tag{2.27}$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}), \tag{2.28}$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}), \tag{2.29}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{2.30}$$

$$h_t = o_t \odot \tanh(c_t), \tag{2.31}$$

where subscript t indicates a discrete time step, i_t is the input gate, f_t is the forget gate, g_t is the cell gate, o_t is the output gate, c_t is the state of internal memory, and h_t is the hidden state. Furthermore, σ is the sigmoid function, W is the weight applied to various signals, such as the input x and the hidden state h, and b is the bias. The subscripts in the weights indicate their *connection*: $W_{source,target}$. For example, $W_{if}x_t$ is the input signal at time t multiplied by the input weights i to be passed to the forget gate f, hence W_{if} [63, 58].

Gated recurrent units (GRU) are a more recent architectural development that helps address vanishing and exploding gradients in a way similar to that of LSTM [64]. LSTMs manage the flow of information through the network using three gates and an internal memory. GRUs foregoes three gates in favor of two: the update and reset gates. Like the LSTM, the input and previous hidden states are passed to the gates, weighted, and passed through a sigmoid function. The update, or new gate, controls how much of the previous hidden state is passed on in time relative to the amount the current internal cell is. Meanwhile, the reset gate determines how much the previous hidden state is allowed to update the internal cell. The update gate of GRU is similar to the input gate of the LSTM, while the reset gate is similar to the LSTM's output gate. In a way, GRU effectively merges the

LSTM's forget gate into the reset gate.

GRU operation can be described as

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}), \tag{2.32}$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}), \tag{2.33}$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t * (W_{hn}h_{(t-1)} + b_{hn})), \tag{2.34}$$

$$h_t = (1 - z_t) * n_t + z_t * h_{(t-1)}, \tag{2.35}$$

where r_t is the reset gate, z_t is the update gate, and n_t is the new gate [64, 58].

2.3.6.5 Seq2Seq

Development of RNN, LSTM, and GRU is roughly spread over the 1980s, 1990s, and 2010s, respectively. A reoccurring theme across their respective foundational papers is the discussion of varied-length sequences. FC NNs are restricted to fixed-size inputs and outputs, while the RNN family of models is not. Tasks such as translation, speech analysis, and time series forecasting can result in varied length sequences that come into and out of a network. Sequence-to-sequence (Seq2Seq) models have been proposed as a way to map input sequences to output sequences [65]. A surface-level view of Seq2Seq models may have them appear to simply be another RNN-family model; however, their translation from the input to output sequence is remarkably different.

Seq2Seq models are a form of the encoder-decoder architecture, where the encoder transfers the last hidden state to the decoder network. This type of model was originally presented for the purpose of neural translation, with the original work encoding English to decode French. This segmentation of the translation process allows for an easier discernment of how Seq2Seq differs from RNN: RNN considers the current and past input values to produce an output, while Seq2Seq can consider all input times before generating the output. Although Seq2Seq generally outperforms the RNN, LSTM, and GRU models in a variety of tasks, it is important to note that the context vector can be a significant performance bottleneck in terms of accuracy and computational efficiency. If an insufficiently large context vector was used, then information would be lost when the signal is encoded to it, thereby meaning that information will be lost on the reconstruction by the decoder. Furthermore, since Seq2Seq models are built using RNNs, they will also suffer from recurrent computational inefficiency.

2.3.6.6 Transformers and Other DL Models

Recurrent connections in Seq2Seq and other RNN-based models allow them to handle arbitrary input and output sequence lengths. Although recurrence is a powerful way to handle time-dependent data, it introduces problematic performance issues: the time and memory complexity of RNNs is $\mathcal{O}(T)$.

In recent years, a significant amount of work has been done to alleviate the performance bottleneck of recurrence. In quick succession, an entire family of models for sequential data has rapidly made progress. A keystone architecture in the handling of sequential data is the Transformer [66]. Transformers forego recurrent and convolution mechanisms in favor of attention mechanisms. The overall structure of Transformers is an encoder-decoder network with additional peripherals such as positional encoding and attention. Attention mechanisms are used throughout the transformer to allow all inputs to be attended to. Transformers themselves allow unknown input sequence lengths to be consumed to produce output sequences of unknown length without the use of RNNs. Although they achieved state-of-the-art neural translation results at the time, transformers ultimately have $\mathcal{O}(T^2)$ time and memory complexity for sequence length T, which is problematic for model scaling and performance [1].

In general, the encoder-decoder and Seq2Seq models achieve excellent results in time series forecasting tasks; however, they quickly run into performance bottlenecks. Following the introduction of transformers, significant efforts were made to improve the scalability of their attention mechanisms. Reformer introduced locality-sensitive hashing to reduce time and memory complexity from $\mathcal{O}(T^2)$ to $\mathcal{O}(T \log T)$ [67]. In the LogTrans transformer, LogSparse self-attention was proposed as a way to reduce complexity by using exponential step sizes between connections with sufficient layers to ensure that each cell eventually is exposed to the input of all previous cells [68].

The efficient scaling of attention in transformers remains a problematic area; however, attention can provide valuable insight into the operation of the architecture. The temporal fusion transformer (TFT) leveraged the scaled-dot product attention from the original transformer architecture [66] and employed additive aggregation for all heads [40]. The outcome of using additive attention aggregation was readily interpretable attention. Explainable results are important for some forecasting applications, but again, TFTs experience high complexity due to the attention mechanisms.

When comparing the results of many popular transformer models in benchmark datasets, the results are rather bleak: ARIMA beats LogTrans and Reformer in all datasets. Informer [44] and Autoformer [69] failed to outperform ARIMA in all datasets tested [1]. Significant strides are being made with transformers, but their staggering complexity has led to an important question asked by an aptly named paper, *Are Transformers Effective for Time Series Forecasting?* [2]. Zhu et al. proposed that transformer models benefit from being an architecture that directly generates multi-step forecasts rather than ARIMA and RNNs which generate multi-steps autogressively. Two models were proposed, DLinear and NLinear. DLinear performs time series decomposition before passing the data through a single linear layer, while NLinear subtracts the first input signal before passing through a linear layer, after which this offset is added back. It should be noted that both of these models are a linear model with a small amount of additional architecture before it. The results of DLinear and NLinear strongly support the bold questioning of transformers for time series forecasting, as they outperformed all baseline transformers while having a time complexity of $\mathcal{O}(1)$ [2].

Although transformer-based models have shown promising results despite their high complexity, time series forecasting architecture research is simultaneously investigating transformer-free forecasting models. The aforementioned DLinear and NLinear fall into this category as well as models such as N-BEATS (neural basis expansion analysis for interpretable time series forecasting) and N-HiTS (neural hierarchical interpolation for time series). N-BEATS and N-HiTS differ from transformer models in that they are pure DL solutions. They do not contain convolution or attention mechanisms, but they still achieve excellent results in benchmark data sets [1, 43].

N-BEATS and N-HiTS are interesting models in that they employ an architecture not unlike successive interference cancelation (SIC). In the context of multiple-access wireless communication networks, SIC enables the recovery of two symbols that arrive at the same time. This is done by extracting the strongest symbol, removing the effects of said symbol on the received signal, and then repeating this procedure until all symbols are demodulated. N-BEATS and N-HiTS follow a similar process using stacked FC NNs that generate a forecast (similar to SIC's decoded symbol) and a backcast (similar to SIC's demodulated symbol removal). These models have been reported to be more accurate than all the transformer models evaluated and ARIMA [1]. This result is particularly impressive, since both models are less computationally complex than the baseline transformers.

DLinear and NLinear are paradigm shifting models for time series forecasting [2]. Although some transformer-based architectures fail to outperform baseline ARIMA models, this is mainly on smaller datasets. N-HiTS outperformed ARIMA across all tested datasets; however, the spread increased as the size of the data sets increased. Cross-referencing the result of N-HiTS [1] and DLinear/NLinear [2] it can also be seen that the Linear models outperformed ARIMA as well. The Linear models are aptly named: the model is a linear network connecting all inputs to all outputs. It can produce an arbitrary number of timesteps and has a depth of 1. The DLinear variant includes a smoothing kernel to split the inputs into trend and seasonal components, while NLinear subtracts the first value from the input sequence and adds it back after prediction. Regardless of the variant chosen, they are ultimately linear models with some peripheral components.

The Linear family of models is of special interest not only for their reduced computational complexity but also for their composition. ARIMA models, specifically AR components, leverage a linear combination of an arbitrary number of historical time lags. Using these forecasts, an arbitrary number of steps can be generated in the future. The Linear models are similar in that they are a linear combination; however, now the model is able to attend to all inputs and how they relate to all output time steps. This is readily visualized as a result of the simplicity of the model. A visualization of the weights of a DLinear model can be seen in Fig. 2.7. The input data were roughly periodic, with three periods being used to predict one output period. This periodicity is clearly reflected in the weights as prominent diagonal components. Some of the aforementioned models, notably N-HiTS and N-BEATS, do provide a level of explainability, but it is not as easily extracted and interpreted as the Linear models.

The simple nature of the Linear models allows their architecture to be described as

$$\hat{y}_{i,t} = \sum_{i \in I} \sum_{t \in T} W_{i,t} x_{i,t} + b_{i,t}, \tag{2.36}$$

where I and T are the input signals and timesteps, respectively. This is not unlike AR models where the history is used to regress the output; however, now all inputs are used to predict all outputs. Furthermore, it simply creates a linear regression of all inputs to all outputs, providing a direct multistep approach.

2.4 Anomaly Detection

Monitoring an ever-growing number of components of the network infrastructure is no simple task. Simple equipment state checks are often sufficient for small-scale systems; however, considering nation-wide networks requires a change in how monitoring is performed. For cellular networks, there are a significant amount of built-in monitoring capabilities; however, they are often primitive in nature.

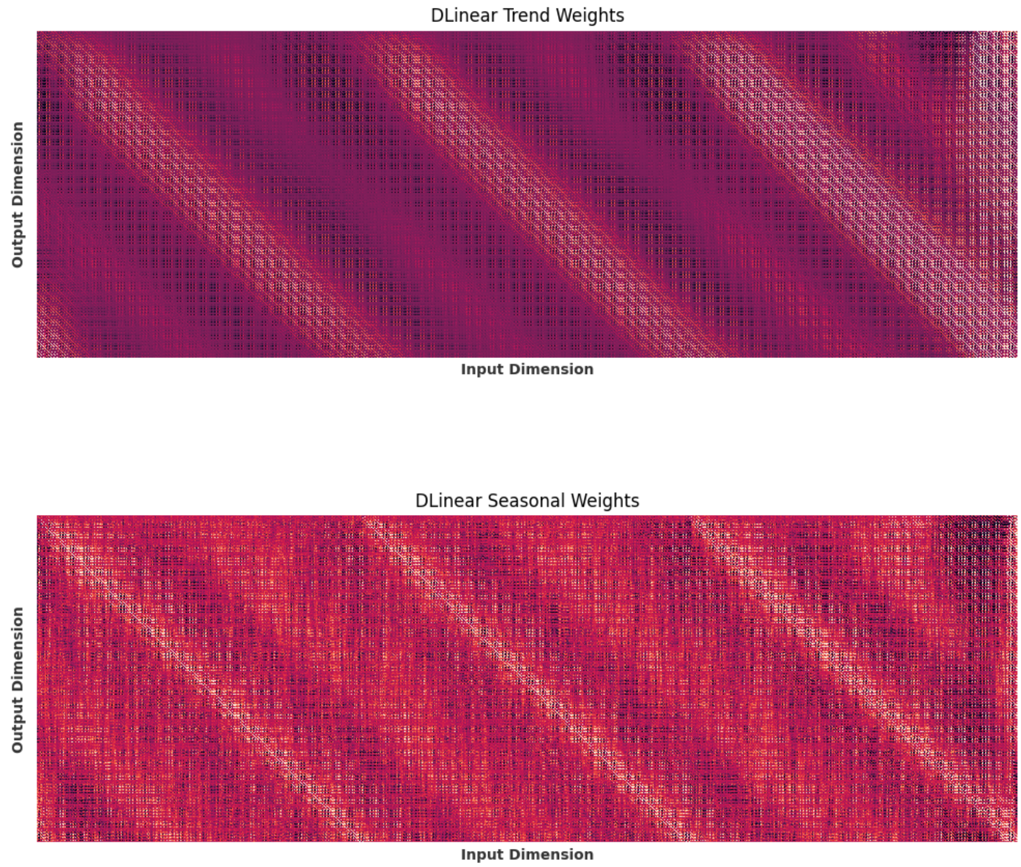

Figure 2.7: DLinear weights extracted from a model trained to predict 5 signals with a horizon of 96 samples based on an input of 288 samples. The input data had a sampling frequency of 15-minutes and had a strong daily seasonality. The three strong diagonal lines in the weights demonstrate the model learning the daily seasonality.

In general terms, the process of anomaly detection (AD) is a classification task. There are ways to determine the intensity of a detected anomaly; however, in the end, the question is whether the sample is anomalous in nature. Although it is a classification task, the underlying techniques for determining this do not need to be directly classification-based. Some form of analysis could be performed, and then a classification could be performed on that new analysis. This has been highlighted in deep learning AD approaches [70]. By taking a modular approach to AD, components can readily be exchanged for better performing ones as time goes on.

For time series AD, the objective is to process a stream of samples and identify if there is a fault. A fault could present as a handful of samples being abnormal or as a total change in temporal properties. A sudden power outage may cause a large number of people to lose internet connection, which would appear as a sudden drop in traffic and connected users, which would be a few samples that are present as anomalous in nature. Similarly, a user jumping between cells erratically could present as a shift in temporal properties, which could be an issue.

The approaches to handle AD tasks can be broken down into roughly five categories: statistical, classification, clustering, soft computing, and knowledge-based [71]. Hybrid methods are also quite popular, where the AD method is some combination of these five methods. Hybrid and ensemble methods have become increasingly popular, as combinations of each of these methods can be used to reduce the potential issues of any individual method. Furthermore, the nature of the data being analyzed can further constrain the choice of method.

Although AD is a classification task, it is not always a supervised task. That is, clean and accurately labeled data are often not available or require a significant cost to obtain. This is another constraint that can be used to select an AD method for a given task: are labeled data available, and if so, how much? If labeled data is available that adequately represents the distribution of possible outcomes, a supervised approach can be taken. If some labels are available but not a significant enough number for good supervised results, a semi-supervised method could be used. Finally, if no labels are known, an unsupervised method must be used.

2.4.1 Supervision

Machine learning and statistical modeling are almost always limited by the quality or quantity of data provided. It is generally difficult to generate data periodically without missing time steps. Furthermore, accurate metadata for each entity that generates a time series can be difficult to obtain due to configuration changes. The increasing number of IoT devices and intelligent infrastructure causes the volume of data generated to be larger than ever across many domains; wireless communication networks are no exception to this observation. Across most domains, the volume of data produced exceeds what can be labeled by a domain expert.

The presence of labels associated with a data set will ultimately limit the selection of AD methods. If labels are present and there are sufficient numbers of them for a given AD method, then a supervised learning method can be used. This type of approach will create a model that will learn to map the input sample space to a set of labels. An error metric can be used to provide feedback to the prediction model, which will then hopefully converge to a good solution assuming that hyperparameters are properly set and training data are sufficiently rich to describe the sample space.

Due to the volume of data produced by infrastructure monitoring, it is rare that a significant amount of data can be labeled. Furthermore, it is unlikely that the data will be consistently labeled and that all types and occurrences of anomalous events will be found. There are approaches that can leverage a data set with which a subset has been labeled, which are known as semi-supervised methods [72]. Like supervised methods, semi-supervision is susceptible to limitations in the labels. If the labels do not adequately describe the full range of anomalies, then they are unlikely to result in a quality AD model.

Unsupervised learning has received significant attention in recent years because a model can be created without the need for labeled data. Since labeling data is an arduous and difficult task, this brings relief to practical applications of classification in big data applications. Without labels to guide the model, information must be extracted from the set of samples. This can be done using a few different techniques, such as reconstruction, forecasting, distance, distribution, or encoding techniques [73]. Although unsupervised classification can be broken down into the categories mentioned above, it can be further compressed into just two categories. Reconstruction, forecasting, and encoding techniques transform the sample space into an alternate one. Distance and distribution methods are related in that they consider the sample space directly.

The ability to create AD models without labeled data is promising; however, it raises the question of how to quantify the performance of a model without any tangible definition of what is anomalous? Receiver operating characteristic (ROC) and precision recall (PR) curves are often used to quantify the performance of a supervised classification algorithm. For unsupervised methods without labels, they are undefined. The consideration of excess mass (EM) and mass volume (MV) has been proposed as a way to characterize the performance of an unsupervised classification model. EM and MV have been shown to emulate the results of the ROC and PR curves. When characterizing the performance of the unsupervised AD method, if EM and MV agree on which model is the best performing, it is probably the best method. If they are not in agreement, there may not be a definitive best method of AD [74].

2.4.2 Statistical

Statistical approaches for AD are extremely popular due to their ease of use and generally explainable nature. The complexity of statistical AD methods can range from a z-score threshold test to a complex distribution analysis. Z-score AD is straightforward and one of the simplest methods to implement. For a given set of samples, the z-score can be expressed as

$$z_i = \frac{x_i - \mu}{\sigma}, \tag{2.37}$$

where z_i is the z-score for the i^{th} sample, x_i is the sample value, μ is the mean of the set and σ is the standard deviation of the set. From the z-scores a threshold can be set to determine which level of 'abnormality' is allowed in the sample distribution. For a time series implementation, the data are generally windows to some amount of time such as a week, a month, or a year rather than considering the entire available history. Selecting a threshold is not always a straightforward process. How significant an event is considered anomalous? What if a set is highly volatile? What if a set is nonstationary and the history window is large enough for shifts in mean or variance to be significant? Despite these drawbacks, statistical AD is powerful in that it does not require labeled data.

AD using z-score analysis can be adequate for very simple univariate AD tasks; however, it does not explicitly consider the distribution of the samples. Z-score analysis is excellent when the data are normally distributed, but in many cases, nonideal sets of samples are being considered. The sample distribution distance analysis has been proposed as a way to improve statistical AD [75]; however, these methods do not consider the temporal properties of the sets.

Time series signals with a periodic component will have some form of structure. Fig. 2.3 shows an excellent example of how the data can have periodicity. There is noise present; however, ultimately the signal is periodic in nature. Distribution analysis loses this structure. Fig. 2.8 provides an example of the density estimation for time series considered in Fig. 2.3. The probability density function was estimated by day. The time series representation of the signal clearly shows that the magnitude was larger for the first day's worth of data. After the first day, the data maintains the same periodic structure; however, the amplitude is generally lower. When considering the density estimates, this intuition is lost. A large amount of samples are present around 90% for the first (0^{th}) day. If only the density were considered for an AD method, it would clearly be a fault; however, in reality it is simply a day with increased use and likely does not merit flagging the day as anomalous.

For AD tasks related to events, distribution analysis is often powerful. For example, the time it takes for a server to produce a webpage could be analyzed. There will always be some mean and associated variance. Changes in the shape or properties of the distribution could be easily identified. As was previously stated, distribution analysis for time series is not always beneficial, especially as the dimensionality of the problem increases. The selected example presented in Fig. 2.8 is a univariate problem. When considering more time series features simultaneously, it becomes unwieldy to perform distribution analysis. Furthermore, the interpretability of the detection process can rapidly be lost as the number of dimensions increases.

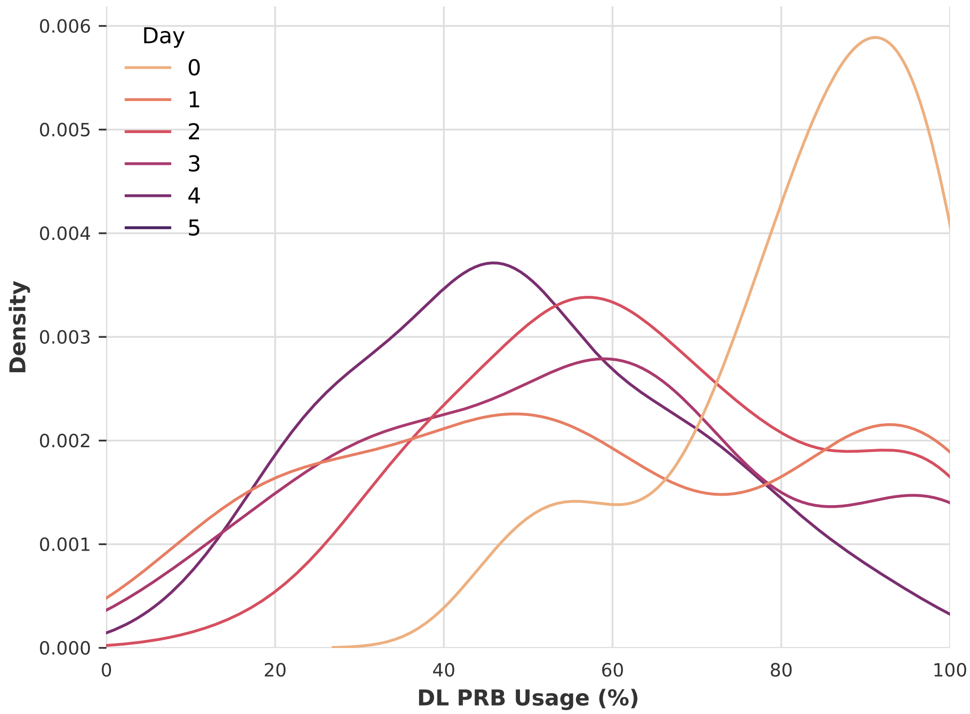

Figure 2.8: Daily probability density estimation for the raw time series data presented in Fig. 2.3

2.4.3 Clustering

Multi-dimensional datasets are often problematic for many AD methods; however, clustering methods can excel at providing a highly interpretable method. Methods like k-nearest neighbors (k-NN) and density-based spatial clustering of applications with noise (DBSCAN) have become immensely popular for the exploration and clustering of multidimensional data. Under the assumption that anomalies are statistically improbable, some clustering methods can be readily used to identify anomalies. If a sample is near its neighbors, then it is likely not an anomalous sample. The converse of this is also valid; if a sample is far from its neighbors, it is likely anomalous.

One of the biggest benefits of using clustering methods for AD is that they are generally unsupervised classification algorithms. Given a set of samples with an arbitrary number of dimensions, without labels or a priori knowledge, the structure of the data can be estimated using a clustering method. At a high level, the objective is to take samples and subdivide them into groups of similar samples. This is generally done using centroid-based algorithms, such as k-NN or density-based methods, such as DBSCAN.

At a high level, k-NN, specifically, k-means clustering, will create a predetermined number of clusters. Each cluster will have an associated centroid. The proximity of these points to the centroid is then used to move samples between groups until some exit condition is reached [76]. On the other hand, DBSCAN iterates through the samples and finds neighbors based on the radial limits of each sample. If there are enough samples in a region, then it is considered a cluster. If a point is not part of a cluster, it is identified as noise [77].

From these brief summaries, the biggest difference between the two algorithms becomes apparent: k-means require a number of clusters to be known, DBSCAN requires the selection of a point density value, and k-means does not have limitations on how far away a point can be from the centroid. In other words, k-means begin with the preconceived notion of what clusters look like, whereas DBSCAN does not make this assumption. DBSCAN requires parameter tuning in terms of the number of points required to be considered a cluster and the distance required for the points to be neighbors. In general, k-means are tuned by using the number of clusters.

Another important difference is that DBSCAN has the ability to declare a sample as anomalous. If a point is sufficiently far from its neighbors, it will not be assigned to a cluster and will be labeled as noise. In the context of AD, this is an extremely powerful behavior. In contrast, k-means do not inherently have the ability to reject points and often require extensive post-processing to identify an anomalous sample.

A visualization comparing the clustering methods in an AD context can be seen in Fig. 2.9. The figure compares the amount of downlink physical resource blocks (DL-PRB) being used for a given downlink throughput (DL-THRP) for a cell. In a sense, this is an approximation of the bandwidth normalized channel capacity described in Eq. 2.2. The relationship between these KPIs is expected to be strongly correlated; increase in DL-PRB will result in an increase in DL-THRP. k-means clearly do not identify any anomalous behavior of the samples. There are samples that are clearly distant from their neighbors; however, they are still clustered with no anomalous points identified by k-means. On the contrary, DBSCAN with a very small amount of parameter tuning was able to identify anomalous points by finding low-density regions.

The use of k-means as part of a hybrid AD method in the wireless network has been proposed [15]; however, the inability of k-means to reject

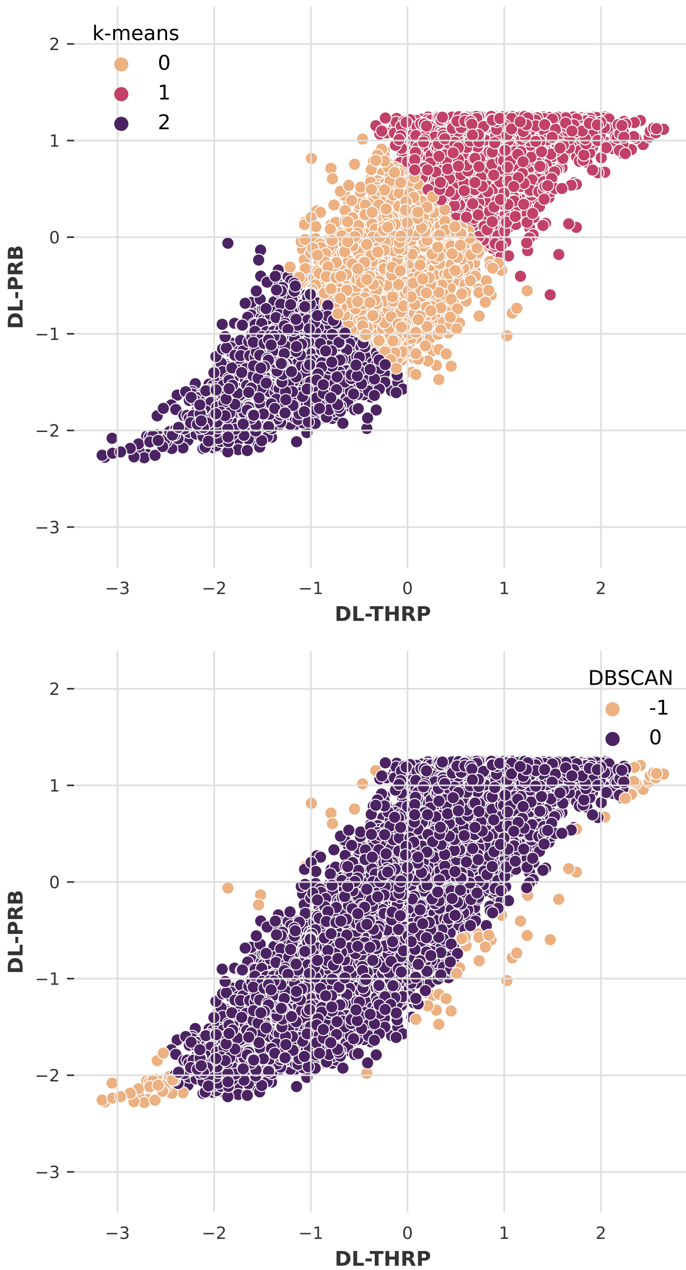

Figure 2.9: Comparison of k-means and DBSCAN for clustering-based AD using standardized KPI values.

the classification of anomalous points is problematic. More specifically, this is similar to picking one of the k-means clusters in Fig. 2.9 and arbitrarily deciding that it is anomalous.

Although DBSCAN and k-means are not the only clustering methods that could be considered, their performant nature makes them well suited for big data AD tasks. DBSCAN and k-means have been shown to outperform many other clustering algorithms in synthetic data sets [78]. In many big data applications, the computational deadlines are not always very strict; however, in short-term time series AD, the deadlines can be very strict. Alternative clustering algorithms can provide better results for anomaly detection, such as OPTICS, which allows variable density clusters, while DBSCAN does not; however, the computational cost increase associated with other algorithms will significantly increase the cost of selecting these algorithms.

2.4.4 Classification

Supervised classification has received significant attention due to its application in computer vision. Although there are semi-supervised classification methods, the vast majority of conventional classification tasks leverage labeled data. Time series AD surveys have highlighted that the rare occurrence of anomalies, along with the requirement to label the data, results in poor performance [73].

2.4.5 Soft Computing

AD using soft computing covers a broad number of methods. Neural networks, genetic algorithms, and fuzzy logic are examples of soft computing; however, neural network approaches, specifically deep learning approaches, have gained significant traction for AD [79].

Deep learning approaches generally take one of two approaches; reconstruction or forecasting [73]. Both approaches are commonly used, as they do not require the labeling of input data, which is often sparse in large data sets [79]. Depending on the architecture of the model chosen, reconstruction and forecasting can be nearly identical in procedure.

Consider some of the forecasting methods discussed in Section 2.3.6. Seq2Seq models map a time series to a latent space and then use the encoded information to create a forecast. This forecast could then be used to detect anomalies by looking at the distribution of the error or the confidence interval of the forecast. Similarly, a reconstruction procedure could map an input series to a latent space and extract information from the latent space; however, the target would be the input data, rather than the next time step. In broad terms, forecasting will map input space to future time steps, while reconstruction will map the input space back to the input space. For both approaches, it is assumed that the number of anomalous samples is much less than the number of normal samples. This allows for a model to be trained across the entire data set and will learn a representation of normal samples, while the sparse anomalous samples will have minimal impact on the learned representation. Both of these methods are incredibly powerful in that they do not need labeled data. The *labels* are the input sequence or future values of the input sequence.

For AD prediction or reconstruction methods using neural networks, a wide variety of architectures can be used. Forecasting models are more flexible in their implementation since they do not need to create a latent state of the input sequence. This means that models such as DLinear and NLinear could be used for forecasting, but not for reconstruction. Some of the forecasting models would directly map the input to the output if a latent space was not used as an intermediary. Other forms of soft computing approaches exist and have been found to achieve excellent results; however, many of them require labeled data, which is not practical for network-scale AD considerations.

2.4.6 Knowledge

A primitive and inherently explainable form of AD is knowledge-based or rule-based methods. Methods in this category construct a framework that formally defines what is normal or anomalous behavior. In the context of wireless networks, rules can be as simple as *the number of users connected to a cell must be greater than 0* or more complex, leveraging multiple signals with more complex analysis.

Most alarms built into LTE and 5G networks are threshold-based; this is one of the most primitive forms of rule-based AD. An issue with threshold-based alarms is that they must be carefully set by experts. Setting thresholds across a large number of cells without careful consideration of each cell can lead to problematic results. If an alarm goes off every day, multiple times a day, then it is likely to be silenced or ignored by network administrators. For primitive alarms, this is problematic because setting an alarm so that it is not triggered as often can lead to under reporting; this could lead to the network experiencing catastrophic anomalous behavior, yet the monitoring system would be reporting everything is online and operating normally. It is well known that AD methods that use knowledge-based systems are more susceptible to over-reporting; that is, the fraction of samples flagged as anomalous is typically much higher than the actual value [71].

To better address the reliability of threshold-based alarms, it is common for network operators to set multiple consecutive thresholds. In other words, one threshold value would indicate a minor problem, while another threshold would indicate the same problem, except with a great impact on users. Adding hierarchical information to detected anomalies can help improve the prioritization of network operator administrators in the event of multiple anomalies; however, this method is ultimately susceptible to the same problems that would occur with a single alarm. If thresholds are not optimally set cell-by-cell, the AD results are unlikely to be meaningful or properly leveraged by a network operator.

Although it is technically a hybrid method that combines knowledge and statistical methods, direct thresholds can be easily replaced with statistical ones. Z-scores can be easily used to identify how unlikely a sample will be given the input distribution. Experts can then select how statistically anomalous a sample must be to be flagged as problematic. In many cases, this method requires less administrative effort, since the thresholds can be set without considering cell-by-cell performance due to the standardization that is inherently part of the z-score calculation.

Despite typically underperforming clustering and soft-compute methods, rule-based approaches are beneficial as they are explainable by nature. Knowing that an anomaly has occurred is important; however, being able to detect an anomaly and know its likely root cause from a single AD method is very powerful from an administrative standpoint. Moreover, this family of methods is often very computationally efficient. Clustering and soft-compute methods generally require extensive setup and computation before being able to detect an anomaly, whereas knowledge-based methods generally are a framework built of simple calculations and threshold analysis.

Although rule-based AD systems are built into modern network infrastructure, extensions of these frameworks are common. Furthermore, rule-based methods are often the *glue* used to join AD methods in hybrid frameworks.

Chapter 3

Seasonal Properties of Rural Fixed Wireless LTE Networks

The relationship between seasonality, user behavior, and environmental factors is critical for understanding what drives network performance indicators. People are known to behave differently depending on where they live and what the weather is like [80, 81, 6]. Furthermore, seasonal events driving population behavior, such as the academic school year and summer holidays, can create a user-driven seasonal change. From an environmental standpoint, significant changes can also be introduced. In Canada, it is not uncommon for temperatures to vary from -30 to +30 degrees Celsius in a given area. Although some studies have considered the temporal changes in a network over the course of a few weeks with good sampling resolution [11], the durations of the studies are insufficiently long to capture the long-term seasonalities and are generally in environments where the temperature swings are not nearly as significant as they are in Canada. Propagation studies have been performed in rural regions that characterize propagation characteristics over a long-time span; however, the number of samples is small, often only a handful of data points over the duration of the study. This chapter will aim to address the lack of temporal duration or the lack of samples by considering nearly two years worth of data with a higher sampling frequency than existing studies.

One of the most difficult aspects of considering long-term network scale seasonality is deciding which performance indicators to monitor. Since thousands of metrics are collected every 15 minutes, distilling the number of KPIs into a robust subset to describe overall performance is critical. To characterize the performance of the network, a set of seven network KPIs was selected. An overview of the selected KPIs can be found in Table 3.1. The overarching objective of the selected KPIs is to describe the performance of a cell with a small number of KPIs. Spectral efficiency is the practical realization of channel capacity. Since this is the upper bound on the communication rates for a given channel, KPIs are selected to allow this to be described: physical resource block (PRB) and throughput (THRP) for both UL and DL. These metrics can be combined to create an approximation of how efficiently the spectrum is being used. AUE is considered since it will allow the number of users on a given cell to be tracked over time. In regions where habitation is primarily seasonal, for example, cottages or summer homes, it is important to be able to relate potential dips in DL-TV and DL-PRB back to what may have been a reduction in the number of users since they have suspended their service while they are not at that location. Finally, UL interference is tracked as changes in the propagation environment may see changes in measured interference as foliage, snow, and other temporal phenomena occur.

KPI	Abbreviation	Units	Comment
Active Users	AUE	Users	Number of users connected to a cell and sharing the cell resources
Downlink Physical Resource Block Use	DL-PRB	%	Percentage of downlink time-frequency resources in use
Downlink Throughput	DL-THRP	Mbps	Total downlink traffic speed across all users
Uplink Physical Resource Block Use	UL-PRB	%	Percentage of uplink time-frequency resources in use
Uplink Throughput	UL-THRP	Mbps	Total downlink traffic speed across all users
Downlink Traffic Volume	DL-TV	GB	Total downlink traffic volume across all users
Uplink Interference	UL-INTRF	dBm	Interference measured at the cell

Table 3.1: Overview of network KPIs used to characterize the seasonal variations of rural fixed wireless networks.

Data related to the environment is much sparser compared to the thousands of metrics offered by the cellular network. Environment Canada is a government agency that provides current and historical weather data. Weather data is available for public use and cover the whole country with reasonable spatial resolution. Throughout Canada there are 1552 weather stations. Data collection is carried out at these weather stations every hour, day, or month. These metrics describe meteorological conditions such as temperature, snow, rain, humidity, and wind speed. Of the aforementioned metrics, the focus will be on temperature and snow, since they can be more directly tied to the network's performance.

Once the relationships between network and environmental features have been characterized, forecasting models were created that attempt to leverage these relationships to improve forecast accuracy. Establishing the existence of a relationship is important; however, it is arguable that the exploitation of identified relationships is equally important. In other words, if a relationship is identified but has limited use in time series analysis and forecasting due to the majority of the information being self-contained in the signal of interest, then the relationship can be ignored. Forecasting models will consider a variety of neural network based forecasting architectures. Several IMS and DMS forecasting architectures will be used.

3.1 Related Studies

The unique nature of the data collected for this study means that there are few or no comparable studies. Propagation-focused studies have considered rural environments specifically; however, network-focused studies often consider only mobile networks.

One of the primary concerns of a commercial fixed wireless network is consistent and quality service. Broadband channels are a requirement to provide quality connectivity to end users. Government spectrum allocations typically increase as the center frequency increases. Increasing the carrier frequency in order to access greater bandwidths is a difficult optimization problem: lower carrier frequencies will have better propagation characteristics with lower bandwidths, while higher carrier frequencies will have worse propagation characteristics with higher bandwidth allocations. Spectrum allocations in the 3500 MHz region have received significant attention as they provide a good trade-off between bandwidth and coverage [9].

A research group studied the characteristics of rural propagation around Hetzwege, Germany [8, 9, 10]. These studies consider both the propagation and networking aspects of performance. The selected radio access technology was WiMAX. In general terms, the three studies consider the effects of foliage [8], the effects of antenna height and season [9], and the effects of season and vegetation in a residential environment [10]. The first two studies are related to the presented work; however, the third study focuses on propagation within a rural town, which does not necessarily have the same user distribution as this study. Furthermore, drive tests were used for these studies, which may yield characteristics different from those of a fixed wireless network.

The first study identified that the presence of foliage added an attenuation of 0.7 dB/m at 3500 MHz [8]. For long-range propagation of the order of kilometers, this is a significant level of attenuation. The second study considered the placement of antennas in terms of height and the effects of foliage. To measure the effects of foliage, a drive test was performed using the same path with and without foliage. At 3500 MHz, it was found that the mean path loss increased by 1.42 dB with a standard deviation of 4.32 dB. Furthermore, it was highlighted that popular propagation models, such as COST-231, SUI, and Hata, did not capture seasonal changes in the propagation environment, leading to significant errors in the models [9].

Snow and ice are environmental factors that can dramatically alter the propagation environment. The aforementioned studies in Germany were conducted strictly in the absence of snow [8, 9, 10]. From a modeling point of view, snow and ice are difficult to characterize. Liquid water is relatively constant in its electromagnetic properties; however, snow and ice are subjective to a wide range of conditions. The relative dielectric constant of snow can vary from similar to that of air to approximately that of water, depending on conditions such as density and *wetness* [82, 83]. The values of ϵ_{snow} generally fall between 1.1 and 2 depending on conditions [84, 85, 86]. It is important to note that these values are strictly *snow*, not ice, not snow with ice crust, or partially melted snow.

In addition to the permittivity of snow that varies significantly with environmental conditions, it is important to note that several rays are present in the reflection process. When the wave in the air collides with the snow surface, part of the wave will be reflected, while part of it is transmitted into the snow. The wave inside the snow will then collide with the media underneath it, and another reflection will occur. The interactions between radiated waves and snow on the ground are likely to interact with the ground beneath it [87].

The dielectric properties of the ground are also important to consider. For heavily forested regions, waves are unlikely to interact with the ground, since they will be largely above and partially in the canopy [8]; however, within Canada, there are regions in Canada that have little or no forests, such as the prairies, where ground and surface cover will interact directly with propagating waves. As with snow, the relative permittivity of the ground can vary widely depending on its composition. Different soil compositions have been shown to vary the relative dielectric constant from less than 3 to as high as 22 [88]. The moisture content of the soil and the presence of ground cover, such as grass, has also been shown to cause significant changes in ϵ_r. The introduction of green grass as ground cover increased ϵ_r by almost 10 at various soil moisture levels [89].

3.2 Environmental Data

As mentioned above, the Canadian government offers weather data at three sampling rates: hourly, daily, and monthly. Although many of the commonly used weather signals, such as temperature, are available across all sampling rates, snow depth and snow fall are only available at daily or monthly intervals. The prospect of analyzing the responses of the network performance to snow with a fine sampling resolution is intriguing; however, the objective of this study is to analyze long-term trends, which means that a daily resolution is adequate. LTE data was be aggregated to match this sampling rate. There are a total of 1552 weather stations reporting data during this study; however, only 1471 stations reported daily.

Of the array of available environmental characteristics, only three were ultimately selected for analysis: temperature, snow fall, and snow depth.

The temperature is highly seasonal, where summers are hot and winters are cold. The coldest months are roughly offset by six months from the warmest months, resulting in the signal being relatively sinusoidal. The average daily temperature, snow depth, and snowfall of all weather stations can be seen in Fig. 3.1.

Of the three environmental signals considered, temperature contains the most information. Although all of the environment signals provide periodic samples, the snow depth and snow fall will be zero for a large part of the year because the temperature is too high for snow. Although snow metrics have long periods of time when they are zero, the contextual information they provide is invaluable. Foliage is an important contributing factor to seasonal propagation changes. As leaves bud, emerge, and fall, they can introduce dramatic changes to the propagation environment. Foliage has been a key component of some rural propagation studies; however, it is not practical to consider in large-scale studies.

In addition to weather data, the Canadian government provides a high-level categorization of the country by environmental properties called *ecozones*. There were a total of 12 ecozones spread throughout the country [90]. For each ecozone, a general classification of terrain, forest composition, and forest coverage percentages is provided. The boundaries in ecozones are roughly defined and can span several provinces; however, they often align around the provincial borders. For example, the maritime provinces of Nova Scotia and New Brunswick are in the same ecozone. A portion of the ecozone

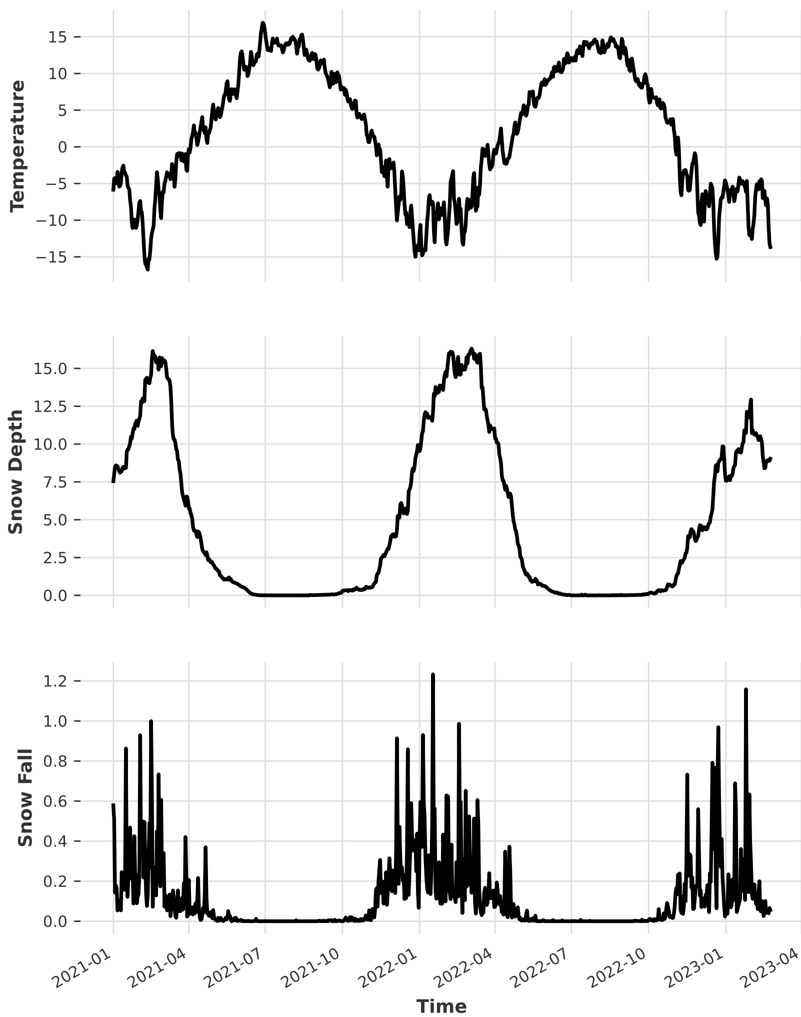

Figure 3.1: Average environmental features by day across all available weather stations. Temperature is measured in degrees Celsius, snow depth and snow fall are measured in centimeters.

extends into the western part of Quebec; however, it remains relatively close to the provincial border.

The permittivity of the medium in the propagation environment has been identified to change significantly with season. Seasonal variations will vary between ecozones. For example, 82.3% of the Atlantic Maritime ecozone is covered with forests. The forests in this region are 41.8% coniferous. In contrast, the Prairie ecozone has less than 5% forest coverage and almost all are deciduous forests [90]. Both ecozones experience loss of foliage; however, the Atlantic Maritime ecozone will likely see smaller variations, as coniferous trees retain their foliage (needles). Furthermore, the significant reduction in forest by coverage area in the Prairies could have snow introduce a much more significant change in network performance and propagation.

3.3 Network Data

There are a wide variety of metrics that can be used to characterize the seasonal performance of a wireless network. Since channel capacity is a critical performance indicator, the selection of DL-PRB, DL-THRP, UL-PRB, and UL-THRP were a natural starting point. Furthermore, UL interference (UL-INTRF) and DL traffic volume (DL-TV) are selected to describe observed electromagnetic and user behavior, respectively. Finally, the number of users connected to a cell, AUE, is important to understand the load and volume of traffic consumed per user over time.

The network data was collected every 15 or 30 minutes from January 1, 2021 to February 23, 2023. The quality of the data was very high and missing values were rare and infrequent. In the event of missing data, linear interpolation was used to fill missing values. To facilitate joining with the environment data that is limited to daily reporting, the network data was aggregated to daily data as well. Despite the temporal aggregation, this study provides unprecedented temporal resolution and duration. The network under consideration is a multi-vendor multi-access technology network; however, only LTE cells have a duration that covers the majority of the data collection period, and so 5G cells are excluded.

In addition to the KPIs outlined in Table 3.1, several composite KPIs are included in the analysis. Firstly, an approximation of the spectral efficiency is presented. This is done by taking the throughput of the cell and dividing it by the amount of PRBs used to achieve that throughput. Instead of considering the true number of PRBs used, the percentage of PRBs in use is instead considered. Since the objective of this analysis is to analyze trends across the network, the relative values of the KPIs are not of particular interest. For example, one cell might have a DL-THRP that is twice as high as another. Relationships across cells are not considered. The nature of this analysis therefore allows for the bandwidth allocation of any cell to be ignored in favor of the percentage of PRBs being used.

The second composite KPI was be DL-TV divided by AUE. This provides a measure for how much data the average user is consuming. Population studies have suggested that inclement weather causes more people to stay inside [80]. Although it is not a guarantee that they will use the internet, it is likely given the exponential increase in internet traffic [4]. It is expected that as temperatures decrease and snow accumulates that the traffic per user will increase due to more time being spent indoors on the internet.

3.4 Network-Environment Relationships

There are numerous ways to segment the data collected for the purpose of analyzing the relationship between environment and network features. Two levels of segmentation are considered: network and cell. The network-level analysis considers the time-aggregated results of all cells over time, while the cell-level analysis considers the performance at the cellular level. Each level of granularity provides a unique perspective. The network scale allows a macroscopic view to be obtained while the cell-level analysis will consider the data in the least aggregated form, which could reveal relationships that are obfuscated by the aggregation. A top-down analysis can be beneficial in many cases; however, for this work, the analysis is ordered by network and then cell. This is to allow the relationship at the cellular level to be introduced after a high-level observation of network performance.

For each level of analysis, Pearson's correlation (PC) and mutual information (MI) will be considered as described in Section 2.3.4. The empirical distribution of each network-environment features pair will be considered to evaluate how the subpopulations of cells behave relative to each other.

3.4.1 Network Analysis

Considering the network as a single entity is interesting in that it provides insight into the temporal behavior while abstracting the noise of any individual cell or site. Terrain, ecozone, configuration, and nearly any other variable in cells are abstracted. Daily cell data was aggregated in time using the mean across all cells.

Network aggregation is not meaningful from a network design and monitoring perspective; however, when considering global performance in terms of congestion or planning, it can provide a comprehensive overview of trend analysis. For example, if the congestion or subscriber count of the network were to be tracked, it could be meaningful to aggregate and analyze trends in a global sense. Additionally, tracking the number of subscribers and average user behavior could be better analyzed in a global context.

First, consider the PC and MI between pair of network-environment features. The I and R for all network-environment feature pairs can be found in Table 3.2. Using a significance value of $p < 0.05$, it was found that almost all pairs of characteristics had a significant relationship. All pairs involving temperature were statistically significant. Only the values that involved snow had insignificant correlations. This is not surprising because half of the year or more does not have snow, which could partially dampen the relationship. Since temperature is a year-round measurement, this does not happen. When comparing temperature and snow, they are inversely correlated; however, the temperature can go above zero while the snow fall and depth get clipped at zero.

The magnitudes of the identified correlations are indicative of moderate relationships between network and environment features; however, correlation does not mean causation. It can be seen that the relative information shared between the feature pairs is low. The magnitudes of the correlations range from roughly 0.1 to 0.6. On the contrary, relative mutual information ranges from approximately 0.02 to 0.17. This means that the amount of information that can be predicted about the network feature using the environmental feature is very small.

The measured metrics can be divided into roughly three groups of metrics related to unique properties: traffic, air interface, and users. PRB, THRP, and TV indicate that as temperatures drop and snow accumulate, the amount of spectrum resources used and the associated traffic increase. Increases in traffic are expected based on existing population behavioral studies [80].

Although increased traffic and resource use was anticipated, it was not expected that DL-SE would decrease with temperature. Furthermore, it can be seen that UL-INTRF increases in winter months. Traffic and resource use may increase; however, they are not more efficient based on the identified DL-SE change. This is probably caused by the noise floor increasing more than the received signal power. This leads to a reduction in the signal-to-noise ratio, which will degrade the spectral efficiency, as was observed. Although this is apparent at a network scale, further analysis is required on a cellular level and will be discussed in further detail in subsequent sections. Although not available for this study, user oriented data would provide an improved level of granularity and insight into the temporal changes of the physical environment.

Network KPI	Temperature (R)	Snow Depth (R)	Snow Fall (R)	Temperature (I)	Snow Depth (I)	Snow Fall (I)
DL-PRB	-0.567	0.470	0.324	0.057	0.052	0.041
AUE	-0.116	0.182	*-0.012	0.130	0.118	0.050
DL-TV	-0.141	0.161	*0.034	0.111	0.089	0.063
DL-THRP	-0.186	0.109	0.178	0.106	0.083	0.045
UL-INTRF	-0.168	0.128	0.188	0.084	0.084	0.035
UL-THRP	-0.294	0.210	0.243	0.065	0.065	0.020
DL-SE	0.256	-0.260	-0.088	0.055	0.051	0.022
UL-SE	-0.072	*0.011	0.103	0.166	0.142	0.078
DL-THRP-PAUE	-0.145	0.133	*0.070	0.066	0.047	0.031

Table 3.2: Correlation (R) and relative mutual information (I) between network scale, time aggregated, environment-network features pairs. Statistically insignificant correlations are indicated by * where $p \geq 0.05$

One of the key difficulties with considering network scale relationships is that commercial networks are constantly changing. Sites and cells are being added, customer counts are changing, and there is a team of engineers constantly redesigning and optimizing parts of the network. Newly added cells are inherently excluded from long-term trend analysis since they do not have a history to include. As a result, the observed cell population will be skewed towards a more *mature* subset of network infrastructure. A specific example of this can be seen in Fig. 3.2 where the number of AUE appears to be declining; however, this is simply due to the fact that more cells and capacity have been added.

At a network level, visual inspection of the time series data does not provide significant insight into performance. The PC and MI indicate that the relationships are low to moderate when the features are temporally aggregated. The strongest relationship between feature pairs is between DL-PRB and temperature. From Fig. 3.2 it can be seen that temporal variations closely align with those of temperature. The relationships are less visually apparent for the other pair of features.

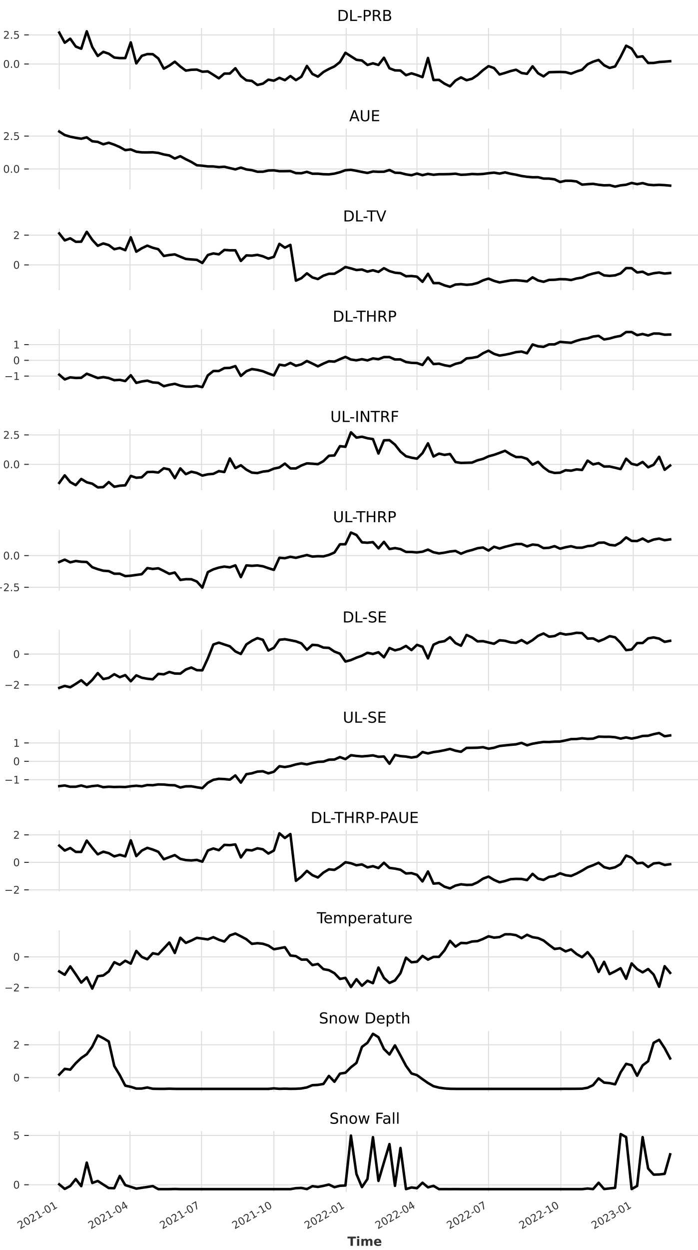

Figure 3.2: Weekly time aggregation of all network and environment features. Y-axis standardized to zero mean and unit variance.

3.4.2 Cell Analysis

Considering the network cell-by-cell will provide the best macroscopic view of the behavior of the network over time. User-level KPIs were not available with a duration as long as the RAN data; however, user-level data would need to be aggregated, since at the user level the behavior can change significantly on a daily basis. As such, the cell-level analysis provides a good trade-off between spatial and user behavior resolution.

From the population studies and the electromagnetic considerations it was expected that the environmental impacts would be more prevalent at this level. Aggregation can be used to help reduce the variations caused by noise; however, configuration changes and the deployment of new cells may result in these changes not always being noise. First, the distribution of the correlations between the network and environment features will be considered.

The network-level results presented in Table 3.2 are scalar in nature. In other words, there is only one value for each pair of metrics. Removing the time aggregation allows for the distributions of the relationships to be considered. In a global sense, metrics related to traffic volume are expected to increase as the temperature decreases. Furthermore, it is expected that the interference and spectral efficiency will increase as the temperature decreases.

When comparing the analysis of PC and MI, it is important to note that MI does not have an associated p-value, while PC does. As a result, the MI for all cells can be considered, while some PC values may be discarded due to statistical insignificance. It is difficult to have statistically significant correlations for small magnitudes of correlation. This is a result of the way p-values are defined. If all statistically insignificant correlations are removed, then the distributions become bimodal when in fact they are likely not. When presenting correlation distributions, the statistical significance will be ignored. Fig. 3.3 shows an example of the change in the correlation histogram that results from filtering by statistical significance. Cells that are discarded generally have correlations near zero. When observing all correlations, it is clearly a unimodal distribution; however, considering statistically significant correlations results in a bimodal distribution may not accurately characterize the relationship. Furthermore, the segmentation of the cell population is considered in the context of forecasting in Section 3.5 where the positive, negative, positive and negative, insignificant, and all cells are considered for modeling purposes.

The distribution of all correlations on a cellular level can be seen in Fig. 3.4. The distributions that included snowfall had the lowest variance of all the distributions. Most of the distributions including snowfall were $|R| < 0.25$ while the temperature and snow depth distribution pairs are closer to $|R| < 0.5$. This result is expected since snow fall is zero for most of the year, while the temperature is continuous and the depth of the snow is less sporadic. The mean values for each of the distributions are non-zero; however, the correlations between snow depth and AUE are approximately zero.

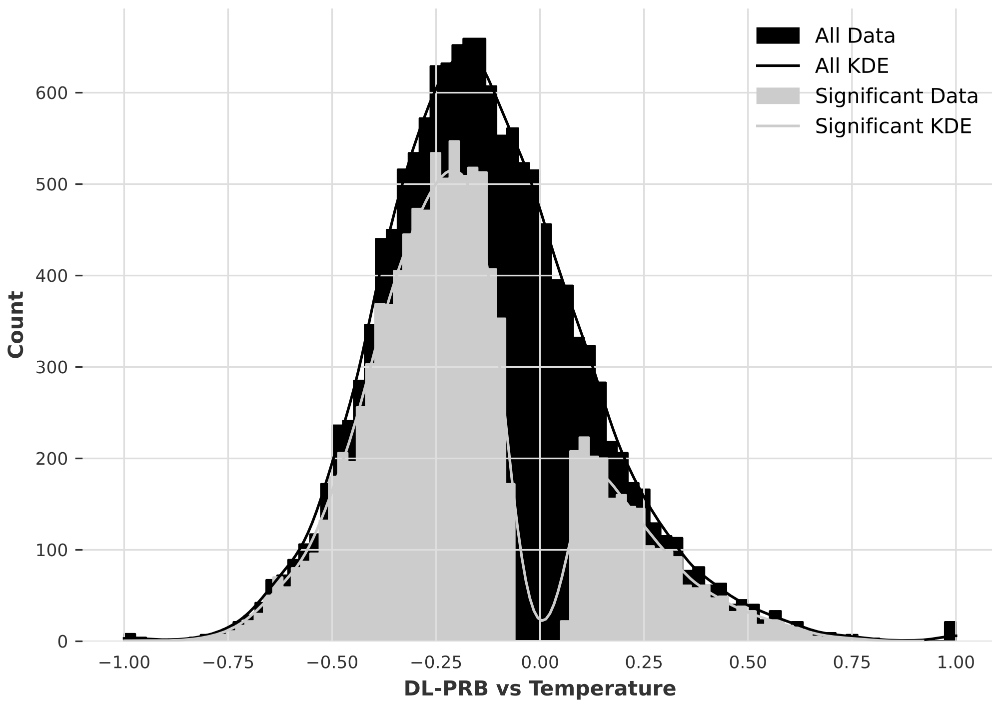

Figure 3.3: Histogram of DL-PRB and temperature for statistically significant and all correlations. The black shaded area represents the distribution of all cell relationships while the grey is only those that are statistically significant. The lines tracing the contour of the histograms are the kernel density estimate (KDE).

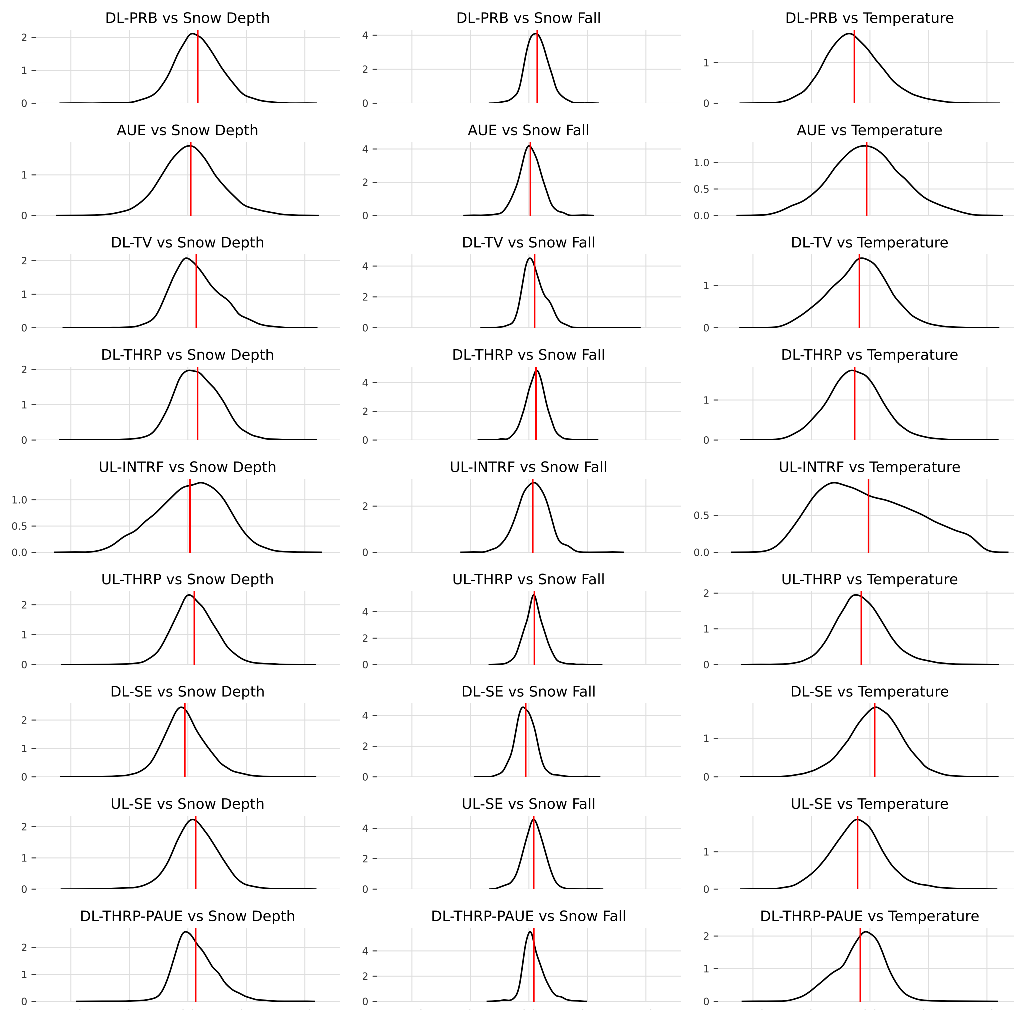

Figure 3.4: Probability density estimates for all correlations regardless of statistical significance. Red vertical line indicates the mean of the distribution.

Temperature and snow depth have similar relationships among network pairs; however, the signs of distribution are inverted. This is intuitively explained by the fact that temperature and snow are inversely related. Low temperatures allow snow to fall and accumulate, while high temperatures cause snow to melt. Similarly to the results of snowfall versus network features, the network-environment feature pairs with temperature and snow depth are somewhat Gaussian in nature. Feature pairs with the least Gaussian-like properties are those that evaluate the correlations between UL-INTRF, DL-THRP-PAUE (DL throughput per active user), temperature, and snow depth.

UL-INTRF was expected to vary significantly between seasons. Changes in the propagation environment introduced by snow and foliage were expected to result in a large change in interference. Snow with a crust of ice on top of it will likely have a higher permittivity than soil, meaning that ground reflections would become stronger depending on their incident angle. Furthermore, the lack of foliage was expected to reduce the loss of signal path, resulting in an increase in interference. From the distributions, it is not immediately apparent that this is the case. The distributions are skewed so that snow and foliage could be the driving factor for UL-INTRF changes; however, the means of the UL-INTRF distributions are still near zero.

Unlike the UL-INTRF distributions, the DL-THRP-PAUE compared to temperature and snow depth exhibits the expected result of increasing traffic per user during the cold and snow months. This aligns with the aforementioned population studies and the inference that a population spending more time indoors results in more internet traffic. Moreover, the mean of the distributions is such that the average cell across the set experiences increased traffic in cold snowy months. Although correlation can be a good indicator of the relationship between, MI will provide a stronger measure of the potential to exploit the relationship for forecasting purposes.

MI will provide a better characterization of the relationships between network and environment feature pairs. PC's constraint of only identifying linear relationships may fail to capture all shared information. Using MI, specifically relative MI, the amount of information known about the network feature contained in the environment features can be determined. The estimates of the probability density function of relative MI can be seen in Fig. 3.5. A large number of feature pairs have $I < 0.1$. All pairs involving snowfall are much lower than this. As with correlation, there is no definitive threshold for what a strong relationship is; however, small values of relative MI mean that most of the information in the signal is self-contained and adding exogenous features may provide little or no benefit.

The correlation and MI relationships between feature pairs are somewhat related to network-wide relationships. The signs of the correlations are generally aligned between the network scale and the negatively correlated cell population at the cell level. All relationship pairs can be found in Table 3.3.

Snowfall has been identified as a feature that has approximately no relationship to network features. At a cell level, the average MI was approx-

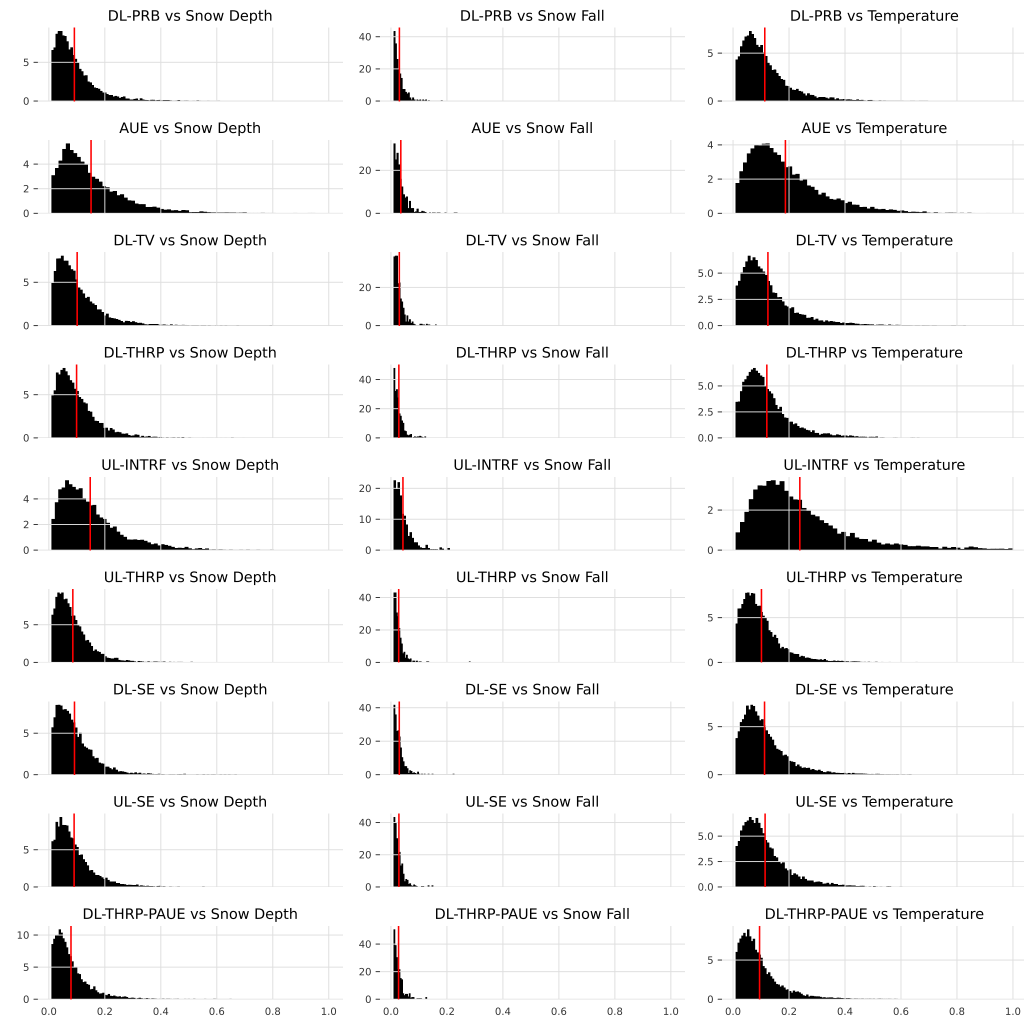

Figure 3.5: Probability density estimates for all MI. Values less than 1E-2 were discarded from the visualization. Red vertical line indicates the mean of the distribution which does include the values discarded from the visualization.

imately zero. Furthermore, The correlations were all less than 0.1 indicating that the relationships are not strong. This result is expected since the amount of snowfall per day will be a sporadic and noise-like signal. For a significant amount of time, the signal will be zero, followed by unpredictable bursts.

Snow depth and temperature exhibited a much stronger relationship with network features in terms of correlation and MI. Normalized MI values were all low, most of them below 0.01; however, the temperature had correlations with magnitudes as high as 0.250. The correlation indicates that including temperature could improve forecast accuracy, but the relative MI indicates the contrary. This strongly suggests that it will not improve performance.

An important thing to note with these relationships is that the correlations and MI values are between the signals at the same time. For forecasting, it is often better to consider the cross-correlation function and partial cross-correlation function to evaluate the temporal relationship between signals. A similar process could be used to identify lagged relationships with MI. Although this would present a characterization of the relationships across time, this is foregone in favor of creating neural network models which leverage the data directly. Moreover, the relationships between the network and environmental features are not particularly strong, and it is not anticipated that lagged correlation analysis will yield further insight. It is also noteworthy that neural network models will be able to create complex functions that capture learnable relationships. There is always the risk of over fitting; however, careful training and configuration can allow the relationship between network and environmental features to be captured and characterized by models.

Network KPI	Correlation	Snow Depth I	Snow Depth R	Snow Fall I	Snow Fall R	Temperature I	Temperature R
AUE	+	0.022	-0.051	0.000	-0.016	0.036	0.115
	-	0.021	0.083	0.000	0.032	0.036	-0.128
DL-PRB	+	0.009	-0.008	0.000	0.022	0.012	0.024
	-	0.013	0.154	0.000	0.098	0.023	-0.250
DL-SE	+	0.011	0.035	0.000	0.014	0.018	-0.061
	-	0.014	-0.065	0.002	-0.043	0.023	0.107
DL-THRP	+	0.012	0.039	0.000	0.038	0.020	-0.046
	-	0.014	0.127	0.000	0.074	0.027	-0.194
DL-THRP-PAUE	+	0.009	0.029	0.000	0.027	0.014	-0.029
	-	0.011	0.084	0.000	0.048	0.019	-0.114
DL-TV	+	0.011	-0.002	0.000	0.008	0.016	0.034
	-	0.016	0.122	0.000	0.068	0.025	-0.175
UL-INTRF	+	0.025	-0.012	0.000	0.033	0.060	0.037
	-	0.024	0.029	0.001	0.039	0.056	-0.032
UL-SE	+	0.012	0.088	0.000	0.063	0.021	-0.128
	-	0.012	0.070	0.000	0.040	0.022	-0.105
UL-THRP	+	0.011	0.024	0.000	0.038	0.017	-0.014
	-	0.011	0.091	0.000	0.061	0.019	-0.130

Table 3.3: Mean correlation and relative MI by correlation population. Only cells with statistically significant correlation are considered.

3.5 Forecasting

The seasonal behavior of a network is critical to understanding network planning and management. In the event that long-term seasonality is not considered, network upgrades or investigations could be performed, despite variations caused by external factors beyond the control of the network operator. One way to address this problem is to use long-term forecasting to predict network performance. Seasonality can be taken into account by using these forecasts to compare with the actual signal. Explainable seasonality is good; however, unpredictable changes will appear when forecasts are inaccurate. This could indicate anomalies or changes in the use pattern that could merit investigation.

Section 3.4 considered the relationship between nine network KPIs, outlined in Table 3.1, and three environmental features. There is a significant amount of shared information between some of the network features, such as DL-THRP and DL-PRB being related because the throughput will be related to the amount of spectrum being used. From a network planning perspective, nine metrics are likely too many for long-term capacity planning. The number of combinations and variability across features can become too difficult to fully characterize. In reality, it is likely sufficient to leverage a smaller subset. The original set of nine KPIs was reduced to a total of four to address this problem and make the forecast more manageable for network engineers. There is also a significant amount of information shared across many of these metrics, which means that a large amount of information will be redundant. Furthermore, the forecasting of metrics like interference are not meaningful from a capacity planning perspective.

Network capacity planning and management generally aim to optimize downlink performance. In other words, most users consume more data than they send. To accommodate this planning, the set of four KPIs selected as the target variables for the prediction will be DL-THRP, DL-PRB, DL-TV, and AUE. This subset of KPIs allows for some important relationships to be observed, such as throughput per use, traffic per user, spectral efficiency, and user growth. Of the nine previously considered KPIs, three were composite KPIs, two of them completely dependent on the DL KPIs. Interference was previously analyzed; however, it was discarded from the forecast, as it is generally ignored in network capacity planning.

3.5.1 Data and Model Configuration

Using network KPIs, forecasting models will be created that consider target features with and without environmental features as covariates. In addition, several different forecasting time horizons will be considered to identify if and when environmental features benefit the forecasting task. The time horizons were set to 4, 8, 12, 16, and 20 weeks. This translates to 28, 56, 84, 112, and 140 day horizons. For each time horizon, the input sequence length is the same length as the target horizon. This is similar to many existing time series forecasting studies [1, 2]. The overarching objective of these forecasting models is to create long-term forecasts; however, long time horizons are problematic in terms of creating training data due to the desire to capture long-term seasonalities.

It is best practice to consider a training, validation, and test set when training a machine learning model. A model will be trained on the training set. Validation sets are used to determine when to stop training, and finally, test sets are used to quantify the performance of the model. In many cases, the validation and test sets are the same. Regardless of the decision to use two or three datasets for training, it is critical not to have data leak from the training set into the validation and test sets. In time series forecasting, this is generally avoided by segmenting the data in a temporal manner. In other words, pick a date and call the data before the date the training set, and anything after it the validation or test sets. Splits are often created as 80-10-10 percent when using train-validation-test or 80-20 when the validation and test sets are combined. This section of the work considers an 80-20 split in which the validation set is used as the test set.

Long-term forecasts with finite histories present a major problem: How can a large test and validation set be obtained when forecasting horizons are almost as long as the duration of the data collected? Instead of temporally cutting the data into sets, the data can be spatially segmented. Instead of temporally cutting the data, it is cut by cell. This allows longer time horizons to be used while still having test sets that are a good representation of the training set. It is also important that spatial slicing is used instead of temporal to ensure that snow will be present during both training and test periods. There is the risk that a model including environment features may train on a set of data that has snow present, but then be validated on a set with no snow present. This is a risk when using temporal slice;

however, this becomes a non-issue when spatial slicing is used. Although spatial slicing resolves many potential problems, it introduces the potential for leakage between data sets. Two cells could share the same environmental data, but one cell ends up in the training set, while the other is in the test set. If performance improvements are significant when environmental covariates are added, a potential cause could be the aforementioned data leakage.

Four different model architectures will be considered: GRU, Transformer, N-HiTS, and DLinear. These models were selected because the first two are iterative forecasting strategies while the latter are direct forecasting strategies. The parameters of the GRU model were set at two layers with 64 hidden units per layer, a configuration similar to that suggested by an experimental survey of model architectures for time series forecasting [91]. The Transformer used the same configuration as in the foundational study [66] except that the number of encoder and decoder layers was reduced from six to three. Similarly, N-HiTS [1] and DLinear [2] consider the same configuration as their respective introductory studies. The learning rate, gradient clipping, and batch size were set to 1E-4, 1, and 2048 respectively. Initial small hyperparameter tuning runs were used to identify the learning rate. Higher learning rates resulted in models that converged to different values; however, for 1E-4 and lower, the results obtained were identical. The batch size was set to optimize the VRAM use of the GPUs used for model training. Again, small hyperparameter tuning runs indicated that training results were insensitive to batch size; however, larger batch sizes greatly accelerated training times. The training epoch limit was set at 100 with the option for training to exit early on the condition that the improvements were less than 1E-4 for 5 epochs.

3.5.2 Results

The performance of trained models will be considered in three unique contexts: performance by forecast horizon, performance by model type, and performance by correlation population. These levels of granularity will provide insight into what the best performing model architectures are, what forecasting strategy is better, and over which time horizon environmental features can improve forecast accuracy. Furthermore, it can assess whether there is merit in further segmentation of cell populations to improve forecast results. For each forecast horizon considered, a model was trained with and without environmental features. Additionally, a model was trained on each population.

It is important to note that long-term KPI forecasting for LTE and 5G networks has been largely ignored. Forecast time horizons are generally short, often less than one day [12, 7]. It is important to clarify two terminologies: long term forecasting or long time horizon forecasting. It is largely a semantics argument; however, long-term forecasting could be a long time duration into the future, while long time horizon forecasting could be a large number of points. For example, consider the process of forecasting the next 24 hours in 5-minute intervals. It is a relatively short duration in time; however, 288 step forecasts are long in terms of a model's ability to forecast accurately. Forecasting the traffic of urban mobile cells for the next 24 hours in 10-minute intervals has been shown to benefit from the inclusion of environmental, spatial, and other exogenous features [7]. The results presented in this section are both long time horizon and long temporal duration in a rural fixed wireless network that has yet to be considered in the published literature.

The MSE and MAE for all combinations of cell population can be found in Table 3.4. To reiterate, the *Combined* population is all cells, *Correlated* are cells with statistically significant positive and negative correlation, *Insignificant* is any cell without statistically significant correlation with environment features, *Negative* is negatively correlated cells, and *Positive* are positively correlated cells. For cell population segmentation, the sign of the relationship is that between DL-PRB and temperature.

The accuracy of the forecast by horizon provides the greatest insight into the relationship between the pair of network and environmental features. In total, there are five correlation populations to consider; however, the horizon analysis considers only two: correlated and combined. Histograms of the correlated and combined population training results with and without environmental features can be found in Fig. 3.6.

For the correlated cell population, it was found that including environmental features only increased the accuracy of the forecast when the horizon reached 12 weeks. Beyond the 12-week horizon, the best models always in-

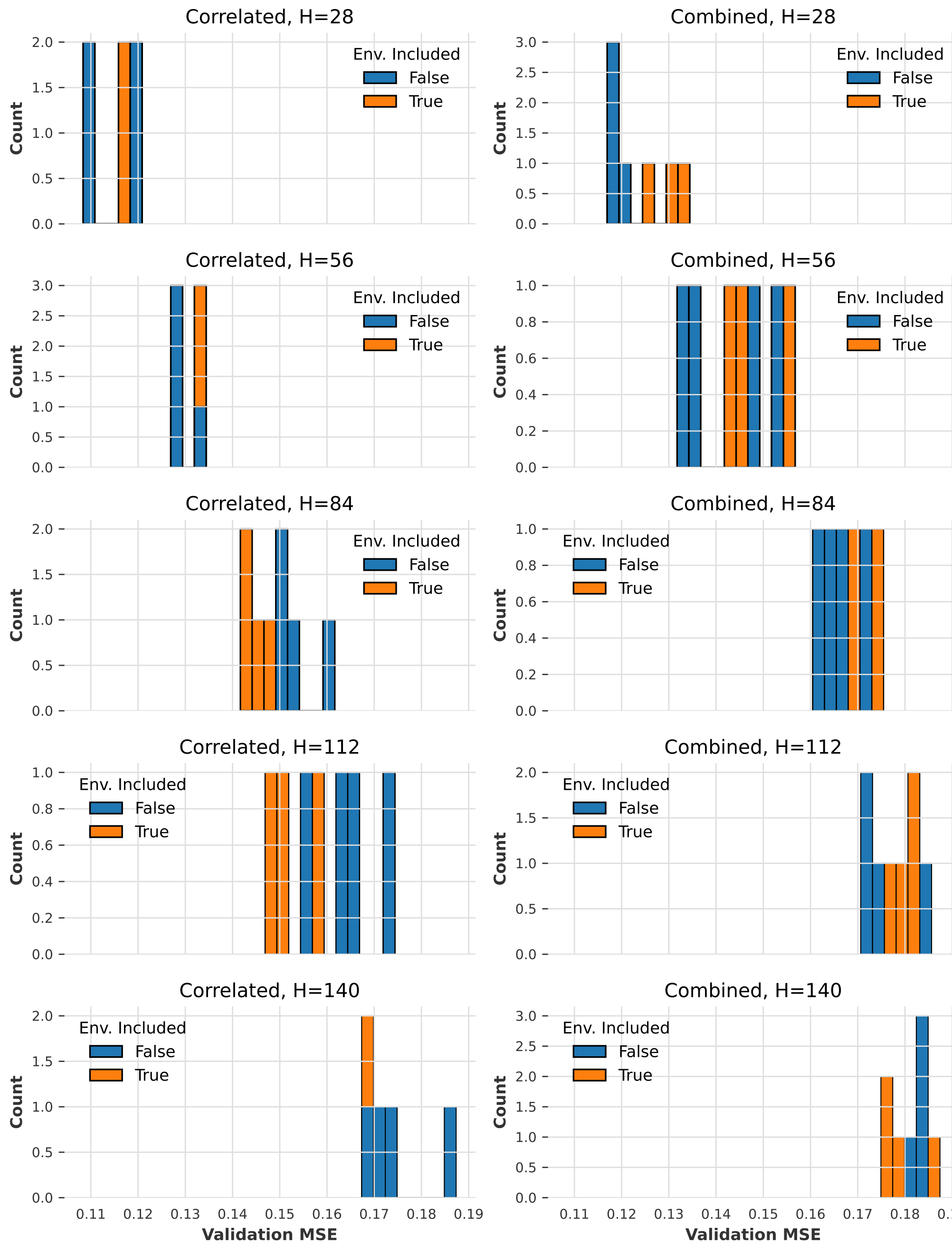

Figure 3.6: Forecast accuracy by time horizon, selected cell correlation population, and inclusion of environment features.

Population	Model	Horizon / Metric / Env.	28 MSE	28 MAE	56 MSE	56 MAE	84 MSE	84 MAE	112 MSE	112 MAE	140 MSE	140 MAE
Combined	DLinear	False	0.119	0.227	**0.132**	**0.240**	0.164	0.273	0.172	**0.278**	0.184	0.291
		True	0.125	0.233	0.145	0.252	0.164	**0.270**	0.182	0.286	0.179	0.287
	GRU	False	**0.117**	**0.224**	0.149	0.255	0.173	0.276	0.184	0.286	0.183	0.289
		True	0.130	0.233	0.153	0.255	**0.160**	0.270	0.179	0.283	**0.175**	**0.283**
	NHiTS	False	0.119	0.230	0.134	0.250	0.163	0.274	**0.171**	0.278	0.181	0.291
		True	0.118	0.226	0.142	0.254	0.168	0.278	0.176	0.285	0.177	0.290
	Transformer	False	0.121	0.230	0.153	0.258	0.166	0.274	0.173	0.284	0.183	0.291
		True	0.133	0.234	0.156	0.256	0.174	0.274	0.181	0.285	0.185	0.292
Correlated	DLinear	False	0.120	0.227	0.129	0.237	0.160	0.263	0.163	0.274	0.170	0.282
		True	0.119	0.226	**0.127**	**0.235**	0.147	0.255	0.157	0.272	0.170	0.283
	GRU	False	0.119	0.226	0.129	0.237	0.152	0.258	0.156	0.267	0.174	0.283
		True	0.118	0.225	0.133	0.244	0.144	0.256	**0.147**	**0.260**	**0.167**	**0.280**
	NHiTS	False	**0.108**	**0.217**	0.129	0.239	0.150	0.259	0.166	0.279	0.171	0.282
		True	0.118	0.227	0.132	0.245	0.144	0.258	0.150	0.265	0.168	0.284
	Transformer	False	0.109	0.217	0.133	0.238	0.154	0.261	0.172	0.281	0.185	0.289
		True	0.119	0.226	0.134	0.244	**0.142**	**0.248**	0.172	0.282	0.175	0.284
Insignificant	DLinear	False	0.147	0.253	0.188	0.284	0.200	0.298	0.246	0.334	0.239	0.329
		True	0.147	0.253	0.178	0.282	0.193	0.294	0.214	0.315	0.225	0.328
	GRU	False	0.147	0.252	0.190	0.286	0.200	0.297	0.240	0.328	0.245	0.330
		True	0.147	0.256	**0.177**	**0.280**	0.202	0.302	0.230	0.326	**0.222**	**0.326**
	NHiTS	False	0.145	**0.251**	0.186	0.286	0.198	0.297	0.245	0.337	0.253	0.346
		True	**0.144**	0.252	0.179	0.283	**0.186**	**0.290**	**0.208**	**0.314**	0.239	0.339
	Transformer	False	0.158	0.263	0.193	0.287	0.197	0.300	0.257	0.345	0.247	0.335
		True	0.147	0.253	0.181	0.285	0.191	0.291	0.245	0.334	0.233	0.336
Negative	DLinear	False	0.115	0.222	0.132	0.242	0.146	0.257	0.154	0.264	0.183	0.294
		True	**0.110**	**0.214**	0.133	0.242	0.145	0.254	**0.146**	**0.260**	0.158	0.274
	GRU	False	0.113	0.219	0.132	0.241	0.146	0.257	0.175	0.276	0.167	0.277
		True	0.116	0.223	**0.130**	**0.240**	**0.141**	**0.250**	0.163	0.269	**0.157**	**0.273**
	NHiTS	False	0.115	0.224	0.132	0.242	0.146	0.258	0.153	0.265	0.169	0.280
		True	0.112	0.224	0.132	0.245	0.142	0.253	0.146	0.263	0.160	0.275
	Transformer	False	0.113	0.219	0.132	0.242	0.151	0.261	0.154	0.264	0.177	0.287
		True	0.116	0.223	0.132	0.242	0.143	0.252	0.154	0.265	0.166	0.281
Positive	DLinear	False	0.140	0.249	0.155	**0.259**	0.175	0.281	0.180	0.284	0.184	0.290
		True	0.140	**0.248**	0.154	0.259	0.174	0.280	**0.178**	0.283	0.182	0.289
	GRU	False	0.141	0.249	**0.153**	0.259	0.176	0.282	0.180	0.284	0.182	0.288
		True	0.140	0.249	0.153	0.259	**0.173**	**0.279**	0.178	**0.280**	**0.179**	**0.286**
	NHiTS	False	0.140	0.250	0.155	0.262	0.175	0.284	0.182	0.288	0.184	0.292
		True	**0.138**	0.249	0.155	0.264	0.173	0.283	0.181	0.288	0.181	0.292
	Transformer	False	0.142	0.249	0.156	0.261	0.180	0.285	0.186	0.287	0.189	0.294
		True	0.141	0.250	0.155	0.261	0.178	0.282	0.183	0.287	0.187	0.294

Table 3.4: Forecasting performance of different models across different forecasting horizons, cell populations, and environmental covariate inclusion. Best results by population are in bold.

cluded environmental features. This result is interesting in that the correlation and the MI were indicative that the instantaneous relationship between network and environment features was low; however, these results indicate that as the time horizon increases, environmental features have a greater impact. Although the results clearly favor the inclusion of environmental features for cells with statistically significant correlation, the performance improvement in the forecast accuracy is minimal. The greatest gains were found with the 16-week horizon, in which the MSE was reduced by almost 6% by including environmental features; however, there was an increase of approximately 1% in the MSE for the 4-week horizon.

The performance of the cell population without attention to the correlation properties had results similar to those of the statistically significant correlation population. Until the horizon increased beyond 12-weeks, the MSE was not definitively in favor of including environmental features. Despite this observation, longer horizons are not significantly better when statistically insignificant cells are added to the available population of cells.

Model architectures for time series forecasting are rapidly evolving. The selected models cover a large number of architecture choices such as the inclusion of attention, block-based models, and IMS or DMS. The performance of the models will be evaluated across the different horizons without segmenting the data by cell population. The model validation MSE by model architecture and time horizon can be seen in Fig. 3.7.

One of the most astonishing results of the model performance is how poorly the Transformer performed. Although it was a vanilla Transformer, the median validation MSE across all time horizons is worse than any other architecture. The use of attention mechanisms has not been shown to benefit short-term forecasting using Seq2Seq models [37]; however, it was expected that long-term forecasting would benefit from the use of attention, as multiple seasonalities and long temporal effects could be easily captured. Furthermore, it was unexpected that the GRU model would be top performing in any configurations; however, it achieved the best results on all 140-day horizon.

The accuracy results by population identified that segmentation based on correlation properties can improve the accuracy of the models. Box plots for the performance by horizon and population can be found in Fig. 3.8. The negative population was approximately 10-times larger than the positive population and 3-times larger than the insignificant population. Across all horizons the positive and insignificant populations had the highest validation MSE. It can be seen that combining the positively and negatively correlated populations yielded better results than the positive population alone. Moreover, combining all populations allowed for the insignificant population to have a significant reduction in forecasting error across all horizons. Models trained on a given population, regardless of environmental covariates, behave similarly. This observation supports the segmentation of the cells into unique populations based on their relationship to the physical deployment environment.

3.6 Conclusion

Rural propagation studies have been performed and provide some information on seasonal changes that a rural wireless network could experience. There are a large number of variables that affect the propagation performance of a network. Seasonal user behavior, snow, temperature, foliage, and many

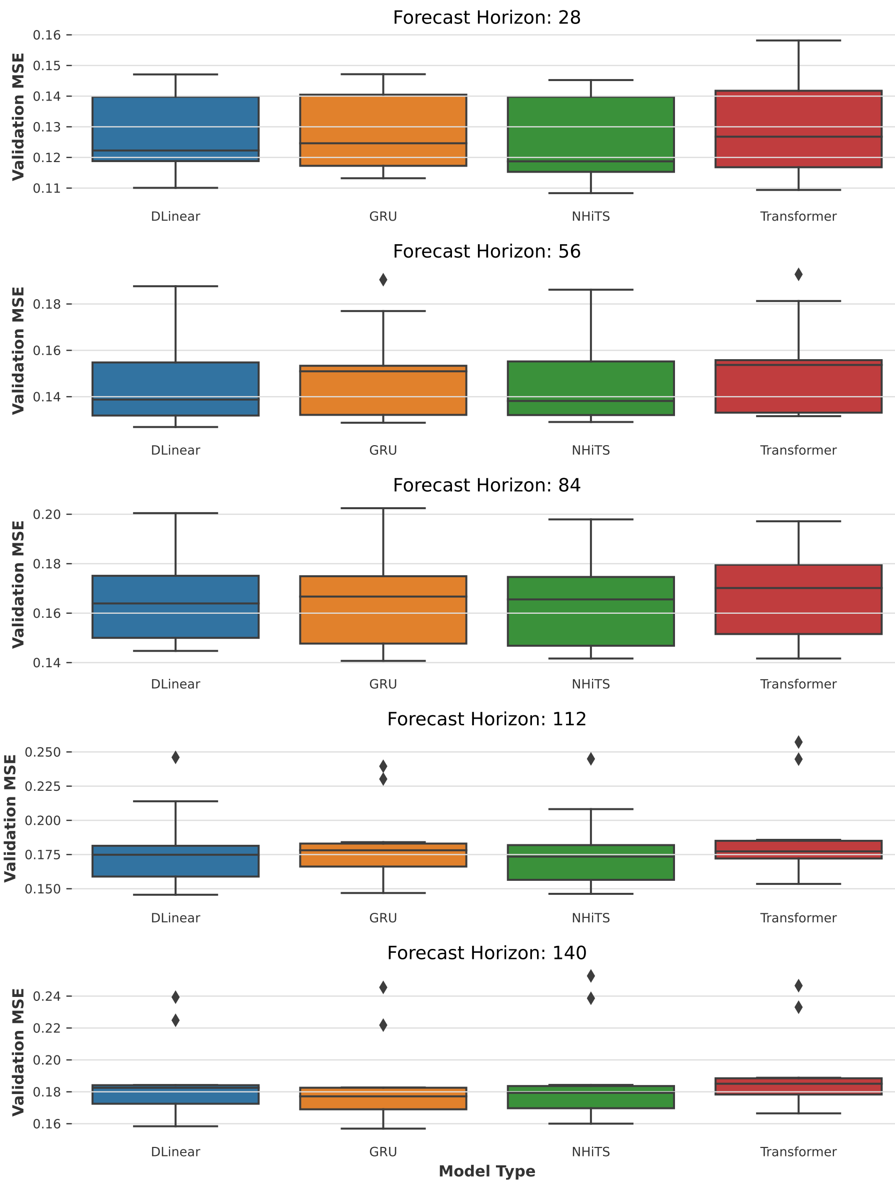

Figure 3.7: Forecast accuracy by model and time horizon. Vertical edges of the boxes indicate the upper and lower quartiles, diamonds are outliers, and the horizontal lines extending from the boxes are the non-outlier limits of the results. Horizontal line in the box indicates the median.

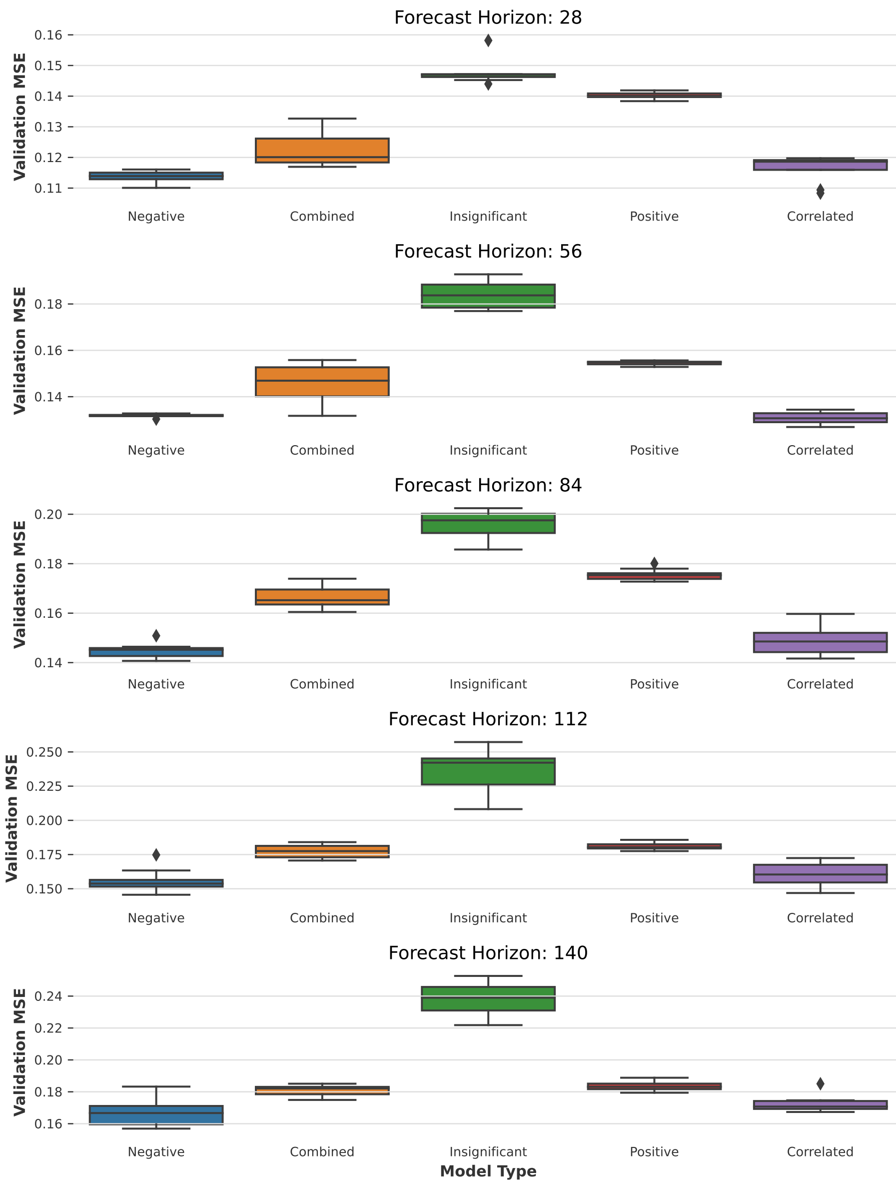

Figure 3.8: Forecast accuracy by correlation population and time horizon. Vertical edges of the boxes indicate the upper and lower quartiles, diamonds are outliers, and the horizontal lines extending from the boxes are the non-outlier limits of the results. Horizontal line in the box indicates the median.

other changes in the propagation environment can impact performance. The work outlined in this chapter has demonstrated that there exist moderate correlations between selected network and environmental features. Using the unprecedented level of temporal granularity and duration, it is observed that the accuracy of the forecast can be improved using the identified relationships.

Forecasting a long time horizon in terms of both duration and number of steps was demonstrated to be improved using environmental covariates. Including covariates was shown to help the forecasting accuracy for longer time horizons. For short-term forecasting, it may not be practical to include environmental covariates as it will increase the difficulty in achieving deadlines due to increased computation time; however, for long-term planning and capacity management, a more accurate forecast will result in more efficient use of network operator capital expenditure. Furthermore, it has been identified that segmenting cells based on their relationship to the physical environment can allow for more accurate forecasts to be obtained in the unique populations.

The expansion of this work has several interesting avenues. First, the exploration of more complex population segmentation could yield further improvements in accuracy; however, there will be a finite limit to how many populations can be created due to the amount of data available. Training models, deep learning or not, with a insufficient data will result in poor performance. Second, instead of segmenting populations further, a hierarchical forecasting model could be created that leverages features relating to province or user behaviour. This could allow for a model to learn the optimal population segmentation rather than a statistical analysis; however, understanding the fundamental properties of the phenomena can be just as important as the result when it comes to the capital expenses required to build and upgrade a commercial wireless network.

Chapter 4

Short-Term Multivariate Forecasting

Long-term forecasting can be easily used for network design and capacity management. Knowing the long-term behavior of a network has many benefits; however, short-term forecasting can enable a separate set of advanced network management features such as dynamic resource allocation and anomaly detection [5]. For clarity, short-term in this context means a short time span rather than a small number of points ahead; however, the intent of this work is to focus on a small number of points ahead.

Urban mobile deployments are the predominant focus of short-term forecasting in LTE and 5G networks. Forecasting LTE performance KPIs has shown promising results in urban environments. Furthermore, including environmental-specific information in the models has been shown to dramatically increase the accuracy of the forecast [7, 12].

The objective of this chapter is to investigate multivariate short-term forecasting in a rural fixed wireless network. LTE and 5G forecasting studies have focused mainly on urban mobile networks. The temporal dependencies of urban mobile networks have been shown to be very long, indicating that they are excellent candidates for forecasting [11]. Similar temporal studies have not been conducted for rural fixed wireless networks. This work will begin with an investigation of the temporal properties of the KPIs. Once the temporal properties have been characterized, the inclusion of covariates will be investigated and compared with urban studies.

There are many KPIs that could be selected for forecasting; however, this section focuses on AUE, DL-THRP, DL-PRB, UL-THRP, and UL-PRB. This is a small set of KPIs that allows for the identification of spectral efficiency and performance per user. It is important that short-term forecasting application consider as succinct a set of KPIs as possible to reduce the dimensionality of the problem. The data was collected from a rural Canadian fixed wireless network with a sampling period of 15 minutes. 1000 cells were considered from September 2022 to February 2023. Missing time steps were infrequent and filled using linear interpolation.

4.1 Temporal Properties

Understanding the temporal properties of the signals to be forecasted is critical. Model selection and hyperparameter tuning can result in a significant search space for an optimal model. If the underlying time series has a strong relationship between the current and the past two or three time lags, the selection of a model that considers the past 100 time steps will be meaningless. The argument can be made that a model should be larger and trained longer [92]; however, with short-term forecasting, a larger model will result in a longer time to make predictions. For activities such as anomaly detection and dynamic resource allocation, it is often better to have a faster model with marginally lower accuracy. In other words, if action cannot be taken on the forecast data before the next samples arrive, the forecast is useless. There are cases where the use of historical forecast data can be beneficial for tasks such as anomaly detection, which will be discussed in Chapter 5; however, for dynamic resource allocation, historical forecasts are often meaningless.

Existing studies on time series forecasting for LTE and 5G networks have been almost exclusively based on deep learning. Deep learning time series studies typically abandon temporal analysis prior to model training and evaluation [2, 1]. An excellent example of temporal analysis can be found in [11] where ACF was identified as significant for weeks in urban mobile networks.

A major fault with many existing studies is that an arbitrary history is selected without taking into account the periodicity of the temporal properties of the KPIs. In this section, three approaches will be considered: stationary properties of the KPIs, ACF, and PACF.

Many deep learning approaches are not affected by aperiodic or non-stationary signals. For example, architectures such as N-HiTS can forecast non-periodic and non-stationary signals [1]. In contrast, ARIMA, SARIMA, and other statistical models often require that signals be stationary.

Stationary testing will be considered using the KPSS and the ADF test, respectively. Two tests and the decision to accept or reject the null hypothesis result in four combinations, each having their own unique meaning. For the ADF, the null hypothesis identifies if the series has a unit root, while the alternative is that there is not one. For the KPSS, the null hypothesis is that the series is trend-stationary, while the alternative is that it has a unit root. If both tests indicate that a series is stationary, then the signal is stationary. If both indicate that the signal is nonstationary then it is nonstationary. When KPSS indicates that the signal is stationary and ADF does not, the signal is trend stationary. If ADF indicates stationary and KPSS does not, then the signal is difference stationary.

Three of the four combinations of the results of the stationary statistical tests indicate that a given signal would be stationary in some capacity. In other words, as long as one of the tests indicates that the signal is stationary, then it is stationary, trend stationary, or difference stationary.

Using a significance level of 0.05, nearly all KPIs in all cells tested were stationary according to the KPSS or ADF tests. Approximately 20 to 40% of cells were determined to be stationary on a KPI basis according to KPSS and ADF. Performance metrics were closer to 40% while AUE was closer to 20% of cells. ADF indicated that 95% of the cells were stationary in all KPIs. In contrast, KPSS found that only 20 to 60% of the cells were stationary across all KPIs. In summary, nearly all cells and KPIs are trend stationary. Furthermore, most of the KPIs are difference stationary. Although stationarity does not alter the forecasting approaches taken in this section, it is beneficial to be aware of the underlying properties of the signals for hyperparameter tuning.

The ACF and the PACF are often used to select the order of ARIMA models [38]. PACF can be used to select the AR order, while ACF can be used to determine the MA order. Although ACF and PACF are not always directly beneficial to neural network-based approaches, they can be used to reduce the hyperparameter space for models to a more manageable search space.

An example of the ACF and PACF for the DL-PRB of a single cell can be found in Fig. 4.1. These results are generally consistent across all cells and KPIs. The ACF indicates that there is a long temporal dependency. In other words, the shape of the signal is repeating. This indicates that a large number of MA lags could be used for an ARIMA model. The PACF indicates that the number of significant independent correlations between the current and past values is small except for the first few lags and near a 24-hour delay. The shaded regions show the 95% confidence interval, which indicates the boundary for statistically significant values. There are not many significant PACF values, which suggests that a low AR order could be used for an ARIMA model or a shorter number of input steps for other architectures.

The results of the PACF and ACF analysis can be used to evaluate the temporal dependencies of a signal. These dependencies can be used to set parameters, such as the number of input samples to be considered. For example, the values presented indicate that including a long history is unlikely to improve the accuracy of the forecast. Furthermore, forecasting models such as LSTM and GRU will struggle with incredibly long dependencies. Not all models will be able to adequately capture the increase in the input sequence length. Transformers and other forecasting models may perform better if they attend to all input samples at the same time.

4.2 Hyperparameters

Many forecasting studies, especially LTE KPI forecasting, provide inadequate information to replicate their work. In many circumstances, closed-source data sets are used; however, the model architectures presented are often poorly described. For example, many studies simply state that they used ARIMA rather than specifying the order of the components [1]. Many studies

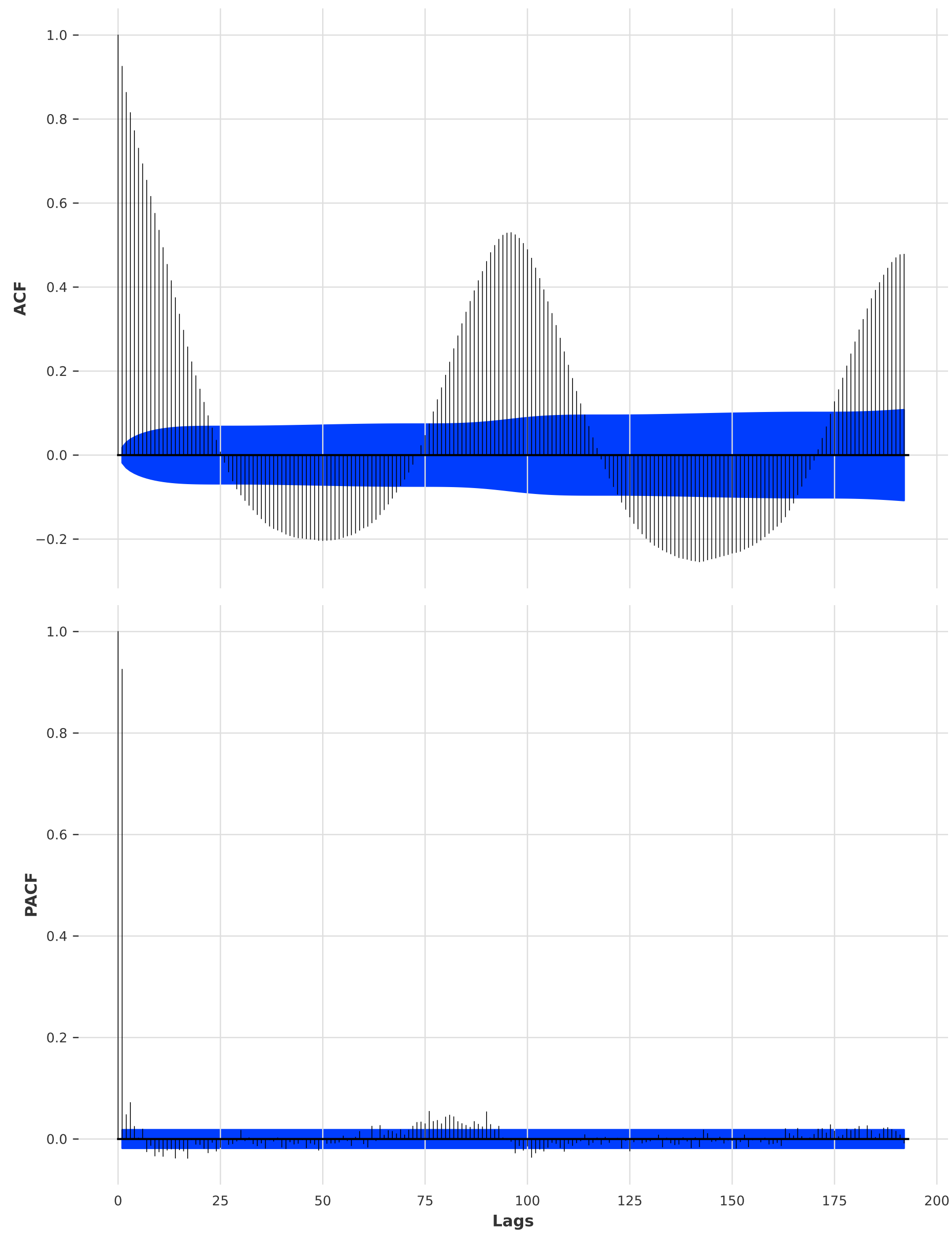

Figure 4.1: ACF and PACF of DL-PRB for zero to 192 lags. Shaded region indicates the 95% confidence interval.

do not present any indication of hyperparemeter tuning [93, 94]. The models considered in this chapter are DLinear, NLinear, GRU, N-HiTS, Transformer, Seq2Seq, and Seq2Seq+Attention.

NLinear and DLinear have a very small number of tunable parameters. NLinear has none, while DLinear has a smoothing kernel size to estimate seasonality. This kernel was kept constant at 25 samples for all input sequence lengths to smooth out short-term noise and variations.

GRU, Seq2Seq, and Seq2Seq+Attention are all RNN-based models. The Seq2Seq models used GRU as the RNN component of the encoder-decoder. This was done instead of using LSTM since Seq2Seq-GRU has been shown to outperform Seq2Seq-LSTM in short time intervals [95]. From a similar study and an empirical analysis in a large number of time series datasets, the number of hidden cells and layers was set to 64 and 2 respectively [37, 91]. For GRU, this describes the entire model, while for Seq2Seq models, it describes the architecture of the encoder and decoder separately. For the attention mechanism in the Seq2Seq+Attention, scaled dot product attention was used as described in the original Transformer manuscript [66].

N-HiTS was configured with three stacks, one block, and two multi-layer perceptrons of width 512. This was found to be the optimal configuration in the original article [1]. Parameter tuning on small subsets of the data found negligible changes in performance across different configurations.

The transformers were configured in the same way as in the original paper; however, the number of layers in the encoder and decoder was reduced from six to three [66]. This reduction was made without a decrease in accuracy while improving computational efficiency.

With the models described, it is also important to note the training parameters. The learning rate was set to 1E-3 with gradient clipping enabled and set to 1. The batch size was set to 4096. These three parameters are closely related in that they will determine how quickly the models can learn. Various batch sizes and learning rates were tested; however, the effects on accuracy were negligible. Lower learning rates and smaller batch sizes caused training to take longer, while precision did not change in terms of MSE and MAE.

Six months worth of data from 1000 cells was used to train the network. Each KPI, including temperature, was scaled to zero mean and unit variance in a global context. Standardization has been shown to outperform Min-Max scaling for LTE KPI forecasting [37]. In other words, all cells were scaled according to global mean and variance. This decision was made to allow the models to learn global patterns, such as when a cell is not experiencing a high load versus when it is experiencing a congested state. The training and validation splits were temporal. The first 80% of the data were used for training, while the remainder was used for validation.

The training was set to run for a maximum of 100 epochs. Early exit conditions were created for the event when the validation accuracy did not decrease by 5E-4 for two epochs. This value was determined empirically. Most models had fully converged after two to three epochs. Convergence at this point was no surprise given the use of global scaling and the amount of data being considered.

4.3 How Long is Long Enough?

Characterizing the performance of forecast model architectures requires testing many different configurations of the input sequence length, output sequence length, and internal dimensions of the model. Most neural network forecasting studies present results similar to those presented in Table 3.4. Benchmarking the performance of a model on standardized datasets leads to specific input and output sequence lengths being selected; however, for neural network application tuning, the inverse is required. For a given application, the forecasting horizon is generally fixed. There are cases where autoregressive forecasts can be used to generate arbitrarily long sequences; however, in general there are some arbitrary number of steps in the future to predict. With a fixed horizon, the length of the input sequence can be adjusted to optimize accuracy. Forecast accuracy is important; however, it is also important to be aware of the trade-off of longer horizons and increased computational cost. An infinitely complex forecasting model might be able to achieve perfect accuracy, but ultimately the forecast will not be available to a network operator during the time when it is needed.

Based on the ACF, PACF, and practical computational limits, the maximum input sequence length will be limited to 96 steps, which corresponds to one day. For each model, the input sequence lengths 4, 8, 16, 32, and 96 were used. Short temporal dependencies have been identified in rural fixed wireless networks; however, input sequence length variations were not tested [37].

A mixture of DMS and IMS forecasting models was selected to test input sequence length variations. For DMS models, N-HiTS, DLinear, and NLinear were chosen. The selected IMS models were GRU, Transformer, Seq2Seq, and Seq2Seq+Attention.

Unlike the forecast performance analysis performed in Chapter 4, where the objective was to compare the accuracy of different models on different time horizons with or without covariates, this analysis examines the effect of the length of the input sequence between models. The performance results for each architecture and parameter configuration can be found in Table 4.1

Horizon	4		8		16		32		96	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Model										
DLinear	0.146	0.220	0.145	0.220	0.145	0.220	0.145	0.220	0.142	0.218
GRU	0.142	0.218	0.141	0.216	0.141	**0.216**	**0.140**	0.217	0.138	0.214
N-HiTS	**0.140**	0.216	**0.139**	**0.214**	**0.139**	**0.216**	**0.140**	0.221	0.138	0.216
NLinear	0.144	0.218	0.144	0.219	0.146	0.221	0.145	0.220	0.140	0.217
Seq2Seq	0.142	0.218	0.141	0.217	0.141	0.217	**0.140**	**0.216**	**0.137**	**0.213**
Seq2Seq+Attention	**0.140**	**0.215**	0.142	0.218	**0.139**	0.218	**0.140**	**0.216**	**0.137**	0.216
Transformer	0.142	0.219	0.142	0.221	0.142	0.222	0.142	0.222	0.140	0.221

Table 4.1: Forecasting performance of different models across different input sequence lengths. Best results by horizon are in bold.

Overall, the best performing architecture was found to be N-HiTS. It was top-performing in six out of ten possible metrics-horizon pairs and was close in the remaining four columns. Surprisingly, the Seq2Seq-based models were the second best model across all horizons.

Seq2Seq and Seq2Seq+Attention were expected to perform well; however, the variation in the input sequence length highlights that Seq2Seq+Attention outperforms a regular Seq2Seq model on most horizons. A similar study had considered an input sequence length of eight [37]. Across more horizons it can be seen that the attention mechanism marginally improves performance.

Increasing the length of the input sequence was found to improve performance across all architectures. On average, MSE and MAE were found to decrease by 2.4 and 0.59%, respectively. Ultimately, the use of these forecasts will depend on how constrained a practical deployment condition will be; however, selecting N-HiTS with an input sequence length of 4 will be within 2% of the best trained model across all input sequence lengths. In an ideal environment, even longer histories could be considered; however, these results agree with the short temporal relationships highlighted by the PACF analysis. Increasing the history beyond a handful of timesteps will ultimately not increase the forecast accuracy, regardless of the model architecture. Longer historical time steps will primarily increase the computation time.

4.4 Implication of Covariates

The amount of information contained within a single target feature signal is limited. Multivariate forecasting generally allows information from one target signal to be shared with others. This can intuitively be seen in models such as the Linear family, where all inputs get mapped to all outputs and in Seq2Seq models where all signals are compressed into a context vector.

The information shared between signals can help improve accuracy; however, there are sometimes additional features that can be added to further improve accuracy. For example, forecasting DL-PRB and DL-THRP together is intuitive, in that they are intimately related. Inclusion of exogenous features such as temperature or snow, as presented in Chapter 3, can further improve accuracy. In this chapter, the inclusion of environmental features is considered; however, snowfall and snow depth are foregone in favor of only temperature. Similar urban studies suggest that a reduction in RMSE above 40% is achievable by including covariate signals, including weather [7].

Environment Canada provides weather data every hour for most weather stations throughout the country. The considered wireless network collects KPIs every 15 minutes. This is problematic in that the KPI data should be kept to the finest granularity possible for short-term forecasting. Linear interpolation was used to resolve the sampling rate mismatch between the hourly environment and 15-minute KPI signals.

Similarly to Section 4.3, the lengths of the input sequences were varied;

however, temperature data was also included. The results can be found in Table 4.2.

	Horizon	4		8		16		32		96	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Model	Env.										
DLinear	False	0.146	0.220	0.145	0.220	0.145	0.220	0.145	0.220	0.142	0.218
	True	0.145	0.220	0.145	0.221	0.144	0.220	0.145	0.220	0.143	0.220
GRU	False	0.142	0.218	0.141	0.216	0.141	**0.216**	0.140	0.217	0.138	0.214
	True	0.142	0.218	0.141	0.218	0.140	0.217	0.140	0.215	0.138	0.215
N-HiTS	False	**0.140**	0.216	**0.139**	**0.214**	**0.139**	**0.216**	0.140	0.221	0.138	0.216
	True	**0.140**	0.216	0.140	0.218	0.140	0.218	**0.139**	0.217	0.138	0.219
NLinear	False	0.144	0.218	0.144	0.219	0.146	0.221	0.145	0.220	0.140	0.217
	True	0.144	0.218	0.143	0.218	0.145	0.220	0.145	0.220	0.140	0.217
Seq2Seq	False	0.142	0.218	0.141	0.217	0.141	0.217	0.140	0.216	**0.137**	**0.213**
	True	0.142	0.218	0.141	0.216	0.140	0.217	0.140	0.217	**0.137**	**0.213**
Seq2Seq+Attention	False	**0.140**	**0.215**	0.142	0.218	**0.139**	0.218	0.140	0.216	**0.137**	0.216
	True	**0.140**	0.216	0.142	0.217	**0.139**	**0.216**	0.140	0.218	**0.137**	0.214
Transformer	False	0.142	0.219	0.142	0.221	0.142	0.222	0.142	0.222	0.140	0.221
	True	0.143	0.219	0.142	0.218	0.140	0.219	**0.139**	**0.214**	**0.137**	0.214

Table 4.2: Forecasting performance of different models across different input sequence lengths and environmental covariate inclusion. Best results by horizon are in bold.

For an input sequence length of 4, the change in MSE and MAE was either zero or 1E-3 when including temperature data as a covariate. For an input sequence length of 8 some changes were introduced by including temperature data; however, it is almost universally a decrease in forecast accuracy. For 16 input steps and higher, the accuracy of the forecasts was unchanged or marginally better. In some cases, the best model was a tie between the same architecture with and without temperature data; see Seq2Seq with an input sequence length of 96.

In contrast to urban results, the addition of temperature data negatively or negligibly affected MSE and MAE. This agrees with a similar rural study [37] while disagreeing with urban results [7]. Furthermore, the addition of covariates increases the complexity of the data collection system. Using only LTE network KPIs allows all data to be sourced from a single location. Unless the network operator collects their own environment data, they will need to rely on an external weather collection service. If there is a delay from the external weather service, then that can delay the ability to generate a forecast. This will further constrain the computation deadlines of a forecasting system without providing a significant increase in accuracy.

4.5 Conclusion

Forecasting LTE KPIs has received significant attention in urban mobile networks, leaving rural fixed wireless networks largely untouched. This has left a significant gap in knowledge about the temporal behavior, statistical properties, and forecasting performance of rural fixed wireless networks. Urban mobile networks have been shown to have long-term temporal relationships [11], while rural fixed wireless networks have been shown to have the opposite [37].

Temporal analysis indicated that most of the network KPIs in all measured cells were stationary in some sense. ACF and PACF analysis identified that the long-term relationship for the KPIs is minimal. Beyond the first few lags, the relationship is no longer statistically significant. There are several points around the 96-lag mark that are significant; however, their magnitude is very small.

Using the identified temporal relationships, several lengths of input sequence were used to evaluate the ability of various forecasting models to capture temporal dependencies. Longer input sequences were found to achieve better performance; however, the improvement in performance was almost negligible. This suggests that using the last hour of KPI data to predict the next 15 minutes is likely the best option for practical forecasting tasks.

Exogenous features continue to be a popular approach to further improve the forecast error. Urban mobile studies have suggested that including environmental features such as weather and spatial information can improve performance [7, 12]. A similar set of environmental features, namely temperature, snowfall, and snow depth, has been determined to provide no tangible performance benefit [37]. This work considered the removal of snowfall and snow depth from the selected set of covariates in favor of only temperature. The inclusion of temperate degraded the accuracy of the forecast in some horizons. In summary, including environmental covariates in rural fixed wireless LTE KPI forecasting were not found to provide a similar performance improvement to that found in urban mobile networks.

The focus of the work presented in this section was on forecasts that consume multiple step input sequences and generate single step output sequences. A natural continuation of this work would be to consider output sequences longer than one time step. Furthermore, as the number of input and output steps continue to grow multiple seasonalities could be encountered in the KPI signals. Daily and weekly seasonalities could be removed and handled as additional covariates using STL [41] or MSTL [42]. It is also anticipated that neural network architecture like Seq2Seq and GRU will achieve worse performance relative to more optimized architecture like N-HiTS or DLinear as the number of input and output sequence steps grows.

Chapter 5

Robust Anomaly Detection

Long term capacity planning is critical for network operators; however, it is equally important to provide quality service at present. Each new generation of wireless networks has introduced a new level of complexity. In addition to this, the growing expectation, dependence, and requirement of subscribers to have access to quality internet service makes outages and degradation of service problematic. LTE and 5G networks have built-in alarm systems that are capable of detecting and correcting certain events, but the capabilities of these fault management systems are limited. Delicately set thresholds provided by network equipment are generally the primary monitoring service that a network operator uses.

A major risk of threshold-based alarms is *alarm fatigue* where the sheer volume of alarms causes those who monitor the alarms to become desensitized. If an alarm goes off, it must have meaning; otherwise, it should not have gone off. It is inevitable that this occurs with naive alarms. Managing thousands or millions of alarm thresholds across the equipment required to support a cellular network is not viable. The classification of severity is one way this can be addressed; however, more thresholds can potentially result in more meaningless alarms.

This chapter introduces RAINFOREST (Robust Anomaly detectIoN FOr Rural fixEd wireleSs lTe), an AD algorithm that leverages temporal context for explainable anomaly detection while being tunable and providing anomaly intensity estimation. Furthermore, it is unsupervised, meaning that the involvement of network experts in model optimization can be greatly reduced. Section 5.1 provides a high-level overview of RAINFOREST and the problems it aims to solve. Section 5.2 describes each individual component of RAINFOREST and the purpose for which it is used. Section 5.3 benchmarks RAINFOREST against popular baseline AD methods and demonstrates the AD capabilities on a known problem.

5.1 Why RAINFOREST?

Enormous collections of historical data across an entire network have made wireless networks exceptionally good candidates for AI, ML, and Big Data approaches to network management [5]. AD is a key area of network management, as service interruption or poor quality can be frustrating. Although deep learning is rapidly developing advanced models, one of the largest problems remains how to identify anomalous behavior. The number of signals that a wireless network generates is unmanageable in terms of the potential for data labeling. Supervised and semi-supervised methods have been proposed [96, 72]; however, generalization of models with any supervision is a concern due to the number of uncharacterized anomalous events that could occur [97].

Supervised methods are less popular for time series anomaly detection because of the number of unique types of anomalous event that can occur. Furthermore, AD methods that are capable of operating in an unsupervised multivariate context are more complex, however, more desirable [73]. Many AD applications are multivariate in nature, especially those that consider wireless networks. Capturing the complete temporal dynamics of a cell using a single variable is unlikely.

One of the key objectives of this work is to achieve robust AD in an interpretable way. Pure deep learning approaches can result in explanations that are not intuitive. The output of the RAINFOREST is an interpretable report of the detected anomalies; however, the results are presented in such a way that it can be used to guide network operators in identifying the root cause of the anomalous behavior. This work does not aim to classify the types of anomaly detected and instead focuses on the explainability of the detection process itself. RAINFOREST is a hybrid multivariate AD algorithm that uses forecasting, clustering, and statistics to overcome the deficiencies of each of the composing methods.

Forecasting and reconstruction AD methods can achieve excellent results without additional downstream processing. For univariate forecasting, the output is generally highly interpretable. A forecast or reconstruction is created, and the error is analyzed using a metric such as the z-score or some other statistical measure. Under the assumption that the output error of the model is noise, this can be a robust approach. When expanding this process to a multivariate problem, analyzing the distribution can become difficult as the number of dimensions grows. RAINFOREST aims to solve this dimensional analysis problem by using clustering to identify anomalies.

Clustering or density-based approaches are very popular for anomaly detection. KNN [13] and K-means [96] have been used for AD in wireless networks. A common problem with density analysis is to select the optimal density and to have the ability to reject noisy samples. The inability of KNN and K-means to natively classify abnormal samples as noise reduces their value in an AD context. To date, DBSCAN or other density-based clustering has not been applied directly to wireless network AD. Clustering solves the multidimensional problem of forecasting and reconstruction methods by considering the proximity of points to each other. Drift is a known problem with clustering methods; however, the assumption that the forecast is accurately generated resolves this problem by adding temporal context. For example, if there is a large traffic burst in a cell at 2 am, it could be an anomaly, or it could simply be computers running scheduled updates. If it were a one-off event, then a forecast would flag it as anomalous; however, if it occurs everyday at 2 am, then the forecast should allow it not to be as significantly anomalous.

Finally, statistical analysis is used after clustering analysis for anomaly intensity purposes. It is important to note that RAINFOREST does not use statistical analysis to detect anomalies. The binary classification of a sample as anomalous or not is performed upstream by clustering analysis of the forecast errors. If a given sample is identified as anomalous, then statistical analysis is used to determine the intensity of the fault. For example, a sample with a z-score of 1 is likely not anomalous, while a z-score of 4 indicates that it is very likely anomalous.

Using a combination of forecasting, clustering, and statistical analysis, RAINFOREST is able to overcome many of the drawbacks of each of the included methods. It is also highly modular in nature and can readily support swapping of the comprising elements. For example, a forecasting model based on ARIMA could be easily swapped to a deep learning approach such as N-HiTS. Alternatively, DBSCAN or another clustering algorithm could be exchanged for HDBSCAN or OPTICS. Another notable feature is that RAINFOREST is highly tunable. These benefits of RAINFOREST are discussed in more detail in later sections.

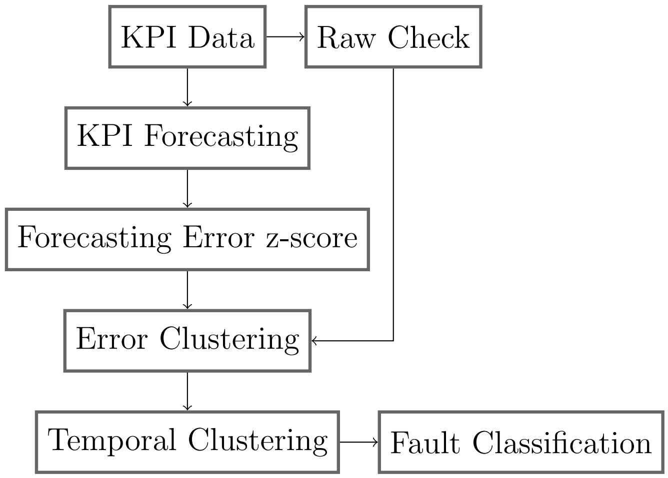

Figure 5.1: Data flow chart for RAINFOREST

5.2 RAINFOREST

As an AD method, RAINFOREST is highly interpretable. A block diagram of the AD procedure can be found in Fig. 5.1. The AD process is relatively direct and cascades from one analytical component to the next. The only exception to this is a skip connection for a *Raw Check* which can be used to add rule-based logic to the algorithm. Each of these unique components are broken down in more detail throughout the remainder of this section.

5.2.1 KPI Data

Ingestion of the KPI data is one of the most critical aspects of any AD process. In AD tasks such as fraud or defect detection, the timelines are generally more relaxed. Some time series AD tasks are much more challenging in terms of strictness of deadlines because of the ramifications of timely detection. For LTE network monitoring, it is critical that anomalous behavior is detected quickly so that corrective action can be taken.

For this application, KPI data are collected every 15 minutes for the entire network. Some network vendors provide direct database access, while others provide data through a standardized XML interface. Extract, transform, and load (ETL) data pipelines are a key component of the anomaly detection process. Although they are not directly related to the anomaly detection process itself, it is important to be aware of upstream processing delays before *clean* data can be fed into a forecasting and AD model. For example, data for the period of time from 00:00:00 to 00:14:59 would become available shortly after 00:15:00. Data must then be fed through ETL pipelines, which can take a few more minutes. After these pipelines, a forecast must be generated either directly for RAINFOREST or pulled from an external source, which introduces further delays. Essentially, the sampling rate may be 15 minutes; however, in reality, the time-to-act is closer to several minutes.

Any number of LTE KPIs could be considered for RAINFOREST. Although it is possible to consider a large number of KPIs with a long history, it is important to note that more data will increase the computation time. This results in the need for more hardware to support the AD process or in more missed AD deadlines. It is also important that the selected KPIs accurately describe the overall performance of the cells being monitored. Due to these conditions, a total of five network KPIs were selected: DL-PRB, DL-THRP, UL-PRB, UL-THRP, and AUE. When combined, they allow key metrics such as spectral efficiency and traffic per user to be evaluated.

5.2.2 Raw Check

The primary focus of RAINFOREST is to detect anomalies in an unsupervised manner; however, there are some anomalous events that can be easily handled by rule-based AD methods. For example, if there are no users in a cell, an anomalous event likely occurred. There are some special cases, such as when a cell is undergoing maintenance or when it has just been installed and there are no users connecting to it; however, these cases are well known and can be filtered out. Furthermore, the raw check allows the sample to bypass the clustering method. In other words, if a rule determines that a sample is anomalous, the sample will be labeled as anomalous regardless of the status determined by the clustering method. This allows data to be fed into statistical analysis to determine the intensity of the fault.

For new cells with only a few users, the detection of no-users could be flagged as an anomaly; however, it is unlikely to be registered as a significant fault. On the contrary, if a more mature cell with many users suddenly has no users, the statistical analysis will indicate a significant change in behavior and identify it as a fault. Although the raw check only considers the detection of no users in a cell, additional rule-based logic could be added depending on the requirements of the network operator.

5.2.3 KPI Forecasting

LTE KPI forecasting has been addressed in both long- and short-term contexts in Chapters 3 and 4. Long-term behavior is of interest; however, the focus of this work is on short-term AD. Namely, some of the objectives of this AD work is for dynamic resource allocation or anomalous configuration changes to be detected in a timely manner.

It is assumed that the forecasting models are accurate. There is no threshold set for the level of accuracy required; however, less accurate forecasts ultimately result in worse performance of the anomaly detection system. The forecast generation technique is also irrelevant. A practical implementation of RAINFOREST in a commercial network checks for anomalies in all samples as they arrive. Since the process is inherently single-stepped, the generated forecast should be single-step. This means that DMS or IMS forecasting strategies can be used without encountering the long-horizon drawbacks of either approach. One architecture may outperform another; however, the accumulation of bias as the number of forecast steps does not affect the IMS models.

Ideally, the forecast is stored in a database to be queried when it is needed. If this is not the case, it is possible to generate a forecast in real time to meet the forecasting data requirement. Although this is a work around, it is still important to note that generating forecasts on the fly may be suboptimal compared to being able to batch and accelerate their generation using centralized GPU resources.

5.2.4 Forecasting Error z-score

One of the most important aspects of forecast error analysis is the ability of input distributions to be transformed from nonparametric to parametric. LTE KPIs are inherently noisy and nonparametric; however, the forecast error distributions have been found to be Gaussian. Some general assumptions can be made about traffic, such as high in the evening and lower during the day. There is a general daily trend across many of the KPIs; however, depending on the cell load and time of day, there can be significant fluctuations within and between cells. For example, a new cell could have a distribution of DL-PRB that is heavily weighted closer to zero, while a more mature cell could be saturated at 100% PRB use. As a cell transitions between these two states, the distributions can vary significantly due to user behavior.

Since the scale of the KPIs can vary from cell to cell, it is important to normalize the scales of the inputs prior to further analysis. The change in distribution between the raw and forecast error distributions can be seen in Fig. 5.2.

Using the Komolgorov-Smirnov (KS) test with a zero-mean unit variance Gaussian distribution as the target distribution, it was determined that the KPIs were less normal than the forecast error. Although normality is not a requirement, transforming a nonparametric distribution into a parametric one can better facilitate generalization in the analytical process. In other words, the analysis can leverage Gaussian assumptions rather than being

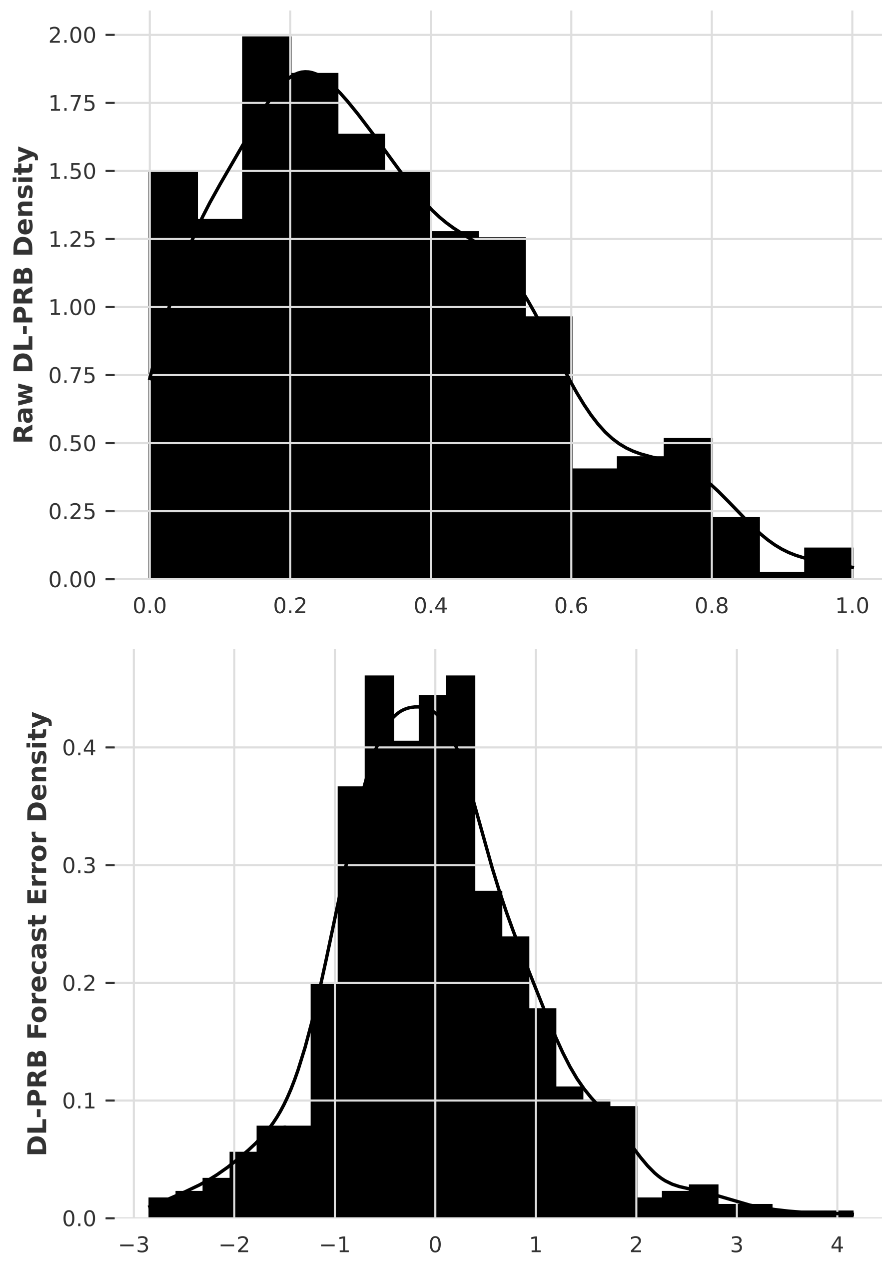

Figure 5.2: Comparison of the probability density function estimate for raw and forecast error. Forecast error was standardized to zero mean and unit variance while the DL-PRB was scaled from zero to unity.

forced to operate on a nonparametric distribution.

5.2.5 Error Clustering

The clustering components of RAINFOREST are the most computationally complex. There are many ways in which clustering can be approached; however, the objective is ultimately to group data that are similar while identifying data that are not.

Existing studies have largely considered K-means [15] or KNN [13] for clustering in AD for communication networks. This is problematic in that the size of the clusters is unbounded and parameter tuning does not provide a meaningful indicator of the properties of the data. For example, a point belongs to a cluster so long as it is closest to the mean of that cluster. This means that a point could be 10-times farther away from any other point; however, it still receives a valid label.

Furthermore, K-means clustering for AD is problematic because an arbitrary cluster can be chosen to be anomalous. AD methods have considered this exact case where a cluster was declared anomalous when, in fact, it does not appear to be explicitly anomalous [15]. Approaches that consider the distance to neighbors using KNN distributions are more robust; however, they ultimately need to choose a statistical threshold to declare a sample as anomalous or not [13].

For RAINFOREST, this issue is resolved by leveraging DBSCAN. As discussed in Chapter 2.4.3, DBSCAN allows points to be labeled as noise based on the condition that they do not meet a density threshold. In other words, if the point is too far from its neighbors, it is noise. A high-level comparison of the results obtained from K-means and DBSCAN can be seen in Fig. 2.9. A key example is the sample near (-2, 0). K-means is content to place this in cluster 2; however, DBSCAN identifies this point as anomalous since it is far away from its neighbors.

It is not an explicit requirement that DBSCAN be used; however, since computational hardware is a finite resource, the options of the density-based clustering algorithm are limited for use in RAINFOREST. If computational efficiency was not a consideration, alternative techniques such as OPTICS [98] or BIRCH [99] could be used instead. Initial testing considered these clustering techniques, but the increase in run time was an order of magnitude, which resulted in them being dropped from consideration.

A comparison between the distribution of the DL-PRB and DL-THRP error versus the raw KPI data can be found in Fig. 5.3. The raw KPIs have a multidimensional nonparametric distribution, making analysis particularly difficult. As discussed and shown in Fig. 5.2, the forecast transforms the distributions into roughly normal distributions, which allows for easier analysis. The forecast error scatter plot clearly takes on an elliptical shape compared to the raw KPIs. Furthermore, the forecast error allows the temporal context to be captured. In other words, using the forecast error allows unexpected events to be analyzed rather than raw changes.

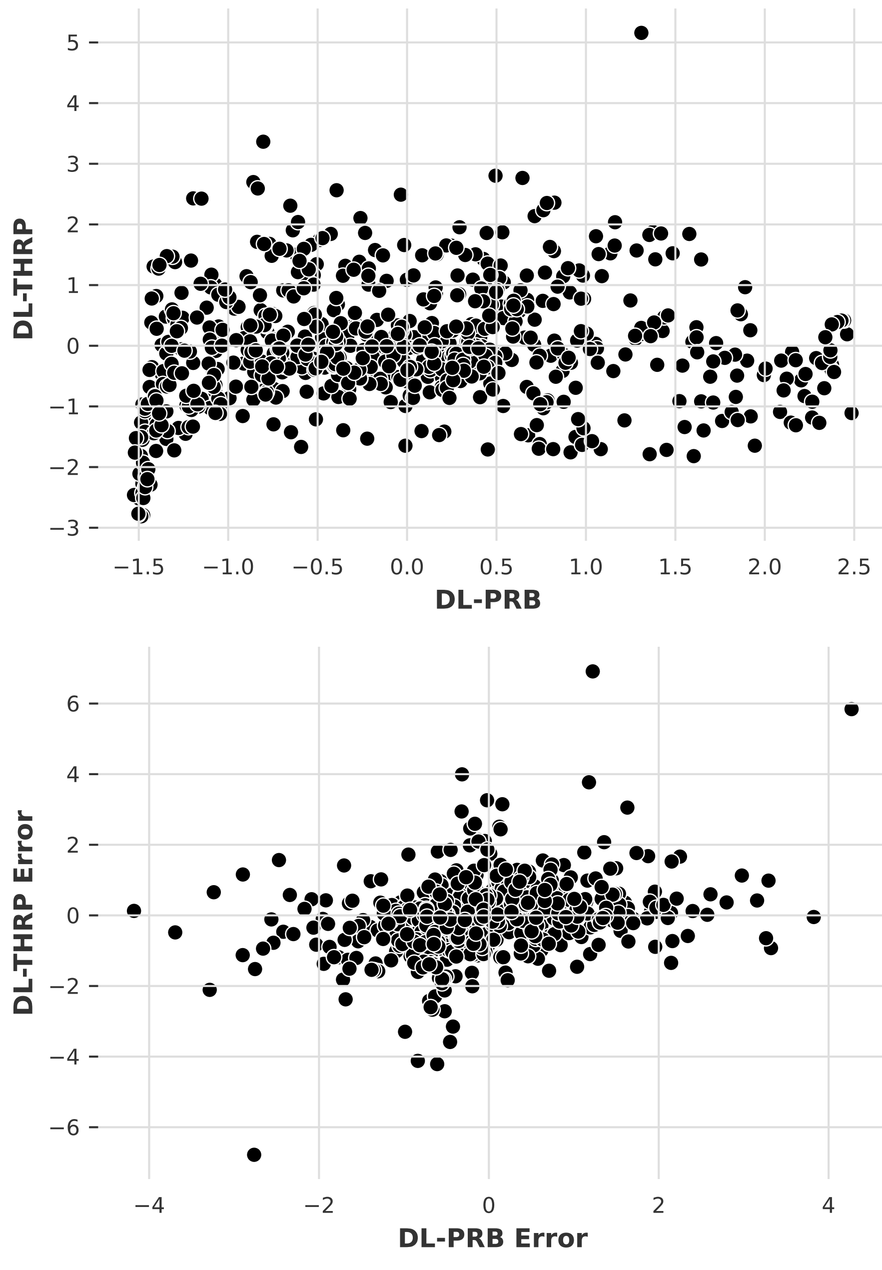

Figure 5.3: Scatter plot comparison between the raw KPI values and the forecasting error for DL-PRB and DL-THRP. Axis are standardized to zero mean and unit variance.

5.2.6 Temporal Clustering

A simple but important part of RAINFOREST is the ability to cluster outages in time. Alarm fatigue can be a major problem. In addition to fault intensity ranking, the ability to cluster outages in time allows less outages to be reported. For example, if a cell had a configuration change that caused all users to disconnect for several hours, flagging each time step as a unique outage would raise many more alarms than if they were grouped in time.

Unlike some AD methods, RAINFOREST does not explicitly consider temporal patterns or shapes in the data, sometimes referred to as *motifs* [100]. The decision for RAINFOREST to be a motif-free AD method was made largely due to the fact that cells exhibit dramatically different behavior based on loading and that the forecasting component of the model captures temporal context, which is similar to why motifs are used. Motifs are interesting in that the start of a pattern could be found to identify when a recurring type of outage pattern was detected. A further extension to RAINFOREST could be the inclusion of motif analysis in error signals to find recurring anomalous forecast error patterns.

Since RAINFOREST does not have the ability to leverage motifs, the data is simply clustered in time. Once the data are grouped in time, they are then ranked according to the worst samples in the group. This is a unique approach in that a cell could go offline for a few reporting periods and then come back online. All users are not expected to reconnect immediately and anomalous behavior could continue for several hours after the initial event.

Density-based clustering is used for temporal grouping. Since the input to the clustering problem is a one-dimensional array of timestamps, clustering analysis using DBSCAN were effectively operating like a sliding window that checks if the points are close enough to be clustered together. The DBSCAN density was configured so that the samples should be within the same one-hour window. The minimum number of points required to be identified as a cluster was set at two. This is important for the fault classification aspects of RAINFOREST, which are discussed later in this chapter.

5.2.7 Fault Classification

Once anomalous samples have been identified, the process of classifying the intensity of each anomaly can be performed. An important consideration is how abnormal the anomalous sample is. RAINFOREST does this in two different ways by addressing the duration and statistical likelihood of the anomaly itself. RAINFOREST classifies anomalous events into three unique categories: noise, soft, and hard faults.

Noise faults are transient in nature and are self-resolving. That is, a network operator does not need to take corrective action. For example, a sudden spike in traffic from scheduled computer updates in many users connected to a cell is not a concern. Noise faults are those that are identified by temporal clustering as being disconnected from other anomalous behavior. The intensity of noise samples is also ignored from their classification because they are self-resolving in nature. In other words, they have a duration of only one sampling period before returning to normal behavior.

Soft and hard faults are closely related in that they both require multiple anomalous samples to occur in the same one-hour window. The difference between the two is that soft faults are below what is called the *hard-soft* threshold and hard faults are above it. This threshold considers the Euclidean distance from the high-density clusters to the potential outlier sample. Based on the clustering density for DBSCAN, the samples labeled as noise are already required to be isolated from the cluster. This threshold is used to determine how severely they are separated. For more advanced analysis, the continuous value is acceptable; however, when relaying the intensity of a fault to a network operator, having discrete values for intensity reduces the knowledge burden on the end users. Continuous value outputs are helpful for comparing the relative intensity of detected anomalies; however, network operators would need to understand the scoring metrics intimately to make informed decisions. The thresholding used alleviates this burden.

The output of the fault classification component of RAINFOREST is noise, soft, or hard. It should be noted that the objective of RAINFOREST is to detect anomalous behavior. It is assumed that RAINFOREST would present both discrete and continuous scoring metrics to a separate model or an expert network operator for root cause analysis. For example; RAINFOREST could identify that there was a hard fault that occurred. The anomaly scoring and ranking system of RAINFOREST could then be used to determine that there was a sudden change in spectral efficiency (DL-THRP went down, DL-PRB went up); however, this analysis would be external to RAINFOREST itself.

5.3 Performance Analysis

Characterizing the performance of RAINFOREST is no simple feat because it is not supervised and there is no set of labeled anomalies to compare with. Statistical measures were considered, namely EM and MV as discussed in Chapter 2.4.1. Furthermore, a known anomaly was analyzed using RAINFOREST to demonstrate the early detection capabilities relative to the built-in network alarming features.

5.3.1 Network Scale EM and MV Analysis

One of the main difficulties in unsupervised AD is how to objectively measure the performance of a detection method. A small set of data with known anomalies could be labeled, or network logs of detected anomalies could be used to obtain labeled data. This could allow an unsupervised method to be trained, tested, and then objectively measured. Although this is a promising idea at first glance; however, the assumption that built-in network alarming is accurate is a poor assumption. Sleeping cells are cells that fail without triggering alarms and are an active area of research due to the commonplace of their occurrence [72]. Furthermore, assuming that manually labeling data would capture all types of anomalies is a poor assumption. The labeling process would not only be time consuming but would also be prone to human bias and error.

EM and MV have been proposed as a way to quantify the performance of an unsupervised AD method. To evaluate EM and MV, a scoring function is required as described by Goix [74]. These scoring functions are used to create estimates of the density level set which are then generally assigned to a binary scoring function that indicates whether a sample is anomalous or not. For RAINFOREST, the scoring method is the distance between the sample of interest and the nearest core DBSCAN point. In other words, the high-density regions are not anomalous, whereas the low-density regions are anomalous. The hard-soft threshold is then used to determine the intensity after DBSCAN determines whether a point is noise or not.

Robust baseline models are important for any unsupervised analysis of AD performance. Three common benchmark models are considered: local outlier factor (LOF) [101], one-class support vector machine (OCSVM) [102], and isolation forest (IF) [103]. When determining the best performance for a given set of data, it has been proposed that the agreement between EM and MV indicates which model is best. This was determined by comparing the results of EM and MV with the precision recall and receiver operating characteristic curves on the benchmark data sets [74]. The objective is to maximize EM while minimizing MV.

To test the performance of RAINFOREST against baseline models, two weeks of network data was used. A difficulty with comparing the baseline is that the temporal context is very critical to the overall AD process. If a spike occurs every day due to scheduled updates or synchronization of cloud data, then it should not cause a major fault. To increase the fairness of comparison, the baseline models consider the forecasting error like RAINFOREST.

The results of the unsupervised classification for the cell set can be found in Table 5.1. RAINFOREST is clearly the top performing model across all metrics. RAINFOREST's dominance was expected, since it operates directly on the density of the cell KPI data. Using the clusters established by DBSCAN, new samples can be easily scored based on their distance from the high-density core points.

	Best EM (%)	Best MV (%)	Best EM and MV (%)
RAINFOREST	98.57	99.97	98.57
LOF	1.23	0.00	0.00
IF	0.08	0.00	0.00
OCSVM	0.12	0.03	0.01

Table 5.1: Performance of AD methods based on EM and MV metrics. Best EM indicates what percentage of cells had the highest EM for a given model type. Best MV indicates what percentage of cells had the lowest MV for a given model type. Best EM and MV indicates when a model had both the highest EM and lowest MV for a given cell. Note that the EM and MV columns sum to 100% while Best EM and MV do not since there is not always agreement between models for a cell.

LOF and IF operate in a similar manner to DBSCAN in that they roughly approximate the point density and assign labels based on some external metric. The LOF metric is based on the density estimate of a given sample relative to that of its neighbors. IF builds a tree that aims to segment the sample space where anomalous samples generally result in shorter tree depths.

When comparing the anomaly detection process, OCSVM is closest to RAINFOREST, which is not surprising given the results. RAINFOREST identifies the high-density region in the sample space and then identifies the samples as anomalous as they move away from the identified regions. OCSVM achieves a similar result by drawing a hyperplane that segments anomalous samples from those considered normal [102].

From the EM and MV results presented in Table 5.1, RAINFOREST is a significant performance improvement over the commonly used baseline models.

5.3.2 Known Anomaly

Thus far it has been shown that RAINFOREST outperforms common baseline AD methods; however, it is also important to demonstrate the benefits of RAINFOREST beyond the analysis of the performance of EM and MV. RAINFOREST can detect anomalies in an unsupervised manner better than baseline methods, but also provides ranking and temporal context which bring a large amount of value to network operators. This analysis of a known anomaly is analyzed to demonstrate that RAINFOREST is able to detect anomalous behavior earlier than the built-in LTE AD capabilities.

Leveraging the forecast allows one to add a temporal context to the AD process. LTE and 5G systems often use threshold-based alarms. In general, more robust unsupervised AD methods set thresholds as well; however, they are based on a probabilistic scoring function [74]. Using the forecast error allows for a dynamic threshold to be set. In other words, the forecast error allows the theoretically known information to be removed, leaving only the unexpected information. Under the assumption that the forecast is accurate, it can be assumed that the forecast captures all predictable information and the error represents only the unexpected information. Since anomalous behavior is assumed to be rare when considering a large number of samples, unpredictable input samples lead to the forecast error increasing abnormally. If the error is sufficiently abnormal, then the samples end up in a low-probability region and be flagged as anomalous.

The anomaly presented in this section occurred in April 2022. RAINFOREST operates under the assumption that forecasts are generated external to the model; however, it is important to note that the forecasting model was not exposed to any of the network during the time in which the anomaly occurred. This was done to ensure that no data leakage had occurred.

N-HiTS was used to generate the forecast. The forecast was a single-step multivariate forecast model without covariates. Any model architecture or forecasting strategy can be used as long as the forecast is accurate. The accuracy in this context is used to describe how much information about the future the model can extract from the input sequence. The model could be exchanged for other models such as GRU, Transformers, or DLinear and could achieve nearly identical results.

The analysis of the known anomaly can be seen in Fig. 5.4. Two critical times in the presented event were when the LTE alarms alerted the network operator of the fault and when RAINFOREST was able to detect a significant anomaly for the first time. Midday on April 21, 2022 the network operators noticed alarms being raised by the cell being analyzed. One of the problems with the threshold-based alarm scheme is that anomalous behavior does not always raise the alarm. The considered event is an excellent example, since RAINFOREST was able to detect the anomaly nearly 24 hours earlier. This includes the hard faults identified mid to late on April 20, 2022, when a significant number of users disconnected and traffic-related KPIs were heavily affected. Intuitively, fewer users will generate less traffic, leading to lower cell throughputs and PRB use; however, it can be seen that UL-THRP and UL-PRB increases despite a decrease in users. Once the network operator was alerted to the problem and took action, it can be seen that the anomalous behavior stops.

According to the network logs, the anomaly was caused by a power configuration problem in the cell. A noteworthy aspect of RAINFOREST and the selected subset of network KPIs is that despite the fact that no metric is related to cell power levels, the selected KPIs are able to accurately capture the anomalous event. This is promising in that the dimensionality of the input space to RAINFOREST or any other multivariate AD method

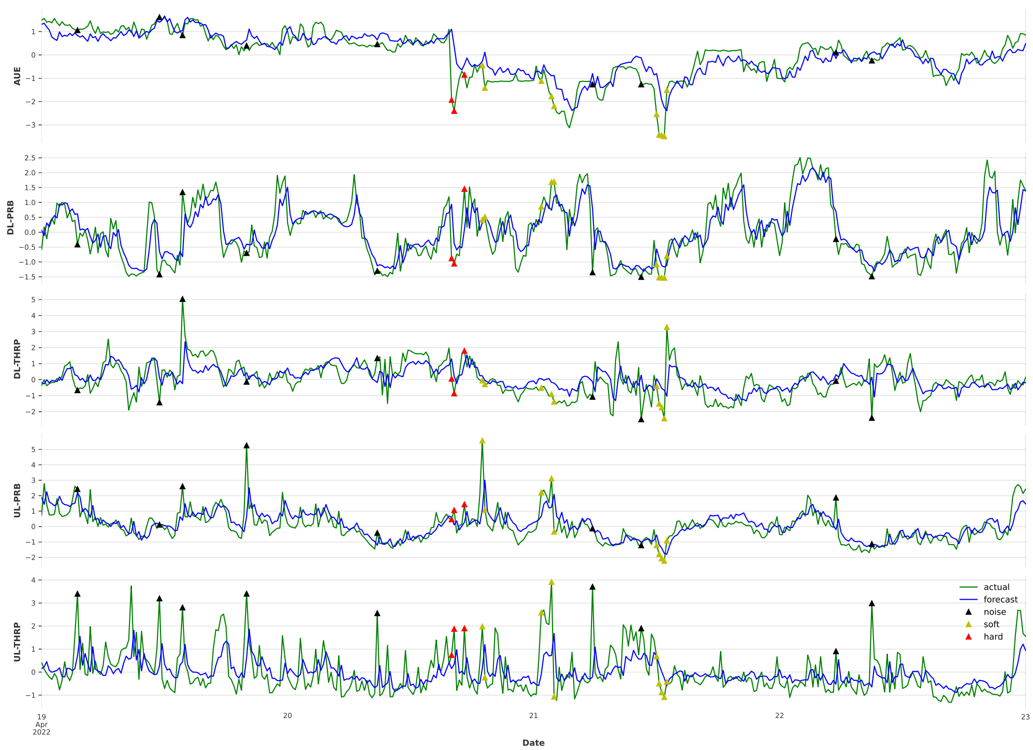

Figure 5.4: RAINFOREST AD and intensity scoring applied to a known outage. Anomalous behaviour was identified by built-in alarms around 12 pm on April 21, 2022.

can be kept constrained to a small number. This reduces the difficulty of achieving real-time online anomaly detection.

Although RAINFOREST is focused on analyzing one cell at a time, the impact of a single anomaly can easily be tracked to other nearby cells. There are many adjacent cells to the originating one; however, the analysis was limited to one for brevity. RAINFOREST analysis can be found in Fig. 5.5. The second cell has more hard faults that are raised compared to the originating cell. It should also be noted that many of the AUE that were disconnected from the initial problem migrated to this cell. Users are clearly hopping between cells in a relatively aggressive manner, which leads to a poor end-user experience.

Using RAINFOREST to track single-cell anomalies has been demonstrated to great effect. Numerous anomalous behaviors can be tracked in a single cell; however, there is no explicit relationship between the cell being analyzed and its neighbors. Although this work does not consider it, graph-based approaches could be leveraged to add a spatial context in addition to RAINFOREST's existing temporal context. SDGNet [104] has been shown to improve the accuracy of forecasts in an urban mobile network by adding spatiotemporal features through a graph. Including a graphical approach in RAINFOREST would increase computational complexity; however, it would allow better tracking of behaviors such as user migration or load balancing.

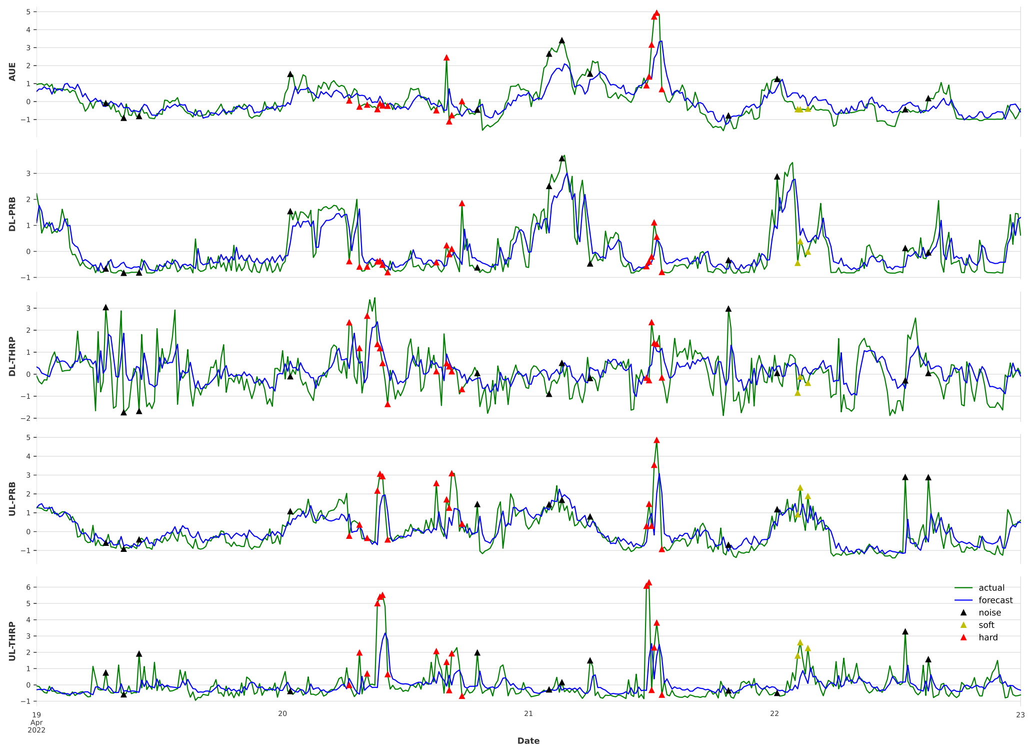

Figure 5.5: RAINFOREST AD and intensity scoring applied to a cell adjacent to a known outage. Anomalous behaviour was identified by built-in alarms around 12 pm on April 21, 2022 on the originating cell.

5.4 Conclusion

Modern wireless communication network infrastructure is being deployed at an astonishing pace. To monitor this new infrastructure, it is critical to leverage automation and machine learning-based approaches. Furthermore, unsupervised approaches further simplify the deployment of these advanced tools, as they do not rely on the participation of network operator experts.

One of the more rudimentary aspects of LTE and 5G networks is the built-in alarm systems. To enhance anomaly detection capabilities, an unsupervised AD model RAINFOREST has been proposed. RAINFOREST has been shown to outperform baseline AD models in 98.57% of the cells tested in terms of EM and MV. This result indicates that RAINFOREST is achieving the best unsupervised AD performance [74]. The density-based clustering and temporal context added by using the forecast error allow RAINFOREST to dramatically improve performance over the baseline models.

Finally, RAINFOREST was tested on a known cell anomaly in a commercial fixed wireless LTE network. The anomalous behavior was first identified by RAINFOREST nearly 24 hours before any built-in LTE alarms tripped. Identifying major outages is not difficult for threshold-based alarm systems; however, the ability for RAINFOREST to rapidly identify a non-hard outage will enable network operators to provide better quality experience to end users by reducing the amount of time that anomalies impact the network. Although false alarms remain a possibility when using RAINFOREST, the ability to rank by severity and temporally group anomalies will reduce the overall number of alarms. Network operators generally receive an unmanageable number of alarms, which may become more manageable when using RAINFOREST.

It was demonstrated that RAINFOREST can be used to track an anomaly in a single cell but can also see the effects of the problem in neighboring cells. This is promising in that it could be used to track anomalous behavior throughout a region. Although cell-by-cell analysis was demonstrated, expanding the capabilities of RAINFOREST to manage network data in a spatiotemporal graph could achieve better results [104].

Chapter 6

Conclusion

Internet access is critical for both work and leisure in the digital era. Anecdotally, this work has been done entirely through telecommuting. People spend a lot of time browsing the web, shopping, meeting, and streaming music or other media content. It is no surprise that governments are supporting the push for quality internet access because the economic benefits are staggering in both developed and developing nations [105].

Canada is no exception to the government-backed push for access to quality broadband internet. The Broadband Fund has been established to help accelerate the deployment of broadband service in rural and remote areas [19]. Providing access to capital to develop network infrastructure is important; however, it is also important to understand how it is being deployed. For most remote rural areas, it is not possible to deploy fiber optic or digital subscriber lines. The main options for internet access are satellite, LTE, or 5G. In less remote areas, it is not uncommon to have access to mobile or fixed wireless LTE and 5G services, but some users are forced to rely entirely on satellite internet if they are outside of cellular coverage.

This work has focused on the fixed wireless aspect of rural broadband internet initiatives. In an ideal world, the design of cellular networks can be considered a *solved* problem, such that with a set of frequency allocations and a coverage area, an optimal network design can be produced. Unfortunately, the real world is messy, spectrum allocations are not always continuous, interference can decimate network performance, and equipment will ultimately fail.

In a modern network, there are many components required to take a request from a user to the internet and to provide them with content. Furthermore, the amount of traffic generated per user is growing exponentially [4]. This growth requires that an incredibly complex system be scaled to handle these increases in traffic. From the perspective of a network operator, there are a few choices; however, the focus will be placed on the cellular level. The best way to add network capacity is to increase the number of cells and the amount of spectrum available.

Unfortunately, there is a finite amount of spectrum available, but adding more cells is generally feasible. Scaling the number of cells increases the capacity of the network at the cost of monitoring efficiency. In this context, monitoring efficiency is how simple it is to perform tasks such as AD, short-term forecasting, long-term forecasting, and other network planning tasks. Automation and machine learning are the keystone to enable a rapidly growing number of subscribers and their required infrastructure. This work considers three tasks performed by network operators: long-term forecasting and network planning, short-term forecasting, and AD.

6.1 Long-term Forecasting

Network operators must understand how their network is going to evolve over time. Is a given cell congested? If not, when could it become congested? Is current congestion a seasonal transient or is it the result of long-term growth? Long-term forecasting can help answer these questions; however, environmental factors can come into play. This problem was investigated in Chapter 3.

The network KPIs were observed to show strong temporal changes throughout the year. Congestion and other metrics associated with traffic levels indicated that the winter months were worse than the summer months. This led to the desire to investigate the cause of the observed seasonality.

Statistically significant correlations were found between the KPIs and snow depth, snow fall, and temperature. The magnitudes indicated a moderate relationship for some KPI-environment feature pairs. Although the correlations appeared to indicate that the inclusion of environmental features was worthwhile, the relative MI between the feature pairs was found to be negligible. Relative MI was generally much lower than 5% of the LTE

KPIs, indicating that the amount of information added by the environmental features is negligible. Despite the relationships appearing minimal; they do not fully capture the temporal relationships, as PC and MI were considered for times $t_{KPI} = t_{environment}$. There is the possibility for forecasting models to extract temporal relationships that could be missed by PC and MI.

To further test how much information could be extracted from the feature set, a variety of forecasting models were selected. Several new and old architectures were considered; namely, DLinear, GRU, N-HiTS, and vanilla Transformers. For each of the models, the input and output sequence lengths were set to 28, 56, 84, 112, and 140 days. Furthermore, in this work, a unique approach was taken that segmented cells into subpopulations based on their relationship with environment features.

Using the relationship between a cell and its environmental features, the cells were segmented into five distinct sets: Combined, Correlated, Insignificant, Negative, and Positive. Combined considered all cells, correlated considered only cells with statistically significant correlation with environment features, insignificant considered only cells with statistically insignificant correlation with environment features, Negative was cells with a significant negative correlation, and Positive was cells with a significant positive correlation. Segmenting cells according to their correlation relationship was identified to improve the precision of the forecast for subpopulations.

Inclusion of environmental features has been shown to greatly improve forecast accuracy in urban mobile networks [7]. This work identifies that the same is generally not true for long-term forecasting in rural fixed wireless networks. For longer time horizons, that is, 84 days or longer, the accuracy of the forecast can be improved by a small percentage when cells are known to have statistically significant relationships with their environment. If this condition is not met, there is little to no improvement in the accuracy of the forecast. Although a small improvement in forecast accuracy was found in longer time horizons, it is significantly less beneficial than the dramatic reductions found in urban networks.

6.2 Short-term Forecasting

Long-term forecasts enable network operators to better support planning and design; however, many additional network management features can be enabled by short-term forecasts. Examples of these advanced services are dynamic resource allocation and anomaly detection. To provide a good trade-off between the number of KPIs and the information contained in the forecast output, a total of five KPIs were selected: DL-PRB, DL-THRP, UL-PRB, UL-THRP, and AUE.

Before approaching a forecasting problem, it is critical to understand the temporal properties of the signals being handled. There is a significant lack of rigor in existing studies in terms of characterizing the properties of the signals of interest. Most of the KPIs in all cells were identified as stationary in some capacity, be it completely stationary, trend stationary, or difference stationary.

The PACF identified that the temporal dependencies of the KPIs were very short. Beyond the first few time steps, the magnitudes of the PACF were negligible. There were some statistically significant lags near 96 lags, which corresponds to a day; however, the magnitudes were small despite their statistical significance. Analysis of the ACF indicated that the patterns in the signal are repeating and that there is a significant relationship between a signal and daily lagged copies. Despite this relationship, it is important to be aware of the way in which the ACF is described, since the PACF will better describe the realizable temporal relationship of the signal for forecasting applications.

Seven unique neural network architectures were selected: DLinear, NLinear, GRU, N-HiTS, Seq2Seq, Seq2Seq+Attention, and vanilla Transformer. A significant gap in short-term LTE KPI forecasting has been the selection of model architectures. Significant strides in terms of accuracy and computational cost have been made in recent years [1, 2]; however, many studies still consider more rudimentary architectures such as GRU and LSTM [7, 12, 93].

In addition to a focus on rudimentary architectures, the existing literature has not considered the effects of input sequence length on accuracy for single-step multivariate forecasts. Increasing the length of the input sequence from 4 to 96 steps was identified to improve MSE and MAE by 2.4 and 0.59%, respectively. Although an improvement was identified, increasing the data required to make a prediction by a factor of 24 is probably not worth the increase in computation time. The use case of the generated forecast will ultimately set the accuracy requirement; however, an input sequence length of 4 will provide a good trade-off between accuracy and computational cost.

In Chapter 3 it was shown that longer time horizons could benefit from the inclusion of environmental covariates under certain conditions. Snow fall and snow depth were foregone in favor of testing only the inclusion of temperature as an environmental covariate. The input sequence length variations described earlier in this section were repeated; however, the inclusion of environmental features was also considered. In general, it was found that including environmental features decreased or did not change the accuracy of the generated forecasts. This agrees with the findings of a similar study; however, it extended it by considering several different time horizons [37].

6.3 Anomaly Detection

Detecting anomalous events in thousands or millions of signals in an efficient and explainable manner is an ongoing research topic in many domains, of which LTE and 5G networks are no exception. The unknown nature and near-infinite combinations of anomalous patterns make time-series anomaly detection a difficult problem to approach. Labeling data for supervised learning approaches is not practical due to cost and the high likelihood of not identifying all types of anomaly. Semi-supervised methods have been proposed; however, flawed methodologies and poor baseline assumptions in existing studies have not produced robust AD algorithms [15, 14].

This work proposed an unsupervised hybrid AD method, called RAINFOREST, which combines statistical, clustering, and soft computing methods to mitigate the problems caused by each of the methods comprised. Instead of directly considering the LTE KPIs, the presented approach leverages a forecast and instead considers the forecast error. Under the assumption that the forecast is accurate and contains all predictable information, the error is expected to be noise. This is noteworthy because the error represents the unpredictable information in the signal. Raw KPIs were found to have parametric distribution properties, while associated forecast errors were approximately Gaussian.

Clustering analysis was performed on the multivariate forecast error using DBSCAN. This is a notable difference from existing studies that generally consider k-means clustering [15, 14]. K-means is a problematic clustering algorithm in that it is unable to reject samples as noise and can result in arbitrary clusters which do not represent the sample space in a meaningful way. For unsupervised AD methods, it is critical that the sample space is analyzed in a predictable and meaningful way. DBSCAN is used to identify which samples are anomalous. If a sample is in a low density region, DBSCAN will label it as noise. The set of points labeled as noise were flagged and selected for further statistical analysis.

The statistical analysis component of RAINFOREST considers the

Euclidean distance of the z-score for a given sample. A threshold was created to determine the intensity of the fault: this is known as the hard-soft threshold. If a sample was far from the cluster but below the threshold, it could be classified as soft, while it could be a hard fault if it was greater than the threshold. An important feature of RAINFOREST is that there is a temporal duration requirement to the anomalous sample classification: if a fault only lasts for one reporting period, then it will be flagged as noise. If a fault were to last for more than one reporting period, then it could be either a hard or a soft fault. This temporal filtering allows anomalous behavior that is transient in nature to be flagged, but also classified separately from hard and soft faults.

Characterizing the performance of an unsupervised algorithm is difficult. A popular and robust way to analyze the performance of unsupervised AD algorithms is EM and MV analysis. These are unsupervised parallels to receiver-operating and precision-recall curves [74]. In simple terms, they are a measure of how well the AD model is able to identify high- and low-density regions in the sample space. Using EM and MV analysis, RAINFOREST was compared to three baseline models: IF, OCSVM, and LOF. Since the temporal aspect added by the forecast is critical to operation, the baseline models were provided the forecasting error too. RAINFOREST outperformed the baselines by a significant margin. It was the best in terms of EM, MV, and combined EM and MV in 98.57%, 99.97%, and 98.57% respectively. The agreement between EM and MV indicates that it is the best performing model [74].

In addition to benchmarking RAINFOREST against baseline models, a case study was presented on using RAINFOREST on a known network issue. RAINFOREST was able to detect network faults nearly 24 hours before they occurred. It should also be noted that RAINFOREST could be used to track the fault between neighboring cells. This strongly supports that RAINFOREST could be used to find, track, and monitor network faults, which could help accelerate root cause analysis for network operators.

6.4 Final Remarks and Future Work

In recent years it has become clear that access to the internet is a requirement for both work and play. This requirement has driven not only the requirement for users to have high-speed services but also reliable service. Providing such services to rural areas is difficult due to the design and cost constraints of doing so. It is generally best to provide fiber optic connections directly to homes and businesses, which then distribute access through local wireless and cabled connections. Installation of fiber optic cables or DSL is generally not economically viable to install in rural areas, forcing network operators to focus on wireless access technologies.

Satellite and fixed wireless services have shown great promise in providing internet in rural areas. Low-earth orbit satellites have been gaining traction in recent years; however, fixed wireless LTE and 5G are generally the best contenders for rural broadband internet access. Fixed wireless services are not implemented without some difficulties. Terrain and other propagation factors can be problematic for network operators. Furthermore, seasonal environmental variations can alter the way users behave.

The cellular concept is one of the key features that helps enable rural broadband internet service. In a way, satellites provide spatial segmentation of service using unique satellite beams. LTE and 5G cellular services use unique cells and spectrum resources to provide service segmentation. Network traffic capacity can generally be increased in one of two ways: by creating more smaller cells to increase frequency reuse or by increasing the available spectrum. The first option is generally more common, since the spectrum is expensive and finite; however, there is a practical limit, since deploying and maintaining cells is not free.

The deployment of more cells to increase network capacity is causing the amount of network infrastructure that network operators need to expand dramatically. Engineering teams are generally unable to scale as fast as a network. Furthermore, it is not cost-effective to simply hire more engineers. Automation and machine learning can enable engineering teams to operate more effectively. Chapter 3 proposed the use of long-term forecasting for tasks such as congestion relief and upgrade planning, while short-term forecasting and AD were proposed in Chapters 4 and 5, respectively.

Network operators should be aware of the many different time horizons that can affect their network operation. There are many logical continuations of this work which would contribute to network management by operators and by expanding the existing literature. A fundamental aspect of this work is the computational infrastructure that supports it. Without databases, distributed computing frameworks, and GPUs and other hardware accelerators, this work would not be possible. Hardware and software architecture is beyond the scope of this work; however, it is important to be aware that there technological landscape is rapidly changing. This work considers the implications of machine learning model computational complexity, which is only one concern of real-time analysis. Processing paradigms such as batch or stream processing will alter the data pipeline that feeds into the forecasting and AD models described in this work. End-to-end performance optimization and machine learning model life-cycle management are no easy task, but ultimately they are a practical engineering problem rather than a research problem.

6.4.1 Long-term Forecasting and Analysis

The investigation of seasonal changes and user behavior, which was discussed in Chapter 3, could be expanded to provide better granularity. The study focuses on cell-level performance and seasonal variation. Several different ways to segment the cell population were proposed. Segmentation of cell populations that were considered improved performance; however, drilling down to the user level and using motif-based analysis [100] could yield interesting results. Analyzing the long-term seasonality of users or cells that behave temporally similar is expected to provide better forecast accuracy.

Cell-level analysis indicated that performance changes caused by seasonal changes in the environment were relatively consistent throughout the country. This was not surprising since the relative MI indicated a small relationship between environmental factors and network performance; however, different regions of Canada may behave differently relative to each other, which could lead to different changes in seasonal user behavior. For example, the population of Alberta may behave differently than that of New Brunswick or another province or territory. At a macroscopic level the KPIs are similar in their relationship to the environment, but from a network planning and congestion management perspective there may be some regional deviation. The continuation of this portion of work quickly begins to enter the anthropological and psychological study of populations and how they use the internet. The sociological study of populations indicates that there are seasonal changes and differences in behavior between populations [80, 6]. These studies and other behavior analysis could help drive improved long-term forecasts and analysis.

Changes in forecast accuracy were minimal when using different forecast models and covariates. Architectural innovation may help improve accuracy; however, the newer architectures, such as TiDE [106] are offering diminishing returns with each new incremental advance. Accuracy improvements have largely stalled in favor of more efficient forecast generation. It is anticipated that the largest performance improvements will come from finding optimal covariate signals. The considered set of environmental features yielded minimal benefit; however, there may exist other sets of features that can augment performance.

6.4.2 Short-term forecasting

Long-term forecasting is poised to benefit from the inclusion of non-environmental covariates. For short-term forecasting it is not anticipated that covariates will greatly improve accuracy. Even from a forecast accuracy perspective the forecast accuracy is not anticipated to improve dramatically at a cell level. The KPIs are incredibly noisy meaning that it is unlikely that more information could be extracted from the signal to benefit forecasting accuracy. This is supported by the results in Chapter 4 where most architectures obtain very similar results regardless of configuration.

Since improvement in accuracy is unlikely, computational efficiency is anticipated to become a key area of development for LTE and 5G KPI forecasting. From RNN-based models to NLinear and DLinear, there has been a significant downwards trend in computational complexity while still achieving excellent forecast accuracy. For downstream analysis of the forecasts it is important to generate them as quickly as possible. Faster forecast generation will enable more complex analysis and actions to be taken without missing computation deadlines. AD and dynamic resource allocation will greatly benefit from faster forecasting architectures.

Preliminary studies on graph-based forecasting models have shown promising results in mobile wireless networks [104]. Although it has been shown that rural fixed wireless networks behave differently, there is the potential for graph-based approaches to improve performance. Configuring the entire cellular network as a single graph is not anticipated to perform well in terms of computational efficiency. Instead, it is likely that segmenting the network into spatially isolated clusters will achieve best results. This may also be guided by environmental characteristics of the cells being grouped together. The first logical configuration to be tested would be combining a single site into a graph based on its comprising cells. Since the network has immobile users, features like handovers will not occur; however, users may behave similarly in a geographical area which could be better learned by a graph.

6.4.3 Anomaly Detection

One of the greatest practical aspects of the work presented is the AD capabilities of RAINFOREST. Across many application AD remains incredibly difficult. Anomalous behaviour is generally unpredictable and infrequent, making for a difficult problem to address. Unsupervised approaches remain popular since they remove the requirement to have labeled data. Although it is not a requirement for LTE and 5G AD, unsupervised AD approaches are highly favored.

RAINFOREST performs exceptionally well at detecting anomalous behaviour at a cell level. Expanding the model's capabilities to consider a cluster of spatially co-located cells may improve detection capabilities. Furthermore, if the forecast generating model were graph-based, it would be aware of the relationship between cells. This relationship could allow for better characterization of the temporal dynamics of a cell. For example, if two cells shared a coverage footprint for capacity management purposes, users could be seen to hop back and forth between cells. This might be flagged as an anomaly; however, a graph-based forecasting model and graph-based implementation of RAINFOREST would be capable of understanding the temporal context.

Further experimentation with the comprising components of RAINFOREST could lead to further performance improvements as well. Several different forecasting algorithms were tested and the results were varied. If a more accurate forecasting architecture or trained model could be used, that would tighten the error-margin on the forecast which would improve the quality of the input data into RAINFOREST. In addition to the forecasting models, it is likely that the clustering algorithm, DBSCAN, could either be further optimized or exchanged for a simpler clustering technique. In general, clustering methods are computationally inefficient so there is the possibility for improvement in performance and efficiency. Other clustering algorithms may be developed; however, it is important that RAINFOREST retains the use of density-based clustering. This will not only provide clustering results that are explainable, but also allow for the spatially similar samples to be clustered together. This will allow for the best theoretical performance in terms of EM and MV based on their definitions.

6.4.4 Closing Remarks

Automation and machine learning are critical for enabling advanced monitoring of LTE and 5G networks. It is not expected that telecommunication experts will be replaced as a result of automation; however, it will change the way they work. The large-scale analysis of a wireless communication network is not a simple feat. Creating tools to allow experts to spend less time massaging data into a usable form and building analysis around it is an area that is ready for improvement. This work has aimed to build a foundation for automation and machine learning for LTE and 5G networks that are deployed specifically in rural areas. Forecasting across multiple time scales and horizons has been demonstrated and can be used for varying tasks such as network capacity planning and dynamic resource allocation. An improved approach to AD was also demonstrated that allows anomalous network behavior to be detected sooner and more accurately than built-in monitoring systems.

The ways in which automation and machine learning will change the ways telecommunication network operators function are nearly infinite. The question ultimately remains, *where can it best be applied to provide the best service to network users*?

Bibliography

[1] C. Challu, K. G. Olivares, B. N. Oreshkin, F. G. Ramirez, M. M. Canseco, and A. Dubrawski, “Nhits: Neural hierarchical interpolation for time series forecasting,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, pp. 6989–6997, 2023.

[2] A. Zeng, M. Chen, L. Zhang, and Q. Xu, “Are Transformers Effective for Time Series Forecasting?,” Aug. 2022. arXiv:2205.13504 [cs].

[3] ITU, “Electrical characteristics of the surface of the Earth P Series Radiowave propagation,” tech. rep., International Telecommunication Union, 2017.

[4] Ericsson, “Ericsson Mobility Report November 2022,” tech. rep., Ericsson Inc., 2022.

[5] C. Zhang, P. Patras, and H. Haddadi, “Deep Learning in Mobile and Wireless Networking: A Survey,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[6] C. J. Matz, D. M. Stieb, and O. Brion, "Urban-rural differences in daily time-activity patterns, occupational activity and housing characteristics," *Environmental Health: A Global Access Science Source*, vol. 14, no. 1, pp. 1–11, 2015.

[7] T. Kuber, I. Seskar, and N. Mandayam, "Traffic prediction by augmenting cellular data with non-cellular attributes," *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2021-March, pp. 1–6, 2021.

[8] K. L. Chee, S. A. Torrico, and T. Kurner, "Foliage attenuation over mixed terrains in rural areas for broadband wireless access at 3.5 GHz," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 7, pp. 2698–2706, 2011.

[9] K. L. Chee, A. Anggraini, and T. Kurner, "Effects of carrier frequency, antenna height and season on broadband wireless access in rural areas," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 7, pp. 3432–3443, 2012.

[10] K. L. Chee, S. A. Torrico, and T. Kurner, "Radiowave propagation prediction in vegetated residential environments," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 486–499, 2013.

[11] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, "Big Data Driven Mobile Traffic Understanding and Forecasting: A Time Series

Approach," *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 796–805, 2016.

[12] A. Bhorkar, K. Zhang, and J. Wang, "DeepAuto: A Hierarchical Deep Learning Framework for Real-Time Prediction in Cellular Networks," *arXiv*, 2019. arXiv: 2001.01553.

[13] F. Chernogorov, S. Chernov, K. Brigatti, and T. Ristaniemi, "Sequence-based detection of sleeping cell failures in mobile networks," *Wireless Networks*, vol. 22, no. 6, pp. 2029–2048, 2016.

[14] M. Nour, M. Awad, M. Kamel, M. Essa, and N. Abdelbaki, "Anomaly Detection using Unsupervised Learning in LTE Mobile Network," in *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 195–199, Dec. 2021.

[15] A. Pelati, M. Meo, and P. Dini, "Traffic Anomaly Detection Using Deep Semi-Supervised Learning at the Mobile Edge," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 8919–8932, Aug. 2022.

[16] M. Stonebraker and L. A. Rowe, "The design of Postgres," *ACM Sigmod Record*, vol. 15, no. 2, pp. 340–355, 1986.

[17] M. Rocklin, "Dask: Parallel computation with blocked algorithms and task scheduling," in *Proceedings of the 14th Python in Science Conference*, vol. 130, p. 136, SciPy Austin, TX, 2015.

[18] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, I. Stoica, and others, "Spark: Cluster computing with working sets," *HotCloud*, vol. 10, no. 10-10, p. 95, 2010.

[19] Canadian Radio-television and Telecommunication Commission, "Broadband Fund: About the Fund | CRTC." https://crtc.gc.ca/eng/internet/fnds.htm, 2019.

[20] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[21] Industry Canada, "Auction of Spectrum Licences for Advanced Wireless Services and Other Spectrum in the 2 GHz Range — Licence Winners," 2023.

[22] C. D. Rouse, B. R. Petersen, and B. G. Colpitts, "Characterising an In-Room MIMO System Employing Elevation-Directional Access Point Antennas," *Wireless Personal Communications*, vol. 96, pp. 3889–3905, Oct. 2017.

[23] A. G. B. Colpitts, *Rural MIMO Fixed-Wireless Channel Capacity Optimization*. MSc Thesis, University of New Brunswick., 2020.

[24] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO Has Unlimited Capacity," *IEEE Transactions on Wireless Communications*, vol. 17, pp. 574–590, Jan. 2018.

[25] D. M. Pozar, *Microwave Engineering.* John wiley & sons, 2011.

[26] T. S. Rappaport, *Wireless Communications: Principles and Practice.* Upper Saddle River, N.J. SE - xxiii, 707 pages : illustrations ; 25 cm.: Prentice Hall PTR, 2002.

[27] Health Canada, "Limits of human exposure to radiofrequency electromagnetic fields in the frequency range from 3 kHz to 300 GHz," *Safety Code 6*, vol. 6, pp. 1–18, 2015.

[28] F. T. Ulaby and U. Ravaioli, *Fundamentals of Applied Electromagnetics*, vol. 7. Pearson Upper Saddle River, NJ, 2015.

[29] C. A. Balanis, *Advanced Engineering Electromagnetics.* John Wiley & Sons, 2012.

[30] D. Gesbert, H. Bölcskei, D. A. Gore, and A. J. Paulraj, "Outdoor MIMO wireless channels: Models and performance prediction," *IEEE Transactions on Communications*, vol. 50, pp. 1926–1934, Dec. 2002.

[31] T. T. K. Tran, T. Lee, and J. S. Kim, "Increasing neurons or deepening layers in forecasting maximum temperature time series?," *Atmosphere*, vol. 11, no. 10, 2020.

[32] D. M. Durairaj and B. H. Mohan, "A convolutional neural network based approach to financial time series prediction," *Neural Computing and Applications*, vol. 2, 2022.

[33] M. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model," *2nd Novel Intelligent and Leading Emerging Sciences Conference, NILES 2020*, pp. 87–92, 2020.

[34] G. Marcjasz, B. Uniejewski, and R. Weron, "Beating the naïve-combining lasso with naïve intraday electricity price forecasts," *Energies*, vol. 13, no. 7, pp. 1–16, 2020.

[35] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020. arXiv: 1704.04110 Publisher: Elsevier B.V.

[36] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–16, 2018. arXiv: 1707.01926.

[37] A. G. B. Colpitts and B. R. Petersen, "Short-Term Multivariate KPI Forecasting in Rural Fixed Wireless LTE Networks," *IEEE Networking Letters*, vol. 5, pp. 11–15, Mar. 2023.

[38] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Australia: OTexts, 3rd ed., 2021.

[39] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *Journal of Economic and Social Measurement*, vol. 29, no. 1-3, pp. 123–125, 2004.

[40] B. Lim, S. Arık, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021. arXiv: 1912.09363.

[41] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "STL: A seasonal-trend decomposition," *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[42] K. Bandara, R. J. Hyndman, and C. Bergmeir, "MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns," July 2021. arXiv:2107.13462 [stat].

[43] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," Feb. 2020. arXiv:1905.10437 [cs, stat].

[44] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," Mar. 2021. arXiv:2012.07436 [cs].

[45] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting," June 2022. arXiv:2201.12740 [cs, stat].

[46] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

[47] J. L. Rodgers and W. A. Nicewander, "Thirteen Ways to Look at the Correlation Coefficient," *The American Statistician*, vol. 42, pp. 59–66, Feb. 1988.

[48] R. Aggarwal and P. Ranganathan, "Common pitfalls in statistical analysis: The use of correlation techniques," *Perspect Clin Res*, vol. 7, no. 4, pp. 187–190, 2016.

[49] R. M. Gray, *Entropy and Information Theory.* Springer Science & Business Media, 2011.

[50] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 69, no. 6, p. 16, 2004.

[51] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting.* Springer texts in statistics, New York: Springer, 2nd ed ed., 2002.

[52] Z. Ouyang, P. Ravier, and M. Jabloun, "STL Decomposition of Time Series Can Benefit Forecasting Done by Statistical Methods but Not

by Machine Learning Ones," in *The 7th International conference on Time Series and Forecasting*, p. 42, MDPI, July 2021.

[53] M. Marcellino, J. H. Stock, and M. W. Watson, "A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series," *Journal of Econometrics*, vol. 135, pp. 499–526, Nov. 2006.

[54] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, Aug. 2016. arXiv:1603.02754 [cs].

[55] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[56] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[57] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[58] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner,

L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Dec. 2019. arXiv:1912.01703 [cs, stat].

[59] B. Lim and S. Zohren, "Time Series Forecasting With Deep Learning: A Survey," *Phil. Trans. R. Soc. A.*, vol. 379, p. 20200209, Apr. 2021. arXiv:2004.13408 [cs, stat].

[60] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2194, 2021. arXiv: 2004.13408.

[61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[62] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[63] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition," Feb. 2014. arXiv:1402.1128 [cs, stat].

[64] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," Oct. 2014. arXiv:1409.1259 [cs, stat].

[65] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, vol. 4, pp. 3104–3112, 2014. arXiv: 1409.3215.

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017. arXiv: 1706.03762.

[67] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The Efficient Transformer," Feb. 2020. arXiv:2001.04451 [cs, stat].

[68] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting," Jan. 2020. arXiv:1907.00235 [cs, stat].

[69] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," Jan. 2022. arXiv:2106.13008 [cs].

[70] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Comput. Surv.*, vol. 54, pp. 1–38, Mar. 2022. arXiv:2007.02500 [cs, stat].

[71] I. A. Karatepe and E. Zeydan, "Anomaly detection in cellular network data using big data analytics," in *European Wireless 2014; 20th European Wireless Conference*, pp. 1–5, VDE, 2014.

[72] B. Hussain, Q. Du, and P. Ren, "Semi-supervised learning based big data-driven anomaly detection in mobile wireless networks," *China Communications*, vol. 15, no. 4, pp. 41–57, 2018.

[73] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: a comprehensive evaluation," *Proc. VLDB Endow.*, vol. 15, pp. 1779–1797, May 2022.

[74] N. Goix, "How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms?," July 2016. arXiv:1607.01152 [cs, stat].

[75] Enthought, "Detecting Anomalies Using Statistical Distances | SciPy 2018 | Charles Masson." https://www.youtube.com/watch?v=U7xdiGc7IRU, July 2018.

[76] M. Kubat, *An Introduction to Machine Learning.* Cham: Springer International Publishing, 2017.

[77] M. Daszykowski and B. Walczak, "Density-Based Clustering Methods," *Comprehensive Chemometrics*, vol. 2, pp. 635–654, 2009.

[78] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Van-

derplas, A. Passos, and D. Cournapeau, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[79] Z. Z. Darban, G. I. Webb, S. Pan, C. C. Aggarwal, and M. Salehi, “Deep Learning for Time Series Anomaly Detection: A Survey,” Dec. 2022. arXiv:2211.05244 [cs].

[80] J. A. Leech, W. C. Nelson, R. T. Burnett, S. Aaron, and M. E. Raizenne, “It’s about time: A comparison of Canadian and American time-activity patterns,” *Journal of Exposure Analysis and Environmental Epidemiology*, vol. 12, no. 6, pp. 427–432, 2002.

[81] A. L. Wagner, F. Keusch, T. Yan, and P. J. Clarke, “The impact of weather on summer and winter exercise behaviors,” *Journal of Sport and Health Science*, vol. 8, no. 1, pp. 39–45, 2019.

[82] F. Di Paolo, B. Cosciotti, S. E. Lauro, E. Mattei, and E. Pettinelli, “Dry snow permittivity evaluation from density: A critical review,” *2018 17th International Conference on Ground Penetrating Radar, GPR 2018*, pp. 1–5, 2018.

[83] S. Evans, “Dielectric Properties of Ice and Snow–a Review,” *Journal of Glaciology*, vol. 5, no. 42, pp. 773–792, 1965.

[84] C. Mätzler, “Microwave permittivity of dry snow,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 2, pp. 573–581, 1996.

[85] A. Mavrovic, J. B. Madore, A. Langlois, A. Royer, and A. Roy, "Snow liquid water content measurement using an open-ended coaxial probe (OECP)," *Cold Regions Science and Technology*, vol. 171, no. November 2019, p. 102958, 2020.

[86] S. Sugiyama, H. Enomoto, S. Fujita, K. Fukui, F. Nakazawa, and P. Holmlund, "Dielectric permittivity of snow measured along the route traversed in the Japanese-Swedish Antarctic expedition 2007/08," *Annals of Glaciology*, vol. 51, no. 55, pp. 9–15, 2010.

[87] B. C. Brown and B. R. Petersen, "Estimating permittivity of snow in a multi-layer model using multi-ray simulation and a genetic algorithm," *ACM International Conference Proceeding Series*, pp. 57–64, 2018.

[88] R. L. Jesch, "Dielectric measurements of five different soil textural types as functions of frequency and moisture content," tech. rep., Electromagnetic Fields Division, Center for Electronics and Electrical Engineering, National Bureau of Standards, 1978.

[89] A. B. Itolikar, A. Joshi, S. Deshpande, and M. Kurtadikar, "Dielectric Response Due to Combine Effect of Soil and Vegetation Layer (Grass) at C-Band Microwave Frequency," *2020 IEEE India Geoscience and Remote Sensing Symposium, InGARSS 2020 - Proceedings*, pp. 9–12, 2020.

[90] Natural Resources Canada, "Statistical data | Forests | Natural Resources Canada." https://cfs.nrcan.gc.ca/statsprofile/overview/NB, Accessed Dec 22, 2023.

[91] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An Experimental Review on Deep Learning Architectures for Time Series Forecasting," *International Journal of Neural Systems*, vol. 31, no. 3, pp. 1–25, 2021. arXiv: 2103.12057v2.

[92] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep Double Descent: Where Bigger Models and More Data Hurt," Dec. 2019. arXiv:1912.02292 [cs, stat].

[93] H. Na, Y. Shin, D. Lee, and J. Lee, "LSTM-based throughput prediction for LTE networks," *ICT Express*, no. xxxx, pp. 1–6, 2021.

[94] S. T. Nabi, M. R. Islam, M. G. R. Alam, M. M. Hassan, S. A. Alqahtani, G. Aloi, and G. Fortino, "Deep Learning based Fusion Model for Multivariate LTE Traffic Forecasting and Optimized Radio Parameter Estimation," *IEEE Access*, pp. 1–1, 2023.

[95] L. Sehovac, C. Nesen, and K. Grolinger, "Forecasting building energy consumption with deep learning: A sequence to sequence approach," *Proceedings - 2019 IEEE International Congress on Internet of Things, ICIOT 2019 - Part of the 2019 IEEE World Congress on Services*, pp. 108–116, 2019.

[96] H. D. Trinh, L. Giupponi, and P. Dini, "Urban Anomaly Detection by processing Mobile Traffic Traces with LSTM Neural Networks," *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks workshops*, vol. 2019-June, 2019.

[97] G. F. Ciocarlie, U. Lindqvist, S. Nováczki, and H. Sanneck, "Detecting anomalies in cellular networks using an ensemble method," in *Proceedings of the 9th international conference on network and service management (CNSM 2013)*, pp. 171–174, IEEE, 2013.

[98] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[99] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," *ACM sigmod record*, vol. 25, no. 2, pp. 103–114, 1996.

[100] K. Kamgar, S. Gharghabi, and E. Keogh, "Matrix Profile XV: Exploiting Time Series Consensus Motifs to Find Structure in Time Series Sets," in *2019 IEEE International Conference on Data Mining (ICDM)*, (Beijing, China), pp. 1156–1161, IEEE, Nov. 2019.

[101] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proceedings of the 2000 ACM*

SIGMOD international conference on Management of data, pp. 93–104, 2000.

[102] A. Thomas, V. Feuillard, and A. Gramfort, "Calibration of One-Class SVM for MV set estimation," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–9, Oct. 2015.

[103] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-Based Anomaly Detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, pp. 1–39, Mar. 2012.

[104] Y. Fang, S. Ergut, and P. Patras, "SDGNet: A Handover-Aware Spatiotemporal Graph Neural Network for Mobile Traffic Forecasting," *IEEE Communications Letters*, vol. 26, no. 3, pp. 582–586, 2022.

[105] I. Bertschek, W. Briglauer, K. HHschelrath, and T. Niebel, "The Economic Impacts of Telecommunications Networks and Broadband Internet: A Survey," *SSRN Journal*, 2016.

[106] A. Das, W. Kong, A. Leach, R. Sen, and R. Yu, "Long-term Forecasting with TiDE: Time-series Dense Encoder," Apr. 2023. arXiv:2304.08424 [cs, stat].

Vita

Candidate's full name: Alexander Gordon Bruce Colpitts

University attended:

University of New Brunswick, B.Sc.E. (EE), 2018

University of New Brunswick, M.Sc.E. (EE), 2020

Conference Publications:

Marco A. Merlini, Alexander G. B. Colpitts, Satinder Gill, Bruce G. Colpitts, and Brent R. Petersen, "Channel Modelling and Efficient Time-Domain Simulation of a Harmonic Transponder Tag," in *Proceedings of the 6th International Conference on Telecommunications and Remote Sensing*, ICTRS'17, (New York, NY, USA), pp. 43–47, Association for Computing Machinery, 2017.

Journal Publications:

Alexander G. B. Colpitts and Brent R. Petersen, "Short-Term Multivariate KPI Forecasting in Rural Fixed Wireless LTE Networks," *IEEE Net-*

working Letters, vol. 5, no. 1, pp. 11–15, Mar. 2023, doi: 10.1109/LNET.2023.3242680.

Alexander G. B. Colpitts and Brent R. Petersen, "Unsupervised Anomaly Detection for Rural Fixed Wireless LTE Networks," *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 46, no. 4, pp. 256-261, 2023, doi: 10.1109/ICJECE.2023.3275975.