

# ML Based Performance Assurance and VoC Management of Highly Convergence Mobile Operator Network.

by

Md. Arifur Rahman  
ID: 20166040

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
M.Sc. in Computer Science

Department of Computer Science and Engineering  
BRAC University  
October 2023

© 2023. BRAC University  
All rights reserved.

# Declaration

It is hereby declared that

1. We independently and creatively created the thesis that is being presented here while pursuing our academic careers at Brac University.
2. This thesis paper does not contain any previously published or written material from a third party, unless it has been properly recognized with appropriate citations.
3. Additionally, this thesis paper has not been submitted to any institution for consideration for any other academic degree or credential.
4. Additionally, we have given all key sources of support used in the research and writing process the due credit they deserve.

**Full Name & Signature of Student:**

A handwritten signature in black ink that reads "Aijaz Rahman". The signature is written in a cursive style and is slightly tilted to the right.

---

Md.Arifur Rahman

Student ID: 20166040

# Approval

The thesis/project titled “ML Based Performance Assurance and VoC Management of Highly Convergence Mobile Operator Network.” submitted by

1. Student Name: Md.Arifur Rahman

The work completed during the fall of 2023 has been deemed acceptable and meets the partial requirements for the award of the M.Sc. in Computer Science degree on October 18, 2023.

## Examining Committee:

Supervisor:

(Member)



---

Moin Mostakim  
Senior Lecturer  
Department of Computer Science and Engineering  
School of Data Science  
BRAC University  
Email:mostakim@bracu.ac.bd

Program Coordinator:

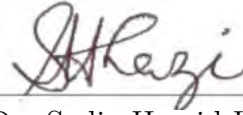
(Member)



---

Dr. Amitabha Chakrabarty  
Professor  
Department of Computer Science and Engineering  
School of Data Science  
Brac University  
Email:amitabha@bracu.ac.bd

Head of Department:  
(Chair)



---

Dr. Sadia Hamid Kazi  
Associate Professor  
Department of Computer Science and Engineering  
School of Data Science  
Brac University  
Email:skazi@bracu.ac.bd

## Examining Committee:

External Examiner:  
(Member)



---

Dr. Mohammad Shamsul Arefin  
Professor  
Department of Computer Science and Engineering  
Chittagong University of Engineering and Technology (CUET)  
sarefin@cuet.ac.bd

Internal Examiner:

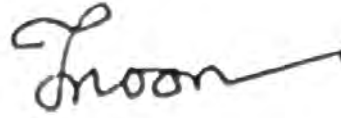


(Member)

---

Dr. Muhammad Iqbal Hossain  
Associate Professor  
Department of Computer Science and Engineering  
School of Data Science  
BRAC University  
Email:iqbal.hossain@bracu.ac.bd

Internal Examiner:



(Member)

---

Dr. Jannatun Noor  
Assistant Professor  
Department of Computer Science and Engineering  
School of Data Science  
BRAC University  
Email: [jannatun.noor@bracu.ac.bd](mailto:jannatun.noor@bracu.ac.bd)

# Abstract

The convergence of technology in mobile operator networks has ushered in a new era of interconnectedness and communication efficiency. As these networks become increasingly complex, ensuring their optimal performance and addressing customer concerns become paramount. This thesis delves into the realm of "ML Based Performance Assurance and VoC Management of Highly Convergence Mobile Operator Network," presenting a multifaceted approach to tackling these challenges through advanced machine learning (ML) techniques by obsolete traditional manual working approach of MNOs. Within the scope of ML-based performance assurance, this study places a strong emphasis on Forecasting Time Series and detecting Anomalies. Leveraging the power of predictive analytics, the research harnesses a spectrum of algorithms including ARIMA, XGBoost, LSTM, Dynamic Linear Model, Prophet, VAR, and GRU. This enables the anticipation of network behavior and facilitates proactive measures for optimization. Additionally, the study integrates sophisticated Anomaly Detection methods encompassing DBSCAN, Isolation Forest, Local Outlier Factor (LOF), One Class SVM, Elliptic Envelope, and Autoencoders. These techniques empower the system to identify and mitigate aberrations in real-time, safeguarding network statistics and ensure business growth. Extending the purview of the research, the study delves into Voice of Customer (VoC) Management within the context of highly converged mobile operator networks. By employing diverse algorithms such as SVM, CNN, GNB, MNB, and LR, the research addresses the critical task of understanding customer insights, preferences, thoughts and concerns. Through effective VoC analysis, operators can tailor their services to meet customer expectations, thereby enhancing overall satisfaction. This thesis contributes to the field by providing a all-encompassing structure for the enhancement and design of mobile operator networks. The amalgamation of ML-based performance assurance and VoC management techniques presents a holistic solution for network operators and service providers. By proactively forecasting network behavior and promptly addressing anomalies, operators can ensure seamless operations. Simultaneously, a holistic and customer centric approach driven by advanced ML algorithms enables the refinement of services based on customer feedback to obsolete traditional working approach of mobile network operators.

**Keywords:** Machine Learning; Time Series Forecasting; Anomaly Detection; VoC Management; Algorithms

## **Dedication (Optional)**

This thesis is a heartfelt dedication to my beloved parent, who served as my ultimate inspiration. Their dream is for me to pursue higher education. Without their boundless love and unwavering support, attaining my Master's degree would have been an insurmountable task.

## Acknowledgement

Firstly, my gratitude to the Almighty Allah, under whose blessings this thesis has been successfully completed without any major hindrance.

Next, I wish to acknowledge the invaluable contributions of my supervisor, Moin Mostakim sir, and the program coordinator, Dr. Amitabha Chakrabarty sir. Their unwavering support, guidance, and encouragement were crucial in shaping the trajectory of this work. Their scholarly insights, patience, and constructive criticism, along with their commitment to supervising and correcting drafts at every stage, have played a pivotal role in bringing this thesis to fruition.

Lastly, I am deeply thankful to our parents whose constant support has been pivotal in making this journey possible. Their unwavering kindness and prayers have brought us to the brink of graduation.



# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Ethics Statement</b>	<b>v</b>
<b>Abstract</b>	<b>v</b>
<b>Dedication</b>	<b>vi</b>
<b>Acknowledgment</b>	<b>vii</b>
<b>Table of Contents</b>	<b>viii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Statement . . . . .	2
1.3 Research System Goals . . . . .	3
1.4 Research Objectives . . . . .	4
1.5 Research Methodology . . . . .	5
1.6 Research Questions . . . . .	7
1.7 Research Scope . . . . .	8
1.8 Research Limitations . . . . .	8
1.9 Research Significance in telecommunication industry . . . . .	8
1.10 Thesis Organization . . . . .	9
<b>2 Literature Review</b>	<b>11</b>
2.1 Evolution of Mobile Operator Networks . . . . .	11
2.1.1 Early Mobile Networks . . . . .	12
2.1.2 Digital Revolution: 2G and 3G Networks . . . . .	12
2.1.3 Data-Centric Networks: 4G and 5G . . . . .	12
2.2 Related Studies and Research on Telecommunication Networks . . . . .	12
2.2.1 Concept of Network Convergence . . . . .	14
2.2.2 Challenges in Network Convergence . . . . .	14
2.3 Performance Assurance in Mobile Networks . . . . .	15
2.3.1 Key Performance Indicators (KPIs) . . . . .	15

2.3.2	Traditional Approaches to Performance Assurance . . . . .	15
2.3.3	Machine Learning for Performance Assurance . . . . .	16
2.4	Voice of the Customer (VoC) Management . . . . .	16
2.4.1	Importance of VoC in Network Management . . . . .	17
2.4.2	Challenges in VoC Management . . . . .	17
2.4.3	Machine Learning in VoC Management . . . . .	17
2.5	ML-Based Performance Assurance and VoC Management Integration	18
2.6	Performance Assurance and VoC Management in ML Techniques . . .	18
2.7	Research Gaps in the Literature . . . . .	20
<b>3</b>	<b>Methodology</b>	<b>21</b>
3.1	Research Design . . . . .	21
3.2	Research Approach . . . . .	21
3.3	Data Collection . . . . .	21
3.3.1	LTE Attach . . . . .	21
3.4	Data Set for Performance Assurance . . . . .	24
3.5	Data Set for VoC Management . . . . .	25
3.6	Time Series Algorithm Selection and Implementation . . . . .	26
3.7	Anomaly Detection Algorithm Selection and Implementation . . . . .	27
3.8	Voice of the Customer Management Algorithm Implementation . . . . .	29
3.9	Architecture of the ML-Driven Performance Assurance and VoC Management System . . . . .	30
3.10	Data Flow and Processing Stages . . . . .	31
3.11	Interaction between Performance Assurance and VoC Components . .	33
<b>4</b>	<b>ML Based Performance Assurance</b>	<b>35</b>
4.1	Time Series Forecasting Algorithms . . . . .	35
4.1.1	ARIMA Model (AutoRegressive Integrated Moving Average) .	35
4.1.2	XGBoost (eXtreme Gradient Boosting) . . . . .	37
4.1.3	LSTM (Long Short-Term Memory) . . . . .	39
4.1.4	Dynamic Linear Model (DLM) . . . . .	41
4.1.5	Prophet . . . . .	42
4.1.6	VAR (Vector AutoRegressive) Model . . . . .	43
4.1.7	GRU (Gated Recurrent Unit) . . . . .	45
4.2	Anomaly Detection Algorithms . . . . .	46
4.2.1	DBSCAN (The Density-Based Spatial Clustering of Applications with Noise) . . . . .	46
4.2.2	Isolation Forest . . . . .	48
4.2.3	Local Outlier Factor (LOF) . . . . .	49
4.2.4	One Class SVM (Support Vector Machine) . . . . .	50
4.2.5	Elliptic Envelope . . . . .	52
4.2.6	Autoencoders . . . . .	53
4.2.7	HBOS (Histogram-Based Outlier Score) . . . . .	55
<b>5</b>	<b>Voice of the Customer (VoC) Management</b>	<b>57</b>
5.1	Voice of the Customer (VoC) Management Algorithms . . . . .	57
5.1.1	Support Vector Machine (SVM) . . . . .	57
5.1.2	Convolutional Neural Network (CNN) . . . . .	59
5.1.3	Gaussian Naive Bayes (GNB) . . . . .	60

5.1.4	Multinomial Naive Bayes (MNB)	62
5.1.5	Logistic Regression (LR)	63
<b>6</b>	<b>Implementation and Results</b>	<b>66</b>
6.1	Time Series Forecasting Results	66
6.2	Anomaly Detection Results	74
6.3	Voice of the Customer Management Results	88
6.4	Comparative Analysis of Different Models' Performance	94
<b>7</b>	<b>Conclusion and Future Work</b>	<b>100</b>
7.1	Conclusion	100
7.2	Future Work	101
	<b>Bibliography</b>	<b>101</b>
	<b>Appendix A</b>	<b>102</b>
	<b>Appendix B Overleaf: GitHub for L<sup>A</sup>T<sub>E</sub>X projects</b>	<b>108</b>

# List of Figures

1.1	Diagram of Research Methodology . . . . .	5
2.1	Flow of Telecom Network Evolution . . . . .	11
2.2	Telecom Network Evolution & Convergence . . . . .	12
2.3	Telecom Network Evolution & Convergence . . . . .	13
2.4	Layer of Telecom Networks . . . . .	14
2.5	Performance KPI Matrices . . . . .	15
2.6	ML Based Performance Assurance . . . . .	16
2.7	ML Based proposed VoC Management . . . . .	18
3.1	LTE Combined Attach . . . . .	22
3.2	LTE Combine Attach Histogram Plot . . . . .	24
3.3	LTE Combine Attach Scatter Plot . . . . .	25
3.4	VoC Sample Count . . . . .	26
3.5	VoC Word Length Distribution . . . . .	26
3.6	VOC Most Frequency Word . . . . .	26
3.7	Telecom Convergence Core Network Architecture . . . . .	32
6.1	ARIMA . . . . .	66
6.2	XGBoost . . . . .	67
6.3	Long Short-Term Memory . . . . .	68
6.4	Dynamic Linear Models . . . . .	69
6.5	Prophet . . . . .	71
6.6	Vector Autoregression . . . . .	72
6.7	Gated Recurrent Unit (GRU) Networks . . . . .	73
6.8	Density-Based Spatial Clustering of Applications with Noise . . . . .	74
6.9	Confusion Matrix of Density-Based Spatial Clustering of Applications with Noise . . . . .	75
6.10	Classification Report of Density-Based Spatial Clustering of Applications with Noise . . . . .	75
6.11	Isolation Forest . . . . .	76
6.12	Confusion Matrix of Isolation Forest . . . . .	77
6.13	Classification Report of Isolation Forest . . . . .	77
6.14	Local Outlier Factor . . . . .	78
6.15	Confusion Matrix of Local Outlier Factor . . . . .	79
6.16	Classification Report of Local Outlier Factor . . . . .	79
6.17	One Class SVM . . . . .	80
6.18	Confusion Matrix of One Class SVM . . . . .	81
6.19	Classification Report of One Class SVM . . . . .	81

6.20	Elliptic Envelope . . . . .	82
6.21	Confusion Matrix of Elliptic Envelope . . . . .	83
6.22	Classification Report of Elliptic Envelope . . . . .	83
6.23	Autoencoders . . . . .	84
6.24	Confusion Matrix of Autoencoders . . . . .	85
6.25	Classification Report of Autoencoders . . . . .	85
6.26	Histogram-Based Outlier Score . . . . .	86
6.27	Confusion Matrix of Histogram-Based Outlier Score . . . . .	87
6.28	Classification Report of Histogram-Based Outlier Score . . . . .	87
6.29	Performance Matrix of Support Vector Machine (SVM) . . . . .	88
6.30	Confusion Matrix of Support Vector Machine (SVM) . . . . .	89
6.31	Performance Matrix of Convolutional Neural Network (CNN) . . . . .	89
6.32	Confusion Matrix of Convolutional Neural Network (CNN) . . . . .	90
6.33	Performance Matrix of Gaussian Naive Bayes (GNB) . . . . .	91
6.34	Confusion Matrix of Gaussian Naive Bayes (GNB) . . . . .	91
6.35	Performance Matrix of Multinomial Naive Bayes (MNB): . . . . .	92
6.36	Confusion Matrix of Multinomial Naive Bayes (MNB): . . . . .	93
6.37	Performance Matrix of Logistic Regression (LR) . . . . .	93
6.38	Confusion Matrix of Logistic Regression (LR) . . . . .	94
6.39	Comparison of the algorithms accuracy of time series forecasting . . . . .	94
6.40	Comparison of the algorithms accuracy of anomaly detection . . . . .	96
6.41	Comparison of the algorithms accuracy of the Customer Management . . . . .	97
7.1	Plagiarism result from Turnitin Software . . . . .	108

# List of Tables

# Chapter 1

## Introduction

### 1.1 Background

Globally, the telecom industry is experiencing instability in the areas of compliance, operations, strategy, business and finances. Even if implementing innovation principles in light of technical improvements might be positive, the telecom industry must make sure it takes advantage of unpredictability.

Degrading Average Return per User (ARPU), uptrending churn, increasing competition, and high cost of technological convergence provide the main difficulties to the industry. Data services are getting more difficult to distinguish from one another, and the voice industry is practically at the end of its lifetime. As organizations are now required to offer distinctive client services, it is no longer sufficient to meet the traditionally anticipated consumer demands such as technical assistance, customer service, and network remediation. In this digital era, when reputations can be key to excellence, this requirement cannot be dismissed.

Sustainability requires a comprehensive strategy that incorporates ethical corporate values, continuous advancement in technology, innovation in customer experience and service assurance. In this research a AI-ML based methodology has been proposed for performance assurance and VoC management to enhance digitization and Sustainability in telecom industry.

The telecommunications industry is in the midst of a trans-formative era characterized by the convergence of technologies and services. Its high time to evolve to digital way of by dumping traditional human intervened working approach. Highly convergent mobile operator networks, where traditional boundaries between mobile, fixed-line, and internet services blur, have become the cornerstone of modern connectivity. This convergence promises increased efficiency, seamless user experiences, and a plethora of new opportunities. However, it also presents unprecedented challenges in terms of network performance assurance and the management of the Voice of the Customer (VoC).

In this context, the deployment of Machine learning (ML) techniques have gained a lot of traction as a tool to address these intricate challenges. ML offers the potential to not only enhance the performance of mobile operator networks but also

to harness valuable insights from customer feedback and behavior. The fusion of ML and telecommunications promises a paradigm shift in the way network operators ensure service quality and satisfaction.

**Highly Convergent Mobile Operator Networks:** Highly convergent networks represent a confluence of diverse services, such as voice, data, video, and IoT, gaming, cashing, cloudification, edge computing, network slicing, EMBB, Ultra low latency all integrated into a single platform. The coexistence of these services creates a complex ecosystem where network performance issues in one domain can ripple across others. Moreover, the introduction of emerging technologies like 5G adds further layers of complexity. As a result, ensuring the uninterrupted and optimal delivery of services within these networks has become a pressing concern.

**Machine Learning in Telecommunications:** Machine Learning techniques has eminence potential to open a new horizon in telecommunication industry, It must already have achieved outstanding results in a number of fields, such as natural language processing, computer vision, and data analytics. In the telecommunications industry, ML's potential to optimize network operations, detect anomalies, and predict failures has been recognized. However, applying ML effectively in the context of highly convergent networks requires extra attention from MNOs to increase its footprint for sustainable business growth.

**Voice of the Customer (VoC) Management:** In the highly competitive telecommunications landscape, understanding and meeting customer expectations are essential for retaining and attracting subscribers. ML offers a way to analyze vast amounts of customer data, including feedback, customers need, insights, complaints, usage patterns, and preferences, enabling network operators to tailor their services and improve customer satisfaction. However, developing comprehensive VoC management strategies that harness the full potential of ML is a multifaceted endeavor.

## 1.2 Problem Statement

As mobile operator networks continue to evolve towards higher levels of convergence, ensuring their optimal performance and meeting the demands of customers become increasingly complex. In this regard, this thesis seeks to explore the challenges posed by the convergence of mobile operator networks and the critical need for effective performance assurance and Voice of Customer (VoC) management. The current working practise of the telecommunication network operators are mostly tradition and manual approach, which are ineffective and labour-some. By leveraging the capabilities of Machine Learning, the research aims to develop innovative approaches that can comprehensively monitor, analyze, and enhance network performance, all while integrating real-time customer feedback to drive continuous improvements. Through this investigation, the study intends to contribute to the development of novel strategies for managing the intricate landscape of converged mobile operator networks. The ultimate goal is to not only elevate the overall quality of service or elevate customer satisfaction but fully autonomous network by use of AI-ML.

In today's rapidly evolving telecommunications landscape, mobile operator networks



face the challenges of delivering high-performance services near realtime, minimizing interruption and optimizing CAPEX and OPEX, while simultaneously managing the Voice of the Customer (VoC) effectively.

Machine Learning (ML) has surfaced as a formidable instrument for enhancing network performance assurance and VoC management. However, despite the potential benefits, there is a pressing need to address several critical issues within this domain.

**Network Performance Assurance:** The convergence of various technologies, including 2G, 3G, LTE, 5G, IOT, VAS, Cloud Core, Gaming, VR, and internet services, introduces intricate network inter dependencies. Ensuring high-quality service delivery in this context requires ML-based solutions that can predict, diagnose, and proactively address network performance issues, faults, degradation, anomalies. These solutions should adapt to the dynamic nature of these networks and deliver actionable insights in real-time.

**Voice of the Customer (VoC) Management:** Understanding and meeting customer expectations, insights, satisfaction are paramount in the telecommunications industry. ML can be harnessed to analyze customer feedback, preferences, and behavior patterns to enhance service offerings and improve overall customer satisfaction. However, a gap exists in developing comprehensive VoC management strategies that leverage ML effectively.

**Highly Convergent Mobile Operator Networks:** Adapting all technologies (2G, 3G, LTE, 5G, IOT, VAS, Cloud Core, Gaming, VR etc) in single architecture called convergence network. The specific challenges posed by highly convergent networks, where traditional boundaries between services blur, require specialized ML models and algorithms. These networks must manage a multitude of services, each with its unique performance requirements and customer expectations.

This thesis aims to address these challenges by developing innovative ML-based approaches tailored to the context of highly convergent mobile operator networks. By doing so, it seeks to provide network operators with the tools and insights necessary to not only ensure optimal network performance but also to create a seamless and satisfying experience for their customers.

### 1.3 Research System Goals

The primary research goals of the ML Based Performance Assurance and VoC Management of Highly Convergence Mobile Operator Network are as follows:

- **Enhance Network Performance:** The system should actively monitor network performance, predict potential issues, and optimize network resources to ensure high-quality service for users. The ultimate goal to introduction AI based self healing and self optimized network.
- **Prioritize Customer Experience:** VoC data should be analyzed to understand user experiences and concerns. The system should prioritize issues that directly impact customer satisfaction. which saves time and money of VoC

Management operation. Long term goal to implement AI based predictive customer experience analytic.

- Integration of ML: The system should seamlessly integrate ML algorithms for real-time anomaly detection, time series prediction, and VoC sentiment analysis. In these sector ML can enhance operational process and mean time and cost of remediation of each issues.

## 1.4 Research Objectives

The main goal of this research is to leverage machine learning techniques to ensure the network performance assurance ( system Quality of Service (QoS), network health, network availability, network KPI and KQI of a highly converged mobile operator network, while concurrently managing the Voice of the Customer (VoC) effectively. This research seeks to address the challenges posed by network convergence in the telecommunications industry and aims to provide practical solutions to optimize network performance and enhance customer satisfaction. The aim of this research are:

- Investigate the application of ML algorithms for real-time performance monitoring and fault detection in a highly converged mobile operator network.
- Develop ML models using ML techniques to proactively identify and mitigate network performance issues, Anomalies, ensuring Quality of Service (QoS).
- Obsolete traditional manual and highly human intervened performance monitoring and remediation process. Which is very ineffective and time consuming.
- Implement natural language processing and segmentation within the VoC management framework to extract actionable insights from customer feedback.
- Create a comprehensive VoC data collection and analysis system that incorporates various sources, such as customer surveys, social media, and call center interactions.
- ML to antiquate VoC Management Manual process of segment, escalation and fixation process.
- Address the unique challenges posed by network convergence, including the integration of different technologies, services, and network elements, within the ML-based performance assurance framework.
- Evaluate the effectiveness of ML-driven performance assurance and VoC management in enhancing customer satisfaction and network reliability through empirical testing and case studies.
- Provide practical recommendations and guidelines for mobile operators to implement ML-based solutions for improved network performance and customer-centric network management.

## 1.5 Research Methodology

This thesis's major research methodology will be a mixed-approaches approach that combines quantitative and qualitative research methods. Quantitatively, data related to network performance metrics, such as LTE attach, and failure trend, will be collected from the highly converged mobile operator network under study. This data will be subjected to analysis and machine ML to create predictive models for performance assurance. Additionally, quantitative data will be gathered from Voice of the Customer (VoC) sources, including customer surveys, social media sentiment analysis, and customer support interactions, to gain insights into customer feedback and sentiment. Qualitatively, these customer feedback datasets will undergo natural language processing and qualitative content analysis to extract meaningful insights. The combination of quantitative and qualitative data will enable a holistic approach to enhancing network performance and customer satisfaction within the highly converged mobile operator network.

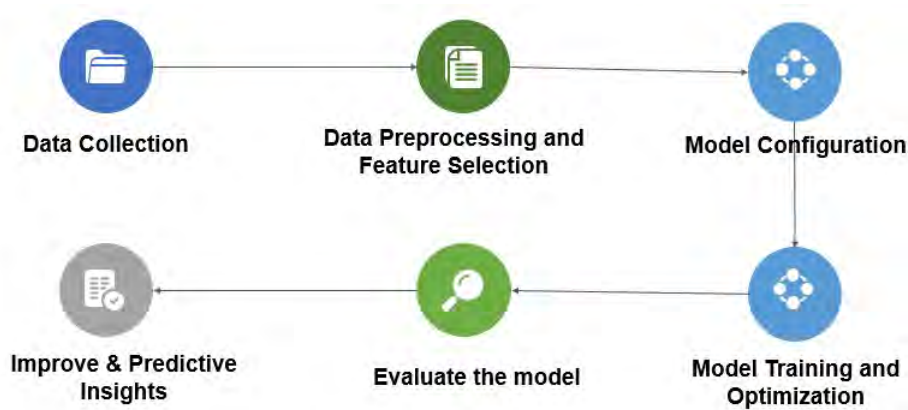


Figure 1.1: Diagram of Research Methodology

Furthermore, the research will employ a case study methodology, selecting one or more highly converged mobile operator networks as the research context. This will allow for in-depth exploration and empirical validation of the ML-based performance assurance and VoC management frameworks developed. The case study approach will involve real-world data collection and implementation of ML models within the chosen network(s). Key performance metrics will be monitored, and customer feedback will be analyzed to evaluate the impact of the proposed ML solutions. Additionally, expert interviews and focus groups with network engineers, operators, and customer service representatives will be conducted to gather qualitative insights and validate the practicality and effectiveness of the proposed methodologies. The triangulation of data from various sources and the combination of quantitative and qualitative approaches will ensure robustness and reliability in the research findings and contribute to the development of practical recommendations for mobile operators in the highly converged network domain.

The first method of this study focuses on Time Series Forecasting to predict network performance metrics. Several algorithms have been chosen to assess their effectiveness in this context. These algorithms include ARIMA (AutoRegressive Integrated

Moving Average), Vector Auto Regression (VAR), XGBoost, Prophet, LSTM (Long Short-Term Memory), and Dynamic Linear Model.

ARIMA and VAR are traditional statistical models widely used in time series forecasting. provide a benchmark for comparison with more sophisticated machine learning models. An ensemble learning method called XGBoost is well renowned for its great prediction accuracy. Facebook created Prophet, a specialized time series forecasting tool that can manage seasonality and vacations. A deep learning model called LSTM is capable of detecting intricate temporal relationships, making it suitable for sequential data like time series. Dynamic Linear Models are a Bayesian approach to time series forecasting, offering a probabilistic framework for modeling uncertainties.

The methodology involves preprocessing the historical network performance data, including data cleansing, normalization, and feature engineering, to make it suitable for input into the selected forecasting algorithms. These algorithms will then be implemented, trained, and fine-tuned using appropriate hyperparameters. Metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) will be used to assess each algorithm's performance. The generalizability of the models will also be ensured by the use of cross-validation techniques.

The second method of the research is Anomaly Detection, which aims to identify and flag abnormal network behavior or performance issues. One-Isolation Forest, Class SVM ,DBSCAN, Local Outlier Factor, Autoencoder, and Elliptic Envelope are just a few of the anomaly detection methods that will be used to accomplish this.

A supervised learning algorithm One-Class SVM classify data into two classes: normal and abnormal. Isolation Forest is an ensemble-based approach known for its ability to isolate anomalies efficiently. A type of neural network Autoencoder used for unsupervised learning and feature extraction, making it suitable for anomaly detection tasks. A density-based clustering technique called DBSCAN may be modified to find outliers. Measures the local density deviation of data points is the local outlier factor, making it effective for identifying local anomalies. Elliptic Envelope is a probabilistic model used to detect outliers by modeling the inlying data distribution.

For the Anomaly Detection method, the research methodology will involve preprocessing the network performance data similarly to the Time Series Forecasting component. A labeled dataset including both typical and abnormal network behavior will be used to train each anomaly detection method. Utilizing criteria like Recall, F1-score, Precision, and Receiver Operating Characteristic curves the effectiveness of these algorithms will be evaluated.

VoC (Voice of the Customer) Algorithm:

The third method of the study focuses on managing the Voice of the Customer to enhance network performance and user satisfaction. Several machine learning algorithms will be employed for this purpose, including Deep Neural Networks (DNN),

Support Vector Machines (SVM), Convolutional Neural Networks (CNN), Logistic Regression (LR), Gaussian Naive Bayes (GNB) and Multinomial Naive Bayes (MNB), and SVM is a versatile algorithm capable of both classification and regression tasks, making it suitable for VoC analysis. DNN and CNN are deep learning models known for their ability to process and extract meaningful information from unstructured data like customer feedback and comments. GNB and MNB are probabilistic models used for text classification and sentiment analysis, which are essential for understanding customer sentiment. An efficient approach for binary classification tasks is logistic regression.

The VoC Algorithm will involve data preprocessing steps such as text cleaning, tokenization, and feature extraction from customer feedback and reviews. The selected algorithms will be trained on labeled datasets to classify customer sentiment, identify common complaints or issues, and provide insights into improving network performance. Metrics like Recall, Confusion Matrix, Precision, F1-score, and Accuracy will be used to assess these algorithms.

In conclusion, the research methodology for the thesis "ML-Based Performance Assurance and VoC Management of Highly Convergent Mobile Operator Network" incorporates a comprehensive approach that involves the selection and implementation of machine learning algorithms for Forecasting Time Series, Anomaly Detection, and VoC Management. The evaluation of these algorithms using appropriate performance metrics will provide valuable insights for enhancing the performance and customer satisfaction of highly convergent mobile operator networks.

## 1.6 Research Questions

The following research questions will be covered in this paper:

1. How can Machine Learning (ML) techniques be effectively leveraged to enhance the real-time performance assurance capabilities of highly convergent mobile operator networks?
2. What are the most crucial Key Performance Indicators that need to be monitored and analyzed using ML algorithms to ensure the optimal functioning of highly convergent mobile operator networks?
3. How can Voice of the Customer (VoC) data be systematically collected, processed, and integrated into network management to enhance customer satisfaction and drive network improvements?
4. What ML-driven anomaly detection techniques are most suitable for identifying network irregularities and potential issues in highly convergent mobile operator networks, and how can they be seamlessly integrated into the existing network monitoring infrastructure?
5. What is the impact of ML-based performance assurance and VoC management on the overall network quality, operational efficiency, and customer perception in highly convergent mobile operator networks?

## 1.7 Research Scope

In this thesis, the primary scope revolves around the utilization of machine learning (ML) techniques for the enhancement of performance assurance and Voice of the Customer (VoC) management within the context of highly converged mobile operator networks. The research will delve into the application of various ML algorithms, focusing on their practical implementation to proactively monitor network performance, optimize Quality of Service (QoS) parameters, and address network convergence challenges. The study will also encompass the collection and analysis of VoC data from diverse sources, including customer feedback channels such as surveys, social media, and customer support interactions. The ultimate goal is to develop a comprehensive framework that leverages ML to enhance real time network reliability, customer satisfaction, and overall operational efficiency. Tradition work approach are time consuming and inefficient to manage huge volume of performance KPI and VoC management.

## 1.8 Research Limitations

Despite the ambitious scope, there are certain inherent limitations to this research. Firstly, proposed models rule based Robotic Process Automation (RPA) part is out of this research scope and the effectiveness of ML-based solutions heavily relies on the availability and quality of historical network data and VoC feedback. Limited access to such data or data quality issues may constrain the depth and reliability of the study's findings. Additionally, while the research aims to provide practical recommendations, the feasibility of implementing ML solutions may vary across mobile operator networks due to resource constraints, including computational resources and technical expertise. Furthermore, ethical considerations, such as data privacy and potential biases in ML algorithms, will be acknowledged but may not be comprehensively addressed within the immediate scope of this research. Finally, time constraints may limit the extent of data collection, analysis, and practical implementation and testing, potentially influencing the breadth of the study's outcomes.

## 1.9 Research Significance in telecommunication industry

This research is significant within the telecommunications industry as it addresses the pressing need for innovative solutions to ensure the performance, network fault, availability and Quality of Service (QoS) of highly converged mobile operator networks. With the proliferation of diverse services, technologies, and network elements in such environments, the effective application of machine learning (ML) techniques can be transformative. By developing and implementing ML algorithms for real-time network performance monitoring and predictive maintenance, this research

can contribute to a substantial reduction in downtime, improved network reliability, and optimized QoS. This, in turn, will lead to enhanced customer satisfaction and loyalty, which is critical for mobile operators in today's competitive landscape. Moreover, the inclusion of VoC management within the scope of the research ensures that customer feedback is not only collected but also utilized to drive network improvements, aligning network performance more closely with customer expectations. This customer-centric approach is vital for maintaining a positive brand image and ensuring long-term success in the telecommunications sector.

Beyond the telecommunications industry, the research significance extends to the broader field of machine learning and artificial intelligence. It offers a real-world application of ML in a complex and dynamic network environment, serving as a valuable case study for academia and industry practitioners alike. The research outcomes have the potential to inform best practices and guidelines for ML adoption in similar contexts, demonstrating the adaptability and effectiveness of ML algorithms in optimizing operations and customer experience. Furthermore, by addressing the challenges of network convergence, the research can pave the way for future developments in the management of integrated technologies, which is relevant not only in telecommunications but also in other sectors undergoing digital transformation. In summary, this research carries substantial implications for both the telecommunications industry and the broader field of machine learning, contributing to improved network performance, customer satisfaction, and the advancement of ML applications in complex operational environments.

## 1.10 Thesis Organization

In Chapter 2 of the thesis, a comprehensive literature review of related works is presented, structured as follows: it provides an introduction to the pertinent technologies examined in this study and offers in-depth insights into the interconnected research endeavors, Related Studies and Research in Telecom Network Analysis. Also discuss the advancement.

Chapter 3 presents a collection of the data, data types, data collection methodology, data preprocessing techniques, and proposed solution and further into the assumptions considered in this thesis. Analysis of the data considered is also presented.

Chapter 4 presents a general overview of the telecom evolution and customer behavior analysis, telecom network availability analysis, technology evolution prediction, customer behavior analysis considered in this thesis.

Chapter 5 shows the Data Training and Validation data Splitting , and Test Sets performed. The data segregation into training, validation, and test sets, model training process for availability prediction, model training process for technology evolution prediction, model training process for customer complaint analysis, model deployment strategies for mobile network operators, integration with existing network management systems considered in this thesis.

In Chapter 6 of this thesis, the case studies, experiments, and results are presented.

The outcomes are thoroughly examined, including an analysis of the factors contributing to the numerical findings and an exploration of the impact of specific features compared to others. Additionally, a comparison study of the performance of various models considered in this study is conducted.

Chapter 7 serves as the conclusion of this thesis, outlining future directions for research and summarizing the primary contributions made throughout the study.



# Chapter 2

## Literature Review

The goal of this literature study is to present a thorough analysis of the body of knowledge about the convergence of mobile operator networks, performance assurance using machine learning, and Voice of Customer (VoC) management. The convergence of telecom-networks has been a significant trend in the telecommunications industry, driven by the growing demand for data and the need to support various services efficiently. This chapter will explore the key concepts and advancements in these areas, laying the foundation for the research conducted in the thesis.

The telecommunications industry is ongoing a significant alteration with the convergence of numerous services, such as voice, data, and video, over a single network infrastructure. This convergence, facilitated by advances in mobile and broadband technologies, presents both opportunities and challenges for mobile operators. Ensuring the performance and quality of service in these mobile operator networks is crucial to meeting customer expectations and maintaining competitiveness. Machine learning (ML) has become an effective instrument for addressing these challenges by replacing the traditional working method, particularly in the context of Performance Assurance and Voice of the Customer (VoC) management.

### 2.1 Evolution of Mobile Operator Networks

Mobile operator networks have undergone a remarkable evolution over the years, shaped by advancements in technology and the changing demands of users. This section explores the key stages in the evolution of mobile networks, leading up to the highly converged environment we see today.

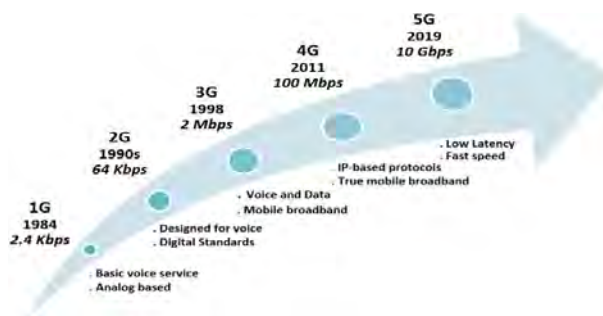


Figure 2.1: Flow of Telecom Network Evolution

### 2.1.1 Early Mobile Networks

The inception of mobile networks dates back to the 1980s with the advent of first-generation (1G) cellular networks. These networks were analog and primarily offered voice services. They were characterized by limited coverage and capacity.

### 2.1.2 Digital Revolution: 2G and 3G Networks

The transition to digital networks marked the emergence of second-generation (2G) and third-generation (3G) networks. 2G networks introduced digital voice calls, while 3G networks brought data services such as mobile internet and video calling. These advancements expanded the range of services but led to network complexity.

### 2.1.3 Data-Centric Networks: 4G and 5G

The deployment of fourth-generation (4G) and fifth-generation (5G) networks signaled a shift towards data-centric services. 4G networks offered high-speed data connectivity, paving the way for video streaming and mobile applications. 5G networks, with their low latency and massive device connectivity, have unlocked the potential for IoT and industrial applications.

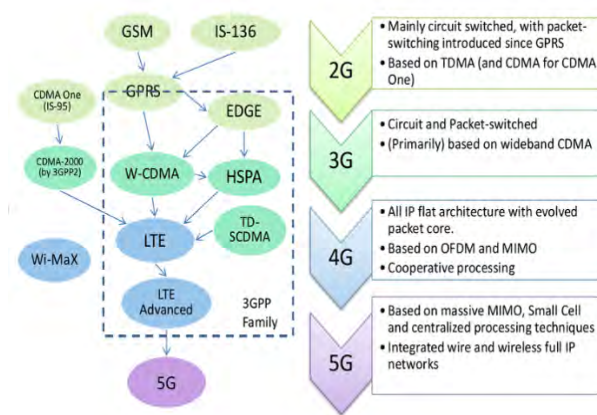


Figure 2.2: Telecom Network Evolution & Convergence

## 2.2 Related Studies and Research on Telecommunication Networks

The concept of network convergence involves the integration of multiple network technologies, including cellular, Wi-Fi, fixed-line networks, into a unified infrastructure, VAS services, IoT, IT Infrastructure, cloud computing, virtualization etc. Convergence aims to optimize resource utilization, reduce operational costs, and deliver a seamless experience to users and open a horizon for next generation evolution.

This paper conducted a comprehensive study on the effect of network convergence on network performance, Service quality, Service Assurance and performance Assurance

.Their research highlighted the potential for converged networks to efficiently utilize resources, improve QoS[20].

This work represent a comprehensive comparison of models like ARIMA, LSTM, and Prophet models,GRU in forecasting time series analysis on performance management . The study utilizes real network data from a major Asian mobile operator to access the performance of these models. Outcome shows LSTM, GRU outperforms ARIMA,VAR, and Prophet, demonstrating its potential as an accurate predictor for time series forecasting [65].

The study utilizes real-time network data from a major Asian mobile operator to evaluate network anomaly detection algorithms the performance of these models. Results show that Elliptic Envelope and Isolation Forest better in Anomaly detection [63].

This research explore options for anomaly detection in telecommunication network with OneClassSVM, AutoEncoder, Isolation Forrest model.Quality of available data, proper model most crucial part for anomaly detection , it directly impact to the models performance.In usual cases most of the models are validated by public datasets or by the the simulation, it is required to validate these models in the real network data. [47].

While network convergence offers numerous benefits and opportunity to the MNO’s to accelerate business, it presents a set of challenges. Managing diverse technologies and ensuring seamless integration,operations, management while maintaining high performance are among the foremost challenges.

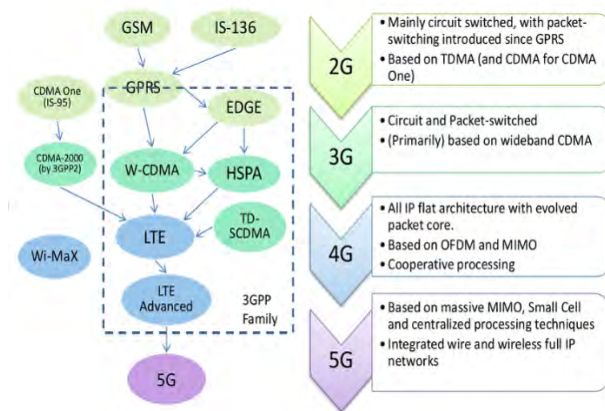


Figure 2.3: Telecom Network Evolution & Convergence

The intricacies of managing network convergence in dynamic environments. Their study emphasized the need for adaptive management solutions that can respond to changing network conditions [21]. Security concerns in converged networks were examined by stressing the importance of robust security measures [22].

This research, customer VoC segmentation has been performed by text classification for improving MNOs traditional work approach which also helps churn prediction.

This research focused by the consideration that the improvement of customer satisfaction will reduce churn and the customer satisfaction will be reflected in improvement by applying on VOC, the unorganized VoC information which captures a review of customer’s insights, expectation and feed backs. To the best of our knowledge, this is the unique work that introduces segmentation of VOC to MNOs working approach improvement[37].

## 2.2.1 Concept of Network Convergence

The telecom sector’s evolution, from the inception of 2G networks to the emergence of 3G, 4G, and the anticipated 5G technology. This section emphasise the key technological evolution, and the shift towards digitalization, Cloudification and data-driven communication systems [27].

Recent years have seen a substantial increase in interest in network convergence, or the merging of many network technologies into a single infrastructure. The convergence aims to enhance resource utilization, reduce operational costs, and deliver seamless services to end-users. Researchers and industry experts have explored various aspects of network convergence.

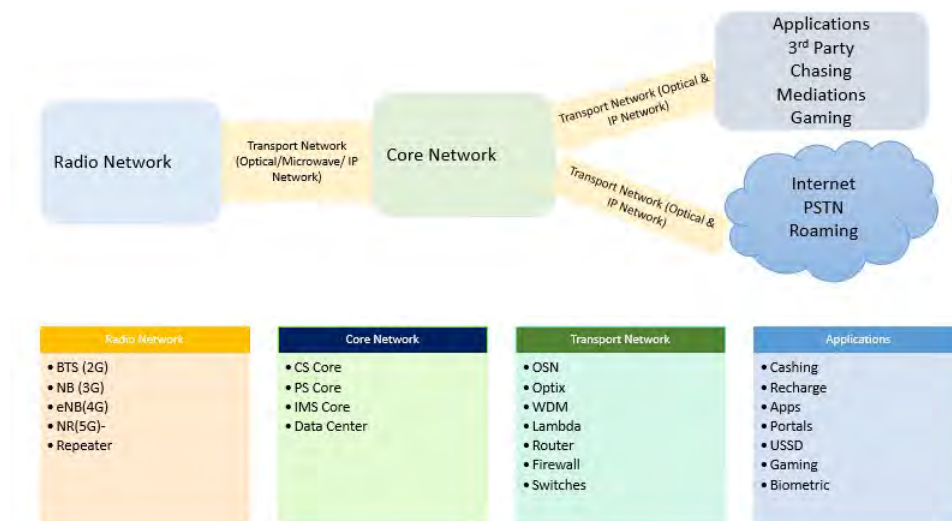


Figure 2.4: Layer of Telecom Networks

The investigated the impact of network convergence on resource allocation and Quality of Service (QoS). Their study revealed that converged networks can achieve more efficient resource utilization, leading to improved QoS and cost savings. Convergence has also been a driving force behind the emergence of 5G networks [23]. In this paper discussed the benefits of network slicing in 5G, allowing the isolation of resources for different services, ensuring efficient resource utilization [24].

## 2.2.2 Challenges in Network Convergence

While the benefits of network convergence are evident, it comes with its set of challenges. Convergence of technologies making network more complex, hard to manage by traditional tools and process. Most of the time its required hours to isolate fault and anomaly identification. Ensuring the seamless integration of diverse network technologies while maintaining high performance remains a complex task.

The focused on the challenges of managing network convergence in highly dynamic environments. They highlighted the need for intelligent management solutions that can adapt to changing network conditions. Network security is another critical concern [25]. These Author explored security issues in converged networks, emphasizing the importance of robust security measures to protect against cyber threats [26].

## 2.3 Performance Assurance in Mobile Networks

### 2.3.1 Key Performance Indicators (KPIs)

Performance assurance in mobile networks revolves around monitoring and optimizing Key Performance Indicators (KPIs). KPIs are essential metrics that assess the quality and efficiency of network services. Common KPIs include call success rate, Network Accessibility, Network Availability, Call Drop rate, MOS, data throughput, latency, packet loss, throughput, and signal strength etc.

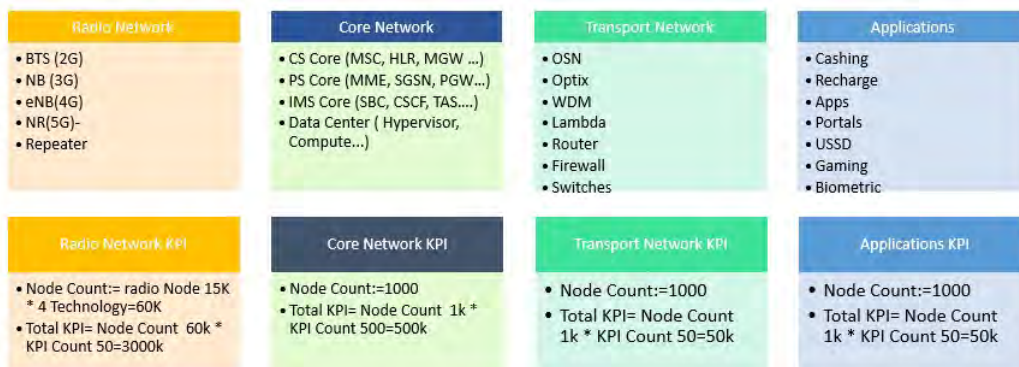


Figure 2.5: Performance KPI Matrices

### 2.3.2 Traditional Approaches to Performance Assurance

There are 3.5M KPIs and performance data generated every hour in MNOs. Traditionally, performance assurance in mobile operator networks relied on manual monitoring and intervention. Its very hectic task to unsure 3.5Million KPIs manually and its really time consuming. Usually team monitor KPIs 12Hr interval Hence, Network performance, downtime, service quality hard to ensure. However, strategy is no longer adequate given the complexity and size of contemporary networks.

This author introduced an early ML-based approach for network fault, KPI degradation and anomaly prediction. Their system utilized historical data to train predictive models, significantly improving the accuracy of fault prediction and reducing downtime. Similarly [27], In this paper authors explored the concept of SONs and how ML-driven optimization can automate network management tasks, leading to enhanced performance [28].

### 2.3.3 Machine Learning for Performance Assurance

The advent of Machine Learning (ML) has brought a paradigm shift in performance assurance. ML techniques can analyze vast volumes of network data in real-time, detect anomalies, and proactively optimize network parameters.

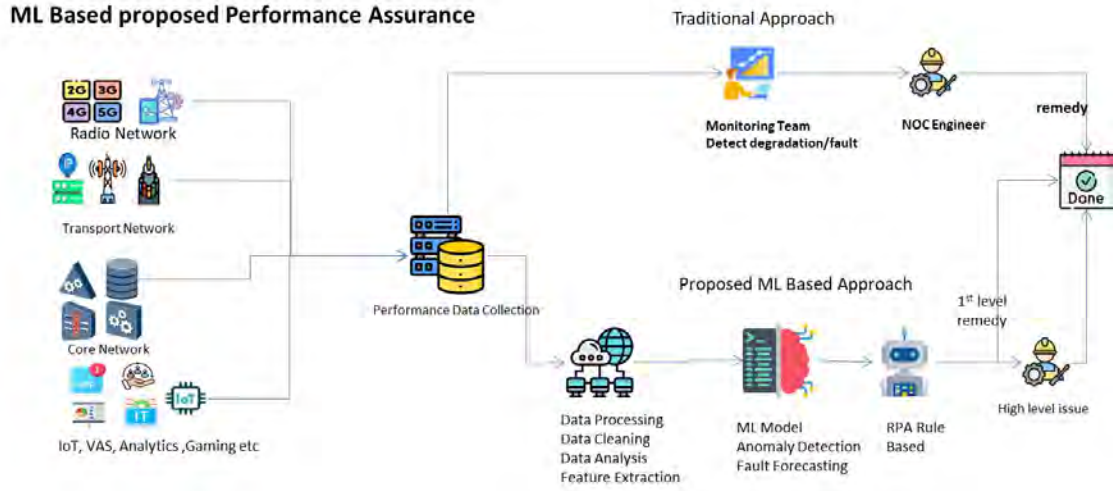


Figure 2.6: ML Based Performance Assurance

In this research, ML based performance assurance model proposed their MNOs can obsolete their tradition working approach, where a monitoring team constantly monitoring KPI, KQI and degradation. In case of degradation the issue raised to maintenance engineer for remediation. However, in the proposed ML model to detect real time service quality degradation and network fault in service and a rule based RPA to remediate and fix the technical issue.

This authors conducted research on ML-driven network performance management, highlighting the benefits of anomaly detection algorithms in predicting network failures. Their study showcased the potential for ML to enhance network stability and reliability [29]. Furthermore, they explored the use of ML for auto resource balance in 5G networks, ensuring efficient utilization of network resources and consequent performance improvements [30].

## 2.4 Voice of the Customer (VoC) Management

VoC Management is refer to the process of systematically capturing, analysing, investigation, addressing and utilizing customer feedback, issues, enquiries, requirements and insights to improve products, services, policies, and overall customer experience. VoC Management involves various methods of collecting customer feedback, issues, and insights such as service centre, mobile application, mail, surveys, reviews, social media mentions, customer support interactions, and more.

Understanding the Voice of the Customer (VoC) is critical for effective network management and delivering superior user experiences. VoC data provides insights into customer satisfaction, service quality, and areas needing improvement.

### **2.4.1 Importance of VoC in Network Management**

VoC management involves collecting and analyzing customer feedback to improve services and customer satisfaction. In the context of mobile operator networks, VoC data provides valuable insights into user experience and network performance enhancement.

In this work focused on conducted a comprehensive study on VoC analysis in the telecommunications industry. They highlighted the significance of timely feedback in addressing network issues and proposed a framework for integrating VoC data into network management systems [31]. This author contributed to the field by applying sentiment analysis techniques to extract valuable insights from customer feedback related to network services [32].

### **2.4.2 Challenges in VoC Management**

Effectively managing VoC data poses challenges, particularly in complex mobile operator networks. The volume and diversity of data generated by modern telecommunications services require advanced analytic and management strategies. Main Challenges are:

- Multi channel VoC source
- Manual Process
- Huge Data Variety and volume
- Inadequate tools
- ineffective Communication
- Frequent technology advancement
- Network Complexity
- Service Quality and reliability

This author discussed the challenges of handling large-scale VoC data and proposed a Big Data analytics framework to extract meaningful insights. They highlighted the potential of this approach in improving VoC management in converged networks [33]. Additionally, in this paper emphasized the importance of real-time VoC analysis and its role in identifying and addressing network issues promptly [34].

### **2.4.3 Machine Learning in VoC Management**

Traditional VoC management process where executives manually go through VoC and segregate based on serving Area. Later the VoC escalated to concern team based on serving area mapping. Related concern team implement remedy.

In this thesis, ML based VoC management process has been purposed. Where MNOs traditional working approach to segment VOC manually and escalate to the concern technical team to be replaced by ML. A rule based RPA to provide 1st level remediation and resolution of VoC.

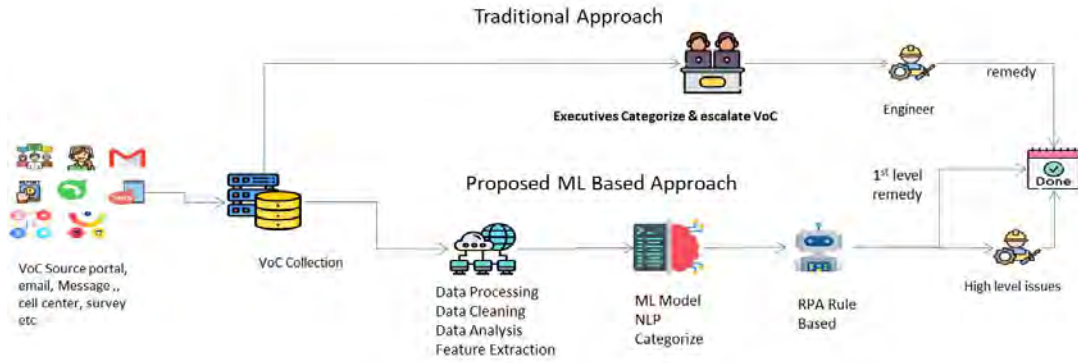


Figure 2.7: ML Based proposed VoC Management

## 2.5 ML-Based Performance Assurance and VoC Management Integration

Integrating Machine Learning (ML) into performance assurance and VoC management is a powerful strategy for enhancing network quality. ML algorithms can analyze vast datasets, detect anomalies, and predict network issues in real-time. Integrating ML-based performance assurance with VoC management enables operators to proactively identify and address network problems, resulting in improved customer satisfaction and operational efficiency.

On the other hand the integration of ML-based performance assurance and VoC management is a relatively unexplored area but holds significant potential. By combining real-time network performance monitoring with customer feedback analysis, operators can proactively identify and address issues, ultimately improving customer satisfaction.

This paper the author experimented a framework that integrates ML-based network performance monitoring with VoC data analysis. Their study demonstrated that this integration can lead to quicker issue resolution and improved network performance [35]. Similarly, other author developed a system that uses ML algorithms to correlate network KPIs with customer complaints, enabling operators to prioritize and resolve issues effectively [36].

## 2.6 Performance Assurance and VoC Management in ML Techniques

### Anomaly Detection:

Anomaly detection is a critical component of performance assurance in highly convergent mobile operator networks. Detecting unusual behavior or anomalies in network data can help operators proactively identify and address issues. Historically, traditional statistical methods and rule-based systems were used for anomaly detection. However, with the growth in data complexity, Machine Learning (ML) techniques have gained prominence.



An analysis of time series data anomaly detection methods was suggested by this author. Their work highlighted the limitations of traditional methods and the need for advanced ML algorithms [37].

Clustering, classification, and deep learning etc Machine Learning techniques such have shown promise in anomaly detection. Isolation Forest, One-Class SVM, and Autoencoders are examples of ML algorithms commonly used for this purpose. On the other hand this author introduced a comprehensive review of anomaly detection techniques, including statistical, neural network-based, and clustering methods. They discussed the challenges and trade-offs associated with different algorithms [38].

In the context of mobile networks, anomaly detection plays a crucial role in identifying network faults, security breaches, and performance irregularities. This authors applied Random Forest and SVM for anomaly detection in mobile networks [39]. Their research highlighted the effectiveness of ML algorithms in improving fault prediction.

#### **Time Series Prediction:**

Time series prediction is vital for forecasting network performance, traffic patterns, and resource utilization. Accurate predictions enable proactive network management and resource allocation.

ARIMA and Exponential Smoothing are common time series prediction techniques. While these methods are widely used, they may not capture complex patterns in network data. This authors provided an extensive guide to time series forecasting, covering traditional approaches and their limitations [40].

Time series prediction has completely changed as a result of machine learning, particularly as it relates to RNNs and LSTM neural networks. In this study, writers analysed the scope of deep learning methods, such as LSTMs, for time series forecasting [41]. Their work experimented huge improvements in prediction accuracy compared to traditional methods.

Time series prediction is indispensable for anticipating network traffic, resource demand, and service quality in mobile networks. In thgis paper employed LSTM networks for time series prediction in 5G networks [42]. Their research showcased the potential of deep learning in optimizing resource allocation and QoS.

#### **VoC Management:**

Voice of the Customer (VoC) management is essential for understanding user experiences, preferences, and grievances. Algorithms that can efficiently process and analyze VoC data are integral to network management.

Sentiment analysis, a subfield of Natural Language Processing (NLP), is commonly used for VoC management. It involves classifying customer feedback as positive, negative, or neutral, providing insights into customer sentiment. This authors presented a comprehensive review of sentiment analysis process, including dictionary-based,

machine learning-based as well as deep learning-based approaches [43]. The study emphasized the applicability of sentiment analysis in understanding VoC.

ML models and techniques, including classification, clustering, and topic modeling, can be employed for VoC analysis. In this paper developed a VoC management system using text classification algorithms [44]. Their research demonstrated how ML can automate the categorization of customer feedback, making it easier for operators to identify common issues. But in this paper I used DNN, CNN, SVM, GNB, MNB, and LR model in VoC management.

The integration of Anomaly Detection, Time Series Prediction, and VoC Management Algorithms in highly convergent mobile operator networks is a promising avenue for network optimization and customer satisfaction improvement. The proposed a comprehensive framework that integrates Anomaly Detection with Time Series Prediction for network performance monitoring [45]. Their study showed that combining these techniques can lead to quicker issue resolution and improved network performance. Additionally, the integration of VoC Management Algorithms can enhance the understanding of customer experiences, enabling more targeted network improvements.

## **2.7 Research Gaps in the Literature**

While significant progress has been made in the areas of network convergence, ML-based performance assurance, and VoC management, there is a notable gap in the literature concerning the integration of these concepts. Existing studies have established the benefits of each area separately, but the synergies and challenges of combining them remain relatively unaddressed. Network performance assurance and VoC management building ML models and identify best suited models in the key scope in this exercise.

# Chapter 3

## Methodology

### 3.1 Research Design

This chapter outlines the research approach taken to examine how machine learning (ML) for performance assurance and Voice of the Customer (VoC) management may be integrated in highly convergent mobile operator networks. The study plan, methods for gathering data, tools for analyzing data, and ethical issues are all discussed in detail.

### 3.2 Research Approach

**Quantitative Research Approach:** This study primarily adopts a quantitative research approach. Quantitative research is suitable for examining the relationships between variables, measuring performance metrics, and conducting statistical analysis on large datasets. This approach is well-suited for assessing the impact of ML-based performance assurance and VoC management on network performance.

**Qualitative Research Approach:** In addition to the quantitative approach, a qualitative research approach will be employed to gather insights from network operators and users. Qualitative research methods, such as interviews and surveys, will be used to collect VoC data and gain a deeper understanding of user experiences and preferences.

### 3.3 Data Collection

#### 3.3.1 LTE Attach

The KPI has chosen for this experiment has immense importance in MNOs Network. LTE Network Attach is a metrics which signifies of LTE(4G) users accessibility connectivity to LTE IMS network.

When a UE initiates a procedure for attaching to the EPC network for fresh location update, the UE includes its IMSI in the Attach Request message. LTE attach procedure includes user authentication, location update in HLR, PCRF data quota reservation, radio resource connection with eNB.

LTE combined attach success rate= $(\text{LTE join attach 4G and non-4G services success times} + \text{LTE EPS services and SMS join success times EPS services only} + \text{LTE join attach success times for EPS services only} + \text{LTE join attach success times for EPS services only}(\#18 \text{ CS domain not available})) / \text{LTE combined attach request times ( Point 2, M5)} * 100\%$

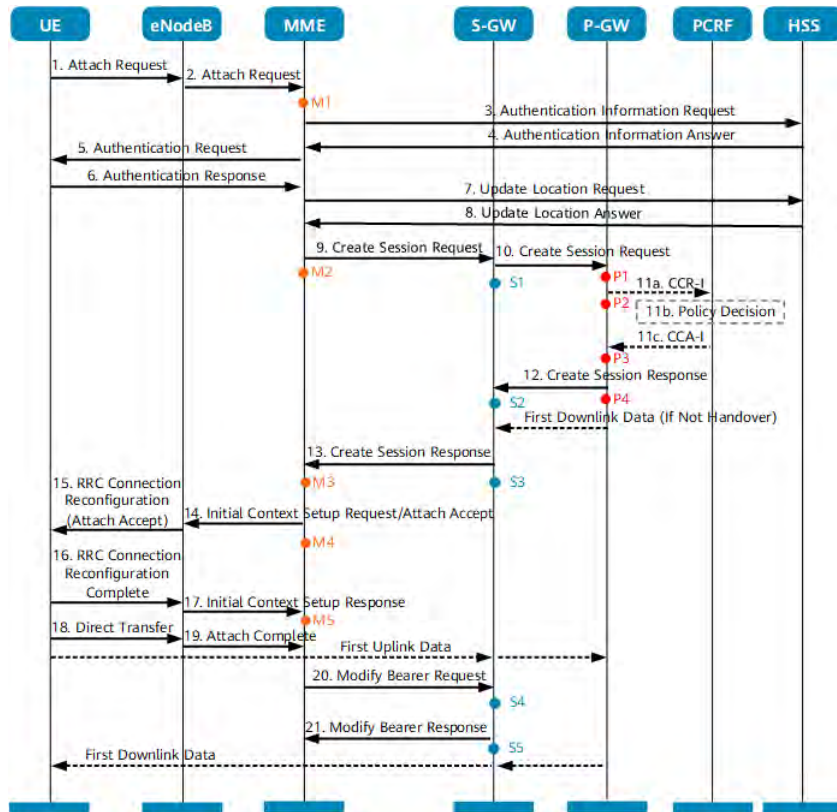


Figure 3.1: LTE Combined Attach

If a UE initiates a procedure for attaching to the EPC network for fresh attach, in the Attach Request message IMSI included by UE. The initial attach consists of the following phases:

- S1 initial attach message sent along with signaling establishment (IMSI carried) The UE creates a connection with the LTE eNodeB over the S1 interface.
  - Authentication  
The MME transfers the IMSI, obtains the authentication quadruplets from the HSS, and helps the network and handset both authenticate with one another. After authentication succeeds, the MME obtains UE subscription data from the HSS.
  - Location update  
The MME generates LU Request message to register with the HSS.
  - Default bearer establishment  
Default bearer established on the EPC network based on the default APN and PDN subscription context in the UE subscription data. After the default bearer is created, the UE successfully attaches to the EPC network.
- As shown in Figure 3.1, Mx is a measurement point on the MME, Sx is a measurement point on the S-GW, and Px is a measurement point on the P-GW.
- MME:

M1; M2; M3; M4; M5

- S-GW:

S1; S2; S3; S4; S5

- P-GW:

P1; P2; P3; P4

1. Attach Request message sent by UE to the eNodeB.
2. The eNodeB transparently transmits the Attach Request message to the MME for establishing an S1 signaling connection. The message carries the IMSI, TAI, and S1 interface-related IDs.
3. The eNodeB transparently transmits to MME the Attach message for establishing an S1 signaling connection. The message carries the IMSI, TAI, and S1 interface-related IDs.
4. The MME sends an Authentication Request message to the UE upon receiving the Authentication Information Answer message. The message contains RAND, AUTN, and KSI<sub>asme</sub> (which functions as the Key Set Identifier for K<sub>asme</sub>).
5. The network is verified by the UE. The UE computes the RES using the RAND and sends the RES in an Authentication Response message to the MME if the network authentication is successful. Once the UE has been verified, the network. The RES and XRES in the quadruplets of authentication are compared by the MME. UE authentication is successful if they are identical. In the absence of that, the MME notifies the UE that its authentication has been rejected.
6. To collect subscriber data, the MME contacts the HSS with an Update Location Request message. The MME attaches first using the HSS domain since it lacks the UE's valid subscription context.
7. By communicating with the MME via an Update Location Answer message, the HSS adds subscription data. One or more PDN subscription contexts and a default APN are included in this subscription data. Each PDN subscription context has an EPS subscribed QoS profile and the subscribed APN-AMBR. The MME denies the UE's request to attach if the UE connects to the network using an APN to which it has not subscribed or if the HSS denies the Update Location request.
8. the UE's Attach Request message includes an APN, the MME utilizes that APN to initiate the default bearer. In the absence of an APN, the MME resorts to the default APN associated with the subscription for activation. The MME, based on the Tracking Area Identity (TAI), retrieves a list of S-GWs through DNS resolution. Simultaneously, it retrieves a list of P-GWs based on the APN through DNS resolution as well. Subsequently, the MME selects a combination of S-GW and P-GW to establish a default bearer, taking into account the priorities and weights of available S-GWs and P-GWs. It follows the principle that a combined S-GW/P-GW is preferred, and proximity in terms of network topology is prioritized. The MME then assigns an EPS bearer ID to the default

bearer and forwards a Create Session Request message to the selected S-GW, requesting the establishment of the default bearer. Key Information Elements (IEs) contained within this message are described as follows.

9. The S-GW establishes an EPS bearer in the bearer list and sends the CCR message to the P-GW based on the P-GW IP address carried in step 9. The message contains the TEID of the S-GW, IP address of the S5/S8 interface, and QCI. The S-GW caches the downlink packets delivered by the P-GW upon receiving a CCR message from the P-GW, and forwards the packets after obtaining the eNodeB TEID from the Modify Bearer Request message in step 20.
10. Optional: If dynamic Policy and Charging Control (PCC) is used, the P-GW initiates a procedure for establishing an IP-CAN (IP-connectivity access network) session the UE.to obtain default PCC rules ,the P-GW uses locally configured policies.
  - 10a. The P-GW sends a CCR-I message to the PCRF, instructing the PCRF to create an IP-CAN session.
  - 10b. The PCRF performs authorization and policy decision-making.
  - 10c. The PCRF responds to the P-GW with a CCA-I message, carrying the selected IP-CAN bearer establishment mode.
11. The P-GW establishes an EPS bearer within the EPS bearer list and generates a charging ID. It is capable of forwarding UP PDUs between the S-GW and the PDN. Charging initiation occurs at the P-GW, and a Create Session Response message is transmitted to the S-GW.

### 3.4 Data Set for Performance Assurance

Performance assurance experimental data set is one of the key parameter of LTE network.The performance indicate who stable and efficient the LTE data network. This is one of the KPI among 3.5M KPIs but its very criticial to the MNOs data network.Key characteristics of the dataset:

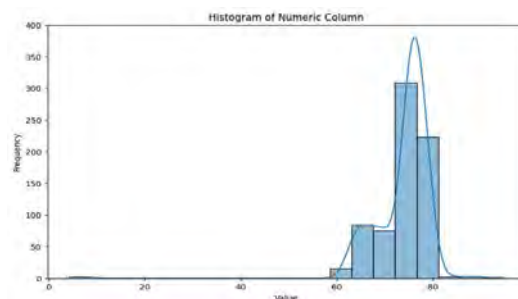


Figure 3.2: LTE Combine Attach Histogram Plot

- 1.The dataset is time series data of 1hr interval data.

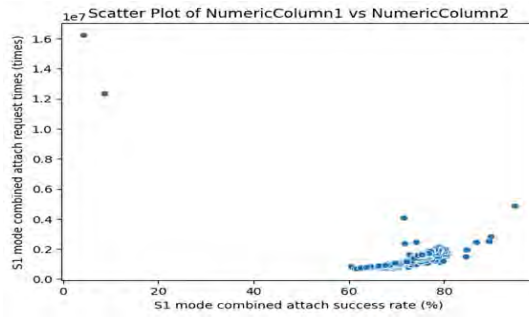


Figure 3.3: LTE Combine Attach Scatter Plot

2. There 714 rows and 16 columns in the dataset.

3. “LTE combined attach success rate” represent LTE rate and “LTE combined attach request times” represents attach request from user equipment.

4. LTE join attach failure count( 19 ESM failure) and LTE join attach failure counts (ESM failure 29 User authentication failed) demonstrate attach failure counts.

5. Histogram Plot most frequency of success rate is 70-75.

### 3.5 Data Set for VoC Management

VoC related to customer’s complain, enquiries, insights, information, suggestions etc are collected from various source. MNOs VoC Management Categorization , Analysis and Investigation are done manually. Usually a Customer experience executive reads insight manually and segregate of its Business area/type based on his understanding.

Key characteristics of the dataset:

1. In this experiment VoC data of 2834 samples has been used.
2. There are 5 columns in this data set among them “AREA” and “Description” Column will used on this experiment.
3. “Business Area” refers to a specific business unit/vendor/concern who will be responsible to take care the VoC.
4. There are 85 unique Business Area, among them sample count  $\geq 60$  considered for the experiment.
5. “Description” Column is the detail of customers VoC. “Description” Column Data is unorganized.
6. There are Special characters, Banglish words, unreadable contents, date, time, amount are exists this data set.

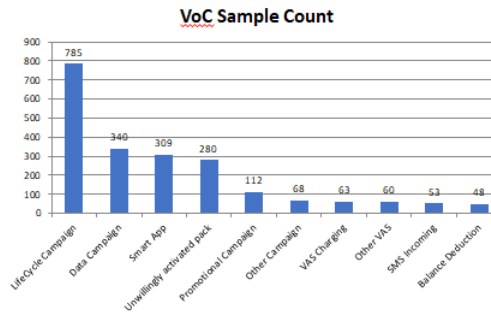


Figure 3.4: VoC Sample Count

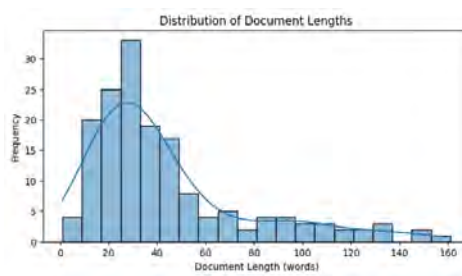


Figure 3.5: VoC Word Length Distribution



Figure 3.6: VOC Most Frequency Word

### 3.6 Time Series Algorithm Selection and Implementation

Selecting and implementing the appropriate time series prediction algorithm is crucial for accurate forecasts in network optimization. This process involves choosing from various algorithms like ARIMA, XGBoost, LSTM, Dynamic Linear Model, Prophet, VAR, and GRU based on the character of the data and the specific requirements of the network. Equipped with these algorithms, their mathematical foundations, and insights from the data, the prediction process can be outlined as follows:

**Data Preprocessing:** Prepare the time series data for analysis by cleaning, converting, and organising it. This might include dealing with missing values, scaling,



and dividing the data into training and testing sets.

**Algorithm Selection:** Select an algorithm that is suitable for the data properties and forecasting goals. For instance, if the data exhibits strong seasonality, Prophet or ARIMA might be appropriate. If the data is complex with nonlinear patterns, XGBoost or LSTM might be considered.

**Algorithm Parameter Tuning:** Many algorithms have hyperparameters that need to be optimized to achieve the best performance. This is accomplished by employing approaches such as random search and grid search to choose the best set of parameters.

**Model Training:** Train the chosen algorithm on the training data to extract the patterns and correlations found in the historical data.

**Equation Implementation:** Depending on the chosen algorithm, equations specific to that algorithm are used during the training process. For instance, in ARIMA, the autoregressive and moving average coefficients are calculated iteratively. In LSTM, the equations governing the gates and cell state updates are utilized for forward and backward passes.

**Model Evaluation:** Using the testing dataset, evaluate the trained model's performance. Prediction accuracy is typically measured using metrics such as Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE) etc.

**Forecasting:** Once the model's performance has been determined to be good, it may be used to forecast future data points.

**Visualization:** To evaluate the model's predictions' accuracy graphically, compare the projected values to the actual values.

**Iterative Refinement:** Depending on the forecasting results, you might need to go back to steps 2-6 to fine-tune the algorithm and improve the predictions.

In conclusion, the process of time series prediction algorithm selection and implementation involves a thoughtful evaluation of data characteristics and forecasting goals. The equations underpinning each algorithm guide the training and prediction process, enabling accurate network optimization and informed decision-making.

### 3.7 Anomaly Detection Algorithm Selection and Implementation

Here's a detailed explanation of the steps involved in selecting and implementing anomaly detection algorithms, along with equations for some common methods:

**Insights of Anomaly Detection:** Anomaly detection is the process of identifying

patterns and data that deviate significantly from the norm. It's crucial for network security and performance optimization as it helps in identifying unusual behaviors that might indicate potential issues.

**Data Preprocessing:** Similar to time series prediction, data preprocessing is essential. Clean the data, handle missing values, and standardize or normalize it if necessary. Data that isn't preprocessed well can lead to inaccurate anomaly detection.

**Algorithm Selection:** Choose an anomaly detection algorithm based on the character of your data and the types of anomalies you're interested in detecting. Algorithms include DBSCAN, One Class SVM, Isolation Forest, Elliptic Envelope, Local Outlier Factor (LOF), and Autoencoders.

**Algorithm Parameter Tuning:** Most anomaly detection algorithms have parameters that need tuning for optimal performance. For instance, in DBSCAN, you'd set parameters like the minimum number of points and the radius. In Isolation Forest, the number of trees could be tuned.

**Model Training:** In anomaly detection, "training" often involves creating a model that captures the normal behavior of the data. This model then helps identify deviations from the norm, which are considered anomalies.

**Equation Implementation:** While anomaly detection algorithms might not have explicit equations like those in prediction algorithms, they are guided by specific principles. For instance, Isolation Forest uses the length of the path traversed in a tree to isolate anomalies. The local density deviation of a data point in relation to its neighbors is calculated using the Local Outlier Factor (LOF).

**Model Evaluation:** Use relevant metrics to evaluate the effectiveness of your anomaly detection model, such as F1-score, accuracy recall, or Area Under the ROC Curve (AUC-ROC). These metrics help quantify how well the model is detecting anomalies.

**Threshold Setting:** Determine a threshold for classifying data points as anomalies. This threshold is often set based on the evaluation metrics, the balance of and false negatives and false positives as well as the acceptable amount of risk.

**Implementation and Deployment:** Implement the chosen algorithm and integrate it into your network monitoring system. The algorithm should run in real-time or periodic intervals to detect anomalies as they occur.

**Iterative Refinement:** Similar to other processes, you might need to fine-tune parameters, adjust the threshold, or even switch to a different algorithm based on real-world performance.

**Visualization and Reporting:** Visualize the detected anomalies in a way that makes them understandable to stakeholders. Provide clear reports indicating what

anomalies were detected, their severity, and recommended actions.

In summary, the selection and implementation of anomaly detection algorithms involve understanding data patterns, choosing suitable algorithms, tuning parameters, training models, setting thresholds, deploying the algorithm, refining the process iteratively, and presenting results. The application of mathematical principles and algorithmic techniques enables effective anomaly detection and contributes to network security and optimization.

### 3.8 Voice of the Customer Management Algorithm Implementation

Certainly, here's a comprehensive explanation of the steps involved in implementing Voice of the Customer (VoC) management algorithms, along with a description of some common methods:

**Understanding Voice of the Customer Management:** VoC management involves capturing, analyzing, and leveraging customer feedback to improve products and services. It's crucial for enhancing customer satisfaction and making informed business decisions.

**Data Collection and Preprocessing:** Gather customer feedback data from various sources like surveys, reviews, and social media.

1. Cleaning Special Characters, non english characters, Number, Date, time etc..
2. Cleaning unusual sentence "As per customer voice" , "as per Cx", "As per mail" etc.
3. Making of vocabulary dictionary of 400 words Tokenize convert raw string "Description" input into integer input.
4. Label encoding "AREA" to convert categorical variables into numerical form.
5. Converts the labels to a one-hot representation

**Algorithm Selection:** Select the best algorithms for assessing consumer feedback. Support Convolutional Neural Networks (CNN), Vector Machines (SVM), Logistic Regression (LR), Gaussian Naive Bayes (GNB) and Multinomial Naive Bayes (MNB) are popular options.

**Model Training:** Train the selected algorithm using labeled data. Labeled data consists of customer feedback samples categorized as positive, negative, or neutral sentiment.

**Equation Implementation:** Each algorithm has its equations that define its behavior. For example, the SVM equation entails determining the optimum hyperplane

to divide various classes while maximizing the margin.

**Algorithm Parameter Tuning:** Tune algorithm parameters to achieve optimal performance. Parameters might include kernel type in SVM, number of layers in CNN, or regularization strength in logistic regression.

**Model Evaluation:** Use measures like precision, recall, accuracy, F1-score and confusion matrices to assess the trained model's performance. These measurements demonstrate how well the model classifies user feedback.

**Sentiment Prediction:** Implement the trained model to predict sentiment (positive, negative, neutral) for new, unseen customer feedback.

**Integration and Automation:** Integrate the sentiment prediction process into your customer feedback pipeline. Automate the sentiment analysis step to process a large volume of feedback in real-time.

**Visualization and Reporting:** Visualize sentiment trends and customer feedback patterns over time. Create dashboards or reports that showcase valuable insights derived from sentiment analysis.

In summary, implementing VoC management algorithms involves collecting and pre-processing customer feedback, selecting appropriate sentiment analysis algorithms, training models, fine-tuning parameters, evaluating performance, predicting sentiment, integrating the process, visualizing insights, and iteratively refining the model. These algorithms, guided by mathematical principles, enable businesses to effectively leverage customer feedback for service enhancement and informed decision-making.

### 3.9 Architecture of the ML-Driven Performance Assurance and VoC Management System

The architecture comprises the following critical requirement or components:

#### 1. Data Collection Module:

- **Network Performance Data:** This part is constantly gathering information on the performance of the network from numerous sources, such as network devices, sensors, and monitoring tools. Data includes Key Performance Indicators (KPIs), latency, traffic, accessibility, throughput, packet loss, and more.
- **Historical Network Data:** Historical network data, containing past performance records, incidents, and resolutions, is retrieved from network archives and databases.
- **Voice of the Customer (VoC) Data:** VoC data is gathered from multiple sources, including customer feedback surveys, social media, and customer support interactions. It includes textual feedback, ratings, and comments.

## 2. Machine Learning Models and Algorithms:

- Anomaly Detection Algorithms: ML algorithms for anomaly detection are at the core of performance assurance. These models identify unusual patterns in network data and trigger alerts when anomalies are detected. Examples include Isolation Forest, One-Class SVM, and Autoencoders.
- Time Series Prediction Models: Time series prediction models forecast network performance metrics based on historical data. These models help in proactive resource allocation and traffic management. RNNs and LSTM networks are often used for this purpose.
- Natural Language Processing (NLP) Algorithms: NLP algorithms perform VoC management analysis and topic modeling on VoC data. They categorize customer feedback, identify common issues, and assess customer satisfaction.

## 3. Integration Layer:

- The integration layer serves as a middleware that connects data sources and ML models. It preprocesses and prepares data for analysis and model training.
- This layer facilitates the seamless integration of ML models into the network management system and VoC analysis module.

## 4. Network Management System (NMS):

- The NMS is responsible for network monitoring, configuration management, and performance optimization. It receives real-time performance insights from ML models and takes corrective actions based on predictions and feedback.
- The NMS is the operational hub of the system, where network administrators can view performance metrics, incidents, and recommendations.

## 5. VoC Management Module:

- This module manages VoC data, conducts sentiment analysis, and extracts valuable insights from customer feedback. It categorizes feedback into topics, identifies network-related issues, and assesses customer satisfaction.
- It provides a platform for network operators to understand the Voice of the Customer and prioritize network improvements based on customer feedback.

## 3.10 Data Flow and Processing Stages

The architecture follows a systematic data flow process to ensure effective performance assurance and VoC management:

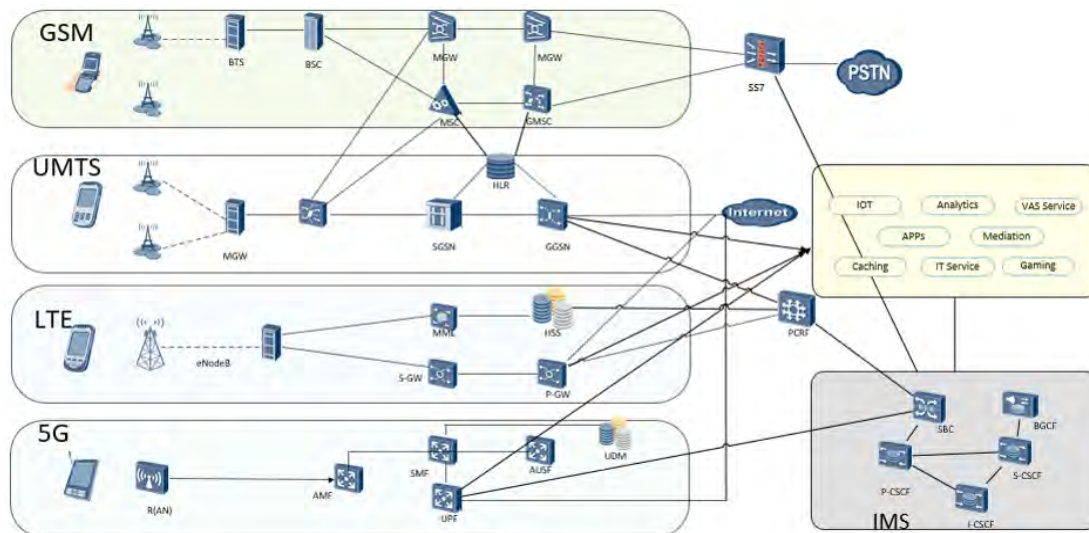


Figure 3.7: Telecom Convergence Core Network Architecture

### 1. Data Collection:

- Network performance data is continuously collected from network devices and sensors.
- Historical network data is retrieved from archives, and VoC data is gathered from various sources, including surveys and social media.

### 2. Data Preprocessing:

- Raw data undergoes cleaning and validation to remove inconsistencies and errors.
- Feature engineering is performed to extract relevant features from the data for improved ML model performance.
- VoC data is integrated with network performance data to correlate customer feedback with network events.

### 3. ML Model Integration

- Anomaly detection models process real-time performance data to identify anomalies and trigger alerts.
- Time series prediction models forecast network performance metrics, helping operators proactively allocate resources.
- NLP algorithms perform sentiment analysis and topic modeling on VoC data, categorizing feedback and identifying network-related issues.

### 4. Insights and Actions:

- The system provides real-time performance insights and alerts to the NMS. Operators can monitor network health and respond to anomalies.
- VoC insights, including sentiment analysis results and identified network issues, are presented on the dashboard. Operators can prioritize issue resolution based on customer feedback.

- A feedback loop is established, where actions taken by the NMS are analyzed for their impact on network performance and customer satisfaction. This feedback informs future network optimization strategies.

The architecture is designed to be scalable and adaptable to the evolving needs of highly convergent mobile operator networks. It accommodates the integration of new ML models and incorporates changes in data sources and network configurations. This section I have presented a comprehensive architecture for the ML-Driven Performance Assurance and VoC Management System. The architecture leverages Machine Learning techniques to enhance network performance, prioritize customer experience, and integrate real-time anomaly detection, time series prediction, and VoC sentiment analysis. The structured data flow and components ensure a holistic approach to network management and optimization.

### 3.11 Interaction between Performance Assurance and VoC Components

In the context of my research, it is essential to establish a robust interaction between the Performance Assurance and VoC components of the ML-Based system designed for highly convergent mobile operator networks. This interaction is pivotal as it bridges the technical aspects of network optimization with the customer-centric focus of VoC management.

**Performance Assurance Component:** The Performance Assurance component within the ML-Based system operates as the technical backbone, continuously monitoring network performance using Machine Learning algorithms. It actively collects real-time data on Key Performance Indicators (KPIs) like as throughput, latency, packet loss, and others. These metrics are crucial for assessing the network’s health and identifying anomalies or performance degradation promptly. For instance, ML models specialized in anomaly detection, such as Isolation Forest or One-Class SVM, analyze the KPI data to pinpoint deviations from expected patterns. When anomalies are detected, alerts are triggered, and corrective actions are initiated by the system, including reconfiguration or resource allocation adjustments. This technical vigilance ensures that network issues are identified and addressed promptly, contributing to improved performance.

**VoC Management Component:** On the other side of the spectrum, the VoC Management component focuses on capturing the Voice of the Customer. From a many channels, including surveys, social media, and contacts with customer service, it gathers client input. This feedback provides invaluable insights into user experiences, satisfaction levels, and concerns related to the network’s performance. The data is subjected to NLP techniques, including VoC analysis and topic modeling, to categorize and quantify customer sentiments and identify common network-related issues. The results of this analysis are then fed into the network management decision-making process. For example, if sentiment analysis reveals widespread customer dissatisfaction due to frequent network outages, the system can prioritize addressing these issues based on customer feedback, thus aligning network management strategies with customer expectations.

**Synergistic Interaction:** The synergy between the Performance Assurance and VoC components is where the true value of your research lies. Performance Assurance relies on data-driven insights to optimize network performance, and VoC Management taps into customer feedback to drive network improvements. This interaction operates as a feedback loop: Performance Assurance ensures that network issues are detected and resolved swiftly, while VoC Management ensures that customer experiences and feedback are integrated into performance optimization strategies. When network anomalies are detected and addressed by the Performance Assurance component, the VoC Management component can measure the subsequent impact on customer satisfaction and perception. This iterative process fosters a dynamic and responsive network management system, aligning the technical aspects of performance assurance with the customer-centric goals of enhancing the network's quality of service. The end result is a highly convergent mobile operator network that not only meets technical benchmarks but also delivers a superior customer experience, a critical factor in today's competitive telecommunications landscape.

In summary, the interaction between the Performance Assurance and VoC components in my ML-Based system for highly convergent mobile operator networks is a dynamic process that marries technical network optimization with customer-centric strategies. Through this connection, network concerns are swiftly resolved, and consumer input is crucial in determining how to maintain the network. The synergy between these components forms the backbone of a responsive and customer-focused network management system, ultimately enhancing both network performance and customer satisfaction.



# Chapter 4

## ML Based Performance Assurance

### 4.1 Time Series Forecasting Algorithms

Time series prediction plays a pivotal role in optimizing network performance, offering insights into future trends and aiding in proactive decision-making. In the view of network optimization, time series forecasting techniques such as XGBoost, ARIMA (AutoRegressive Integrated Moving Average), LSTM (Long Short-Term Memory), Prophet, Dynamic Linear Model, VAR (Vector AutoRegressive), and GRU (Gated Recurrent Unit) are employed to predict network behavior and facilitate strategic adjustments.

#### 4.1.1 ARIMA Model (AutoRegressive Integrated Moving Average)

Popular time series forecasting model ARIMA handles non-stationary data by combining autoregressive (AR), moving average (MA), and differencing integration (I) components. With univariate time series data that have temporal relationships, it functions well.

##### Equations:

The ARIMA model is explained as ARIMA(p, d, q)

- $p$  = Autoregressive component order.
- $d$  = Differencing (integration) degree.
- $q$  = Moving average component order.

The basic equation for an ARIMA(p, d, q) model is :

$$Y_t = c + \sum_{i=1}^p \phi_i Y_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t$$

Where:

- $Y_t$  = Observed value at time  $t$

- $c$  = A constant term
- $p$  = Autoregressive component order
- $\phi_i$  = Autoregressive coefficients
- $q$  = Moving average component order
- $\theta_j$  = Moving average coefficients
- $\varepsilon_t$  = White noise error term at time  $t$

### Model Components:

**i. AutoRegressive (AR) Component:** The AR component express the relationship between the current value and its previous values (lags). The autoregressive equation is  $Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \varepsilon_t$ .

**ii. Integrated (I) Component:** The integration component shows differencing the time series data to make it stationary. The differenced data is denoted as  $Y'_t = Y_t - Y_{t-1}$ .

**iii. Moving Average (MA) Component:** The MA component express the relationship between the current value and past, white noise error terms. The moving average equation is  $Y_t = \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$ .

### Steps for ARIMA Modeling:

**i. Identification:** To identify the order  $p$ ,  $d$ , and  $q$  by analyzing plots of autocorrelation and partial autocorrelation .

**i. Estimation:** T estimate the coefficients  $\phi_i$  and  $\theta_j$  using methods like Maximum Likelihood Estimation (MLE).

**i. Diagnostic Checking:** To check the model's residuals for randomness and stationarity.

**iv. Forecasting:** To use the trained ARIMA model to make future predictions.

### Advantages:

- ARIMA can model time series data patterns and seasonality whith a wide range of data.
- It provides interpretable coefficients that can offer insights into the time series behavior.

### Limitations:

- ARIMA assumes linear relationships, which might not hold for all data.
- The model's performance can degrade with noisy data or when underlying patterns change over time.

ARIMA is a powerful model for time series forecasting, polular in various fields such

as economics, finance, and engineering. Its combination of autoregressive, differencing, and moving average components allows it to capture and predict complex temporal dependencies in data.

### Stationary Check:

The Augmented Dickey-Fuller (ADF) test is a statistical hypothesis test employed to assess the stationarity of a given time series dataset. A stationary time series is characterized by unchanging statistical properties like mean, variance, and autocorrelation over time.

The p-value associated with the Augmented Dickey-Fuller (ADF) test is significantly low, with a value of 4.3134080073948825e-16, which indicates strong evidence in favor of stationarity.

## 4.1.2 XGBoost (eXtreme Gradient Boosting)

**XGBoost (eXtreme Gradient Boosting):** Popular ensemble learning technique XGBoost is well known for its ability to create a robust predictive model by accumulating the predictions of multiple weaker models. Furthermore, it has gained recognition for delivering impressive performance across wide range of machine learning tasks, such as classification and regression. The base equation for XGBoost can be defined as:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i)$$

Where:

- $\hat{y}_i$  = Predicted value for observation  $i$
- $K$  = Weak models (trees) Number
- $f_k(x_i)$  = Prediction made by the  $k$ -th tree for observation  $i$

### Equations:

The core equation for XGBoost involves optimizing a loss function by accumulating a series of weak learners (typically decision trees) to build a strong predictive model. Let's break down the equation step by step:

#### i. Objective Function:

XGBoost optimizes a regularized objective function  $\mathcal{L}$  that combines a loss term  $\mathcal{L}(y_i, \hat{y}_i)$  with a regularization term  $\Omega(f)$ :

$$\mathcal{L}(f) = \sum_{i=1}^n \mathcal{L}(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

## ii. Loss Term:

The loss term  $\mathcal{L}(y_i, \hat{y}_i)$  measures differences between the predicted value  $\hat{y}_i$  and the actual label  $y_i$ . For regression tasks, the squared error (MSE) is popular, while for classification tasks, the log loss (cross-entropy) is often employed.

## iii. Regularization Term:

The regularization term  $\Omega(f_k)$  discourages complex models that may overfit. It usually consists of L1 (Lasso) and L2 (Ridge) regularization terms to constrain feature weights and tree structures.

## iv. Additive Training:

XGBoost builds an ensemble model by adding weak learners (decision trees) to the ensemble. Each new tree attempts to rectify the errors of the previous trees.

## v. Prediction:

The final prediction  $\hat{y}_i$  for a data point  $x_i$  is obtained by summing the predictions from all individual trees in the ensemble:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i)$$

## Key Concepts:

**Gradient Boosting:** XGBoost employs gradient boosting, where each new tree is built to minimize the negative gradient of the loss function.

**Regularization Techniques:** XGBoost includes L1 and L2 regularization, as well as a "max-depth" hyperparameter to control the depth of individual trees.

**Feature Importance:** XGBoost delivers feature importance ratings, assisting in determining the model's most important features.

**Handling Missing Values:** XGBoost can manage missing values when building trees.

**Early Stopping:** XGBoost supports early stopping to prevent overfitting by monitoring a validation dataset's performance.

## Advantages:

- XGBoost demonstrates cutting-edge performance across a diverse array of tasks..
- It handles missing values and regularization effectively.

- Feature importance analysis provides insights into model behavior.

**Limitations:**

- XGBoost requires careful parameter tuning to prevent overfitting.
- It might be computationally expensive for very large datasets.

In summary, XGBoost is a powerful and versatile algorithm that delivers exceptional performance by employing gradient boosting with regularization techniques. Its flexible nature and ability to handle various types of data make it a popular choice in machine learning competitions and real-world applications.

### 4.1.3 LSTM (Long Short-Term Memory)

**LSTM:**

LSTM is a specialized form of recurrent neural network (RNN) used for the processing of sequential data. It has a unique architecture that introduces gates to control information flow over time. Unlike traditional RNNs, which struggle with the vanishing gradient problem, LSTM units are equipped with mechanisms to remember and forget information over extended sequences.

**Equations:**

The LSTM design uses a variety of gating techniques to regulate the information flow over time steps. Here are the key equations that govern the behavior of an LSTM unit:

**i. Input Gate:**

The decision regarding which data from the latest time step should be incorporated into the cell state is governed by the input gate. It has a sigmoid activation function that generates values in the range of 0 and 1, indicating which areas of the input should be updated.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

**ii. Forget Gate:**

The forget gate undermines what information from the previous cell state should be kept or discarded. It utilizes a sigmoid activation function, just as the input gate.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

**iii. Cell State Update:**

The cell state  $C_t$  is updated by combining the input gate output with the proposed new cell state  $\tilde{C}_t$ , which is computed using a tanh activation function.

$$\tilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

#### iv. Output Gate:

Components of the cell state should be conducted to produce the hidden state output are decided by the output gate. Both sigmoid and tanh activation functions are involved.

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

#### Key Components:

**Cell State:** The cell state  $C_t$  acts as the memory of the LSTM unit, allowing it to remember relevant information over long sequences.

**Gating Mechanisms:** Input gate, forget gate, and output gate facilitate the LSTM unit to control information flow, recall crucial information, and provide valuable outputs.

**Long-Term Dependencies:** Traditional RNNs have a vanishing gradient problem, but LSTMs solve it, enabling them to represent long-range relationships in sequential data.

**Backpropagation Through Time (BPTT):** Training LSTMs involves BPTT, a technique for propagating gradients through time to update the model's weights.

#### Advantages:

- LSTMs excel at capturing sequential patterns and long-term dependencies.
- They are effective in various tasks, including text generation, speech recognition, and time series forecasting.

#### Limitations:

- LSTMs may need careful parameter tweaking and can be computationally costly.
- For simpler tasks, they can be overkill and less interpretable than other models.

In summary, LSTMs are a fundamental advancement in neural network architectures, enabling the effective modeling of sequential data. They are a cornerstone in several disciplines, including time series analysis and natural language processing, due to their capacity to capture long-term dependencies.

#### 4.1.4 Dynamic Linear Model (DLM)

**Dynamic Linear Model (DLM):** A Dynamic Linear Model (DLM) is a probabilistic model used for forecasting and time series analysis. It is a flexible framework that captures the dynamics of time-varying processes through a combination of linear transformations and stochastic components. DLMs are particularly useful when dealing with time series data that exhibit changing patterns over time.

##### Equations:

The key equations of a DLM involve two main components: the observation equation and the state equation.

##### i. Observation Equation:

The observation equation represents how the observed data  $y_t$  at time  $t$  is related to the underlying state  $\theta_t$  and an observation noise  $\varepsilon_t$ .

$$y_t = F_t\theta_t + v_t$$

Where:

- $F_t$  = observation matrix relates the state to the observed data at time  $t$ .
- $\theta_t$  = state vector at time  $t$ .
- $v_t$  = observation of noise.

##### ii. State Equation:

The state equation models the evolution of the underlying state  $\theta_t$  over time  $t$  in terms of a state transition matrix  $G_t$ , a control vector  $u_t$ , and a state noise  $w_t$ .

$$\theta_{t+1} = G_t\theta_t + u_t + w_t$$

Where:

- $G_t$  is the state transition matrix that defines how the state evolves from  $t$  to  $t + 1$ .
- $u_t$  is a control vector that may represent exogenous inputs.
- $w_t$  is the state noise.

##### iii. Initial State Distribution:

The initial state distribution  $p(\theta_1)$  specifies the initial state at time  $t = 1$ . It is often assumed to follow a Gaussian distribution.

##### Components and Features:

**Time-Varying Dynamics:** DLMs allow for changing dynamics over time by adapting the state transition matrix  $G_t$  and observation matrix  $F_t$  at each time step.

**Latent States:** The underlying states  $\theta_t$  capture the unobservable dynamic patterns in the data. These states can represent trends, seasonality, or other latent features.

**Observation Noise and State Noise:** The observation noise  $v_t$  and state noise  $w_t$  account for uncertainties and random fluctuations in the observed data and the

state evolution.

**Forecasting and Filtering:** DLMS can be used for both forecasting future observations and filtering past observations to estimate the underlying states.

**Advantages:**

- DLMS provide a flexible framework to capture various time-varying patterns in data.
- They can handle missing data and accommodate exogenous inputs.

**Limitations:**

- DLMS require proper specification of state transition and observation matrices, which might be challenging in complex scenarios.
- Interpretation of DLM results might be less intuitive compared to simpler models.

In summary, Dynamic Linear Models offer a probabilistic approach to modeling and forecasting time series data, hence it highly able to adapt to changing dynamics and incorporate uncertainty makes suitable for a variety range of applications, such as financial modeling, macroeconomic forecasting, and sensor data analysis.

### 4.1.5 Prophet

**Prophet:** Prophet is a forecasting tool developed by Facebook that handles time series data with strong seasonal patterns. While the model doesn't have a single equation, it uses an additive model that accounts for seasonality, holidays, and trend components to make predictions.

**Equations:**

Prophet uses an additive model that combines several components to model time series data. While Prophet does not rely on explicit equations as some other models do, it is based on the following components:

**i. Trend Component:**

The trend component captures the underlying growth or decay patterns in the data. Prophet allows for flexible trend specification by using piecewise linear functions with changepoints, enabling the model to adapt to shifts in the data trend.

**ii. Seasonality Component:**

Prophet accommodates multiple seasonalities, both daily and yearly. Each seasonal component is modeled using Fourier series expansion, which approximates complex seasonal patterns with a sum of sinusoidal terms.

**iii. Holiday Effects:**

Prophet can include custom holiday effects by incorporating additional regressors into the model. It provides the flexibility to define holidays and their impact on the



time series.

#### **iv. Additional Regressors:**

In addition to holidays, Prophet allows for the inclusion of user-defined regressors that might have an impact on the time series.

**Key Features: Automatic Seasonality Detection:** Prophet automatically detects and models yearly and weekly seasonality patterns in the data.

**Holiday Effects:** You can specify holidays and their impact on the time series, accounting for significant variations during holiday periods.

**Changepoint Detection:** Prophet automatically detects changepoints in the data, enabling the model to adapt to shifts in trends.

**Forecast Uncertainty:** Prophet provides uncertainty intervals around the forecasts, helping to quantify the uncertainty in the predictions.

**Flexibility:** It can handle missing data and outliers gracefully, making it suitable for real-world and messy time series data.

#### **Advantages:**

- Prophet is easy to use and requires minimal tuning, making it accessible to both beginners and experts.
- It manages a variety of time series data characteristics, including seasonality, holidays, and trend variations.
- Prophet's automatic seasonality detection and changepoint detection simplify the modeling process.

#### **Limitations:**

- Prophet's model structure is designed for forecasting rather than modeling underlying dynamics.
- While it is versatile, it might not handle extremely complex or irregular data patterns as well as some other advanced models.

In summary, Prophet is a valuable tool for time series forecasting, particularly when dealing with data that exhibits multiple seasonalities, holidays, and changing trends. Its ability to handle various components of time series data with ease makes it a popular choice for both beginners and experienced data analysts.

### **4.1.6 VAR (Vector AutoRegressive) Model**

**VAR (Vector AutoRegressive) Model:** A multivariate time series model Vector AutoRegressive (VAR) model that captures the linear relationship between multiple variables over time. Unlike univariate autoregressive models, which consider only one

variable's past values, VAR models incorporate lagged values of multiple variables to predict their future values. The basic equation for a VAR( $p$ ) model is given by a system of equations:

$$Y_t = c + \sum_{i=1}^p A_i Y_{t-i} + \varepsilon_t$$

Where:

- $Y_t$  = vector of observed values at time  $t$
- $c$  = constant term
- $p$  = order of the model
- $A_i$  = coefficient matrices
- $\varepsilon_t$  = error vector at time  $t$

### Equations:

The VAR model involves a system of equations, each representing the evolution of one variable over time based on its own lagged values and other variables lagged values. The general equation for a VAR( $p$ ) model with  $K$  variables is:

$$y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \dots + \Phi_p y_{t-p} + \varepsilon_t$$

Where:

- $y_t$  =  $K$ -dimensional vector of variables at time  $t$ .
- $c$  = constant term.
- $\Phi_i$  are  $K \times K$  coefficient matrices for lag  $i$ .
- $y_{t-i}$  = the lagged values of the variables at time  $t - i$ .
- $p$  is the order of the autoregressive model.

In matrix notation, the equation can be written as:

$$Y_t = C + \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \dots + \Phi_p Y_{t-p} + \varepsilon_t$$

Where

$Y_t$  is the matrix of  $K$  variables at time  $t$ ,  $C$  is a constant matrix, and  $\varepsilon_t$  is the  $K$ -dimensional vector of error terms at time  $t$ .

### Model Order Selection:

Selecting the appropriate order  $p$  is crucial for the VAR model. This can be done using methods such as information criteria (AIC, BIC) or cross-validation.

### Granger Causality:

One of the key insights from VAR models is the concept of Granger causality. If the lagged values of one variable help predict another variable, the first variable is said to "Granger cause" the second.

### Advantages:

- VAR models capture the interactions and feedback between multiple variables,

making them suitable for analyzing interconnected data.

- They can be used for forecasting multiple variables simultaneously.
- VAR models provide insights into Granger causality relationships.

**Limitations:**

- VAR models assume linear relationships, which might not hold for all types of data.
- Interpretation of the coefficients can be challenging when dealing with many variables.

In summary, the VAR model is a valuable tool for analyzing multivariate time series data. By considering the relationships and interactions between multiple variables over time, VAR models offer insights into the dynamics of complex systems and can be used for forecasting and causal analysis.

### 4.1.7 GRU (Gated Recurrent Unit)

**GRU (Gated Recurrent Unit):** Vanishing gradient problem and the identification of persistent patterns in sequential data were the driving forces behind the development of the Gated Recurrent Unit (GRU), a particular kind of recurrent neural network (RNN). While providing a more simplified design than Long Short-Term Memory (LSTM) networks, GRU offers comparable performance.

**Equations:**

GRU introduces gating mechanisms to control information flow through the network, which helps it to remember and forget information over time. Here are the key equations of a GRU unit:

**i. Update Gate (z):**

The update gate  $z_t$  determines value of the previous hidden state  $h_{t-1}$  should be incorporated with the candidate state  $\tilde{h}_t$  to produce the new hidden state  $h_t$ .

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

**ii. Reset Gate (r):**

The reset gate  $r_t$  controls the previous hidden state  $h_{t-1}$  should be ignored or not when computing the candidate state  $\tilde{h}_t$ .

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

**iii. Candidate State ( $\tilde{h}_t$ ):**

Candidate state  $\tilde{h}_t$  is a temporary state that incorporates the current input  $x_t$  and the reset gate  $r_t$ .

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t])$$

**iii. Hidden State (h):**

New hidden state  $h_t$  is calculated by interpolating between the previous hidden state  $h_{t-1}$  and the candidate state  $\tilde{h}_t$  based on the update gate  $z_t$ .

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

### Key Features:

**Gating Mechanisms:** The update and reset gates allow GRUs to control information flow and prevent the vanishing gradient problem.

**Simplicity:** GRUs have a simpler architecture compared to LSTMs, which can make them easier to train and interpret.

**Efficiency:** GRUs typically have fewer parameters than LSTMs, which can lead to faster training times.

### Advantages:

- GRUs can capture long-range dependencies in sequential data without the risk of vanishing gradients.
- They perform well on tasks involving sequences with varying time dependencies.
- GRUs can be more memory-efficient compared to LSTMs due to their simplified architecture.

### Limitations:

- GRUs might struggle with very long sequences or tasks requiring precise modeling of complex temporal dependencies.
- Their performance can vary based on the nature of the data and the problem at hand.

In summary, GRUs offer an effective solution for capturing temporal dependencies in sequential data. Their gating mechanisms allow them to regulate information flow over time, making them a popular choice for various applications in natural language processing, speech recognition, and time series analysis.

## 4.2 Anomaly Detection Algorithms

### 4.2.1 DBSCAN (The Density-Based Spatial Clustering of Applications with Noise)

Here is a description of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) technique for clustering and recognizing noise in data, as well as its main characteristics and guiding concepts:

DBSCAN is a density-based clustering algorithm that is particularly effective at identifying clusters of arbitrary shapes in data. It defines clusters as areas of high data point density and is capable of detecting noise points as well. DBSCAN groups data points that are compacted in the same cluster and marks isolated points as

noise.

**Principles:**

DBSCAN operates based on two main parameters: **Epsilon ( $\varepsilon$ ):** A distance threshold that defines the radius around a data point, forming its neighborhood.

**MinPts:** The minimum requirement of data for dense region. Points within the  $\varepsilon$  neighborhood of a data point are considered part of that point's neighborhood.

**Equations:** The DBSCAN algorithm can be described through a few key concepts and equations:

**i. Epsilon Neighborhood:**

Given a data point  $p$ , the epsilon neighborhood  $N_\varepsilon(p)$  of  $p$  is defined as all data points within a distance  $\varepsilon$  of  $p$ :

$$N_\varepsilon(p) = \{q \mid \text{dist}(p, q) \leq \varepsilon\}$$

where  $\text{dist}(p, q)$  is the distance between data points  $p$  and  $q$ .

**ii. Directly Density-Reachable:**

A point  $p$  is said to be directly density based reachable from point  $q$  if  $p$  is within the epsilon neighborhood of  $q$  and  $q$  has at least MinPts points within its own epsilon neighborhood:

$$\text{dist}(p, q) \leq \varepsilon \quad \text{and} \quad |N_\varepsilon(q)| \geq \text{MinPts}$$

**iii. Density-Connected:**

Two points  $p$  and  $q$  are density-connected if there exists a data point  $o$  such that both  $p$  and  $q$  are directly density-reachable from  $o$ :

$$\exists o \mid p \text{ and } q \text{ are directly density-reachable from } o$$

**Key Features:**

**Cluster Discovery:** DBSCAN can discover clusters of arbitrary shapes, including those that might not be well separated.

**Noise Detection:** DBSCAN can identify isolated data points as noise, which helps in cleaning the data.

**Parameter-Free Clustering:** DBSCAN doesn't require specifying the number of clusters and is relatively robust to the choice of parameters.

**Scalability:** It's suitable for large datasets due to its density-based nature.

**Advantages:**

- DBSCAN can handle clusters of varying shapes and densities.
- It's well-suited for noisy data and can automatically detect and label outliers.
- DBSCAN doesn't assume a fixed number of clusters.

**Limitations:**

- It might struggle with clusters of significantly varying densities.
- Choosing suitable parameters ( $\epsilon$  and MinPts) can be challenging, and their values can impact the results.

In summary, DBSCAN is a valuable algorithm for clustering and outlier detection in data. Its ability to define clusters based on density allows it to identify complex cluster structures and handle noise effectively.

## 4.2.2 Isolation Forest

An anomaly detection algorithm that focuses on isolating anomalies rather than clustering normal data points. It operates by constructing a tree-based structure to separate anomalies from the other data. Anomalies are expected to be isolated more quickly during tree construction, making them stand out as shorter paths in the forest.

**Principles:**

Isolation Forest operates based on two main principles:

- Anomalies are Fewer:** Anomalies are expected to be fewer in number compared to normal data points in a dataset.
- Anomalies are Different:** Anomalies are different from normal data points and are easier to separate.

**Equations:** Isolation Forest operates without a traditional set of equations. Instead, it relies on the following concepts:

**i. Path Length:**

During the construction of an isolation tree, a data point's path length is the number of edges traversed from the root node to the leaf node where the point resides. In an ideal scenario, anomalies should require fewer splits to be isolated compared to normal data points.

**ii. Path Length and Anomalies:**

The intuition behind Isolation Forest is that anomalies will have shorter average path lengths within trees. Since anomalies are different from normal data points, they should be separated more quickly.

**Key Features:**

**Random Partitioning:** Isolation Forest uses random partitioning to build trees, leading to efficient and scalable anomaly detection.

**Fast Detection:** Anomalies are expected to have shorter average path lengths, enabling faster detection.

**Parameter-Free:** Isolation Forest does not require specifying the number of anoma-

lies or clusters in advance.

**Advantages:**

- Isolation Forest is efficient and scalable for detecting anomalies, making it suitable for large datasets.
- It works well even with high-dimensional data.
- The algorithm does not rely on assumptions about the distribution of the data.

**Limitations:**

- Isolation Forest might struggle with datasets where anomalies are of the same density as normal data points.
- The algorithm might require tuning of hyperparameters like the number of trees and the maximum depth.

In conclusion, Isolation Forest is a powerful and efficient anomaly detection algorithm that leverages the properties of anomalies to separate them from normal data points. Its ability to handle high-dimensional data and its parameter-free nature make it a popular choice for various applications in fraud detection, cybersecurity, and quality control.

### 4.2.3 Local Outlier Factor (LOF)

The LOF is an anomaly detection technique that finds data points with densities that are noticeably different from those of their nearby neighbors. LOF evaluates a data point's local density in relation to the densities of its neighbors. Anomalies are anticipated to have a lower density than their surrounding areas, resulting in greater LOF values.

**Principles:**

On the premise that anomalies are less densely surrounded by other data points, LOF operates. A data point's local density is calculated by comparing its distance to that of its k-nearest neighbors. Because anomalies often have lower densities than their surrounding areas, their LOF values are higher.

**Equations:**

The LOF algorithm involves a few key equations:

**i. Local Reachability Density (LRD):**

The local reachability density of a data point  $p$ , denoted as  $LRD_k(p)$ , measures how reachable  $p$  is from its neighbors within a certain radius  $k$ . It is computed as the inverse average of the reachability distances of  $p$  from its neighbors:

$$LRD_k(p) = \frac{1}{\text{avg}(N_k(p))}$$

where  $N_k(p)$  is the set of  $k$  nearest neighbors of  $p$ .

**ii. Local Outlier Factor (LOF):**

The LOF of a data point  $p$ , denoted as  $LOF_k(p)$ , quantifies the extent to which  $p$  differs in density from its neighbors. It is the average ratio of the LRD of  $p$  and the LRD of its neighbors:

$$LOF_k(p) = \frac{\text{avg}(LRD_k(N_k(p)))}{LRD_k(p)}$$

**Key Features:**

**Local Density Comparison:** LOF compares a data point's local density to that of its neighbors.

**Anomaly Ranking:** When compared to their neighbors, anomalies with lower density receive higher values from LOF.

**Parameter  $k$ :** The parameter  $k$  determines the number of nearest neighbors to consider when computing LRD and LOF.

**Advantages:**

- LOF can identify anomalies of varying shapes and densities.
- It can capture local anomalies that might not be clear in global analysis.
- The algorithm does not assume any specific distribution of data.

**Limitations:**

- LOF's performance might be sensitive to the choice of parameter  $k$ .
- The algorithm can be computationally expensive for large datasets.

In summary, By comparing the density of data points with their neighbors, the Local Outlier Factor (LOF) algorithm is a potent tool for spotting local abnormalities. It is excellent for a wide range of applications in anomaly detection, fraud detection, and quality control because to its flexibility in adapting to different forms and densities.

#### 4.2.4 One Class SVM (Support Vector Machine)

One-Class SVM is an anomaly detection technique that aims to locate a hyperplane that separates the vast majority of data points from a small subset of likely outliers. It is a binary classification variation of the classic Support Vector Machine.

**Principles:**

One-Class SVM operates on the assumption that usual data points are dense and tightly clustered, while anomalies are relatively far from the majority of data points.



It constructs a boundary (hyperplane) around the normal data points, aiming to include as many normal data points within the boundary as possible while excluding anomalies.

### Equations:

The One-Class SVM algorithm involves a few key equations:

#### i. Optimization Objective:

Finding the best hyperplane to maximize the margin around the training data is the objective of One-Class SVM. It aims to minimize the classification error while accommodating a predefined fraction of training data as outliers. The optimization problem can be represented as:

$$\min_{w, \xi, \rho} \frac{1}{2} \|w\|^2 + \frac{1}{\nu n} \sum_{i=1}^n \xi_i - \rho$$

subject to:

$$\begin{aligned} w \cdot \phi(x_i) &\geq \rho - \xi_i \\ \xi_i &\geq 0 \\ \sum_{i=1}^n \xi_i &\leq \nu n \end{aligned}$$

where:

- $w$  is the weight vector.
- $\xi$  represents slack variables for soft margin classification.
- $\rho$  is the offset of the hyperplane from the origin.
- $\phi(x_i)$  is the feature vector transformed by a kernel function.
- $n$  is training data points count.
- $\nu$  is a parameter that controls the trade-off between maximizing the margin and accommodating outliers.

#### ii. Decision Function:

The decision function of the One-Class SVM classifies a new data point  $x$  as an anomaly (outside the boundary) if  $f(x)$  is greater than a predefined threshold  $\rho$ :

$$f(x) = w \cdot \phi(x) - \rho$$

### Key Features:

**Binary Classification:** One-Class SVM performs binary classification where the normal class is labeled as inliers, and anomalies are considered outliers.

**Non-Linear Transformations:** One-Class SVM can handle non-linear relationships between features through kernel functions.

**Controllable Outlier Fraction ( $\nu$ ):** The parameter  $\nu$  allows you to control the expected fraction of training data to be considered as outliers.

**Advantages:**

- One-Class SVM can handle high-dimensional data and is effective for low-density anomaly detection.
- It works well in scenarios where normal data is well-clustered.

**Limitations:**

- One-Class SVM might require careful tuning of the kernel and parameter  $\nu$  to achieve optimal results.
- It assumes that anomalies are present in the training data, which might not always be the case.

In summary, One-Class SVM algorithm is a powerful tool for detecting anomalies by generating a boundary around normal data points. Its ability to handle non-linear transformations and control the outlier fraction makes it suitable for various applications in anomaly detection, fraud detection, and quality control.

## 4.2.5 Elliptic Envelope

The Elliptic Envelope is an anomaly detection algorithm that assumes the normal data points follow a Gaussian distribution and aims to identify anomalies that deviate significantly from this distribution. It models the inlying data as an elliptical envelope and detects points that fall outside of this envelope as anomalies.

**Principles:**

The Elliptic Envelope operates based on the assumption that normal data points follow a multivariate normal distribution. It estimates the mean and covariance of the data and constructs an ellipse that encompasses a certain proportion of the data points. Data points falling outside this ellipse are considered anomalies.

**Equations:**

The Elliptic Envelope algorithm involves a few key equations:

**i. Gaussian Distribution:**

The Elliptic Envelope assumes that the normal data points follow a multivariate Gaussian distribution having mean  $\mu$  and covariance  $\Sigma$  and the probability density function (PDF) of the Gaussian distribution is given by:

$$f(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$

## ii. Mahalanobis Distance:

To identify how far a data point  $x$  is from the mean  $\mu$  in terms of the covariance  $\Sigma$ , the Mahalanobis distance is used:

$$d(x, \mu) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$$

### Key Features:

**Gaussian Assumption:** The algorithm assumes that the normal data points follow a multivariate Gaussian distribution.

**Elliptical Envelope:** The algorithm models the inlying data as an elliptical envelope defined by the estimated mean and covariance.

**Anomaly Threshold:** Data points lying outside the envelope are considered anomalies based on a predefined threshold.

### Advantages:

- Elliptic Envelope can handle multivariate data and captures correlations between features.
- It is efficient and suitable for datasets with well-defined Gaussian-like clusters.
- The algorithm is relatively simple to understand and use.

### Limitations:

- Elliptic Envelope's performance might degrade if the data deviates significantly from the Gaussian distribution assumption.
- It might not be suitable for data with complex or non-Gaussian distributions.

In summary, the Elliptic Envelope algorithm is a valuable tool for detecting anomalies by assuming that normal data points follow a Gaussian distribution. Its simplicity and efficiency make it a good choice for situations where the underlying data distribution is approximately Gaussian-like.

## 4.2.6 Autoencoders

Autoencoders represent a neural network structure utilized for tasks like reconstruction-based anomaly detection, feature extraction, and unsupervised learning, particularly in the realm of anomaly detection. In this setup, an encoder network compact input data into a lower-dimensional data representation, while a decoder network aims to regenerate the original data from this reduced representation.

### Principles:

Autoencoders operate based on the principle of reconstructing input data using a compressed representation in the middle layer. In anomaly detection, the model is

trained on normal data, and anomalies are identified based on their reconstruction error.

### Equations:

Autoencoders involve a few key equations:

#### i. Encoder Function:

The encoder function  $h(x)$  maps input data  $x$  to a lower-dimensional latent space representation  $z$ :

$$z = h(x)$$

#### ii. Decoder Function:

The decoder function  $g(z)$  maps the latent representation  $z$  back to the original data space to reconstruct  $x'$ :

$$x' = g(z)$$

#### iii. Reconstruction Loss:

The reconstruction loss measures the dissimilarity between the input  $x$  and its reconstruction  $x'$ . It is often the mean squared error (MSE) between the two:

$$\text{MSE}(x, x') = \frac{1}{n} \sum_{i=1}^n (x_i - x'_i)^2$$

### Key Features:

**Feature Learning:** Autoencoders can automatically learn relevant features from the data, capturing important patterns and variations.

**Dimensionality Reduction:** The latent space representation obtained from the encoder serves as a compressed version of the input data, reducing dimensionality.

**Reconstruction Loss:** Anomalies can be detected by comparing the reconstruction error (difference between input and output) with a predefined threshold.

### Advantages:

- Autoencoders can handle complex and high-dimensional data.
- They are versatile and can be applied to various data types, such as images, sequences, and tabular data.
- Autoencoders can capture non-linear relationships in the data.

### Limitations:

- Autoencoders might overfit to the training data if not properly regularized.
- They might struggle with rare or unseen anomalies during training.

In summary, autoencoders are a powerful tool for unsupervised feature learning and reconstruction-based anomaly detection. By learning a compressed representation of data, they offer insights into normal data patterns and can be effective in identifying deviations from these patterns.

### 4.2.7 HBOS (Histogram-Based Outlier Score)

The HBOS is an anomaly detection algorithm that leverages histograms to estimate the distribution of normal data and identify anomalies based on their deviation from this distribution. HBOS assumes that anomalies are sparsely distributed and deviate significantly from the majority of data points.

#### Principles:

HBOS operates based on the principle that anomalies are rare and have distinct values that differ from those of normal data points. It constructs histograms for individual features and computes the outlier score by combining the histograms' values for each feature.

#### Equations:

The HBOS algorithm involves a few key equations:

##### i. Feature Histograms:

For each feature  $x_i$ , a histogram  $H_i$  is constructed with  $b$  bins, covering the range of values in that feature. The histogram  $H_i$  counts how many data points fall within each bin.

##### ii. Outlier Score:

The outlier score  $S(x)$  for a data point  $x$  is calculated by multiplying the normalized bin count for each feature:

$$S(x) = \frac{1}{n} \sum_{i=1}^d \frac{H_i(x_i)}{B_i}$$

where:

- $n$  = number of features.
- $d$  = number of dimensions.
- $H_i(x_i)$  = count of  $x_i$  = corresponding bin.
- $B_i$  = width of the bin.

#### Key Features:

**Histogram-Based:** HBOS constructs histograms for individual features to capture the distribution of normal data.

**Outlier Score:** Anomalies are identified by their high outlier scores, indicating they deviate significantly from the normal data distribution.

**Scalability:** HBOS is relatively lightweight and can be efficient for large datasets.

**Advantages:**

- HBOS is computationally efficient and suitable for datasets with high dimensionality.
- It can capture anomalies that have distinct values in one or more features.
- HBOS does not assume specific data distributions.

**Limitations:**

- HBOS might struggle to capture complex relationships between features.
- It might require careful tuning of parameters like the number of bins ( $b$ ).

In summary, the Histogram-Based Outlier Score (HBOS) algorithm is a simple yet effective tool for anomaly detection by capturing the distribution of normal data using histograms. Its ability to identify anomalies based on deviations from the normal data distribution makes it suitable for various applications in fraud detection, cybersecurity, and quality control.

# Chapter 5

## Voice of the Customer (VoC) Management

### 5.1 Voice of the Customer (VoC) Management Algorithms

#### 5.1.1 Support Vector Machine (SVM)

Powerful machine learning techniques like Support Vector Machine (SVM) are employed for both regression and classification problems. In order to maximize the gap between the classes, the ideal hyperplane that best separates data points from various classes is sought after.

#### Principles:

The foundation of SVM is the idea of locating a hyperplane that optimizes the margin between two classes of data points. Support vectors are the nearest data points to the hyperplane and are very important in determining where the hyperplane is.

#### Equations:

The SVM algorithm involves a few key equations:

##### i. Hyperplane Equation:

The equation of the hyperplane for a binary classification is following:

$$f(x) = w \cdot x + b$$

where  $w$  = weight vector perpendicular to the hyperplane,  $x$  = input feature vector, and  $b$  = bias term.

##### ii. Distance from Hyperplane:

The distance of a data point  $x$  from the hyperplane can be calculated using the formula:

$$\text{distance} = \frac{|f(x)|}{\|w\|}$$

### iii. Margin and Support Vectors:

The margin shows how far apart the hyperplane is from the nearest data points in each class, measured in angles. This separation is what SVM attempts to increase. The data points that are on the edge of the data distribution and have an impact on where it is located are referred to as support vectors.

### iv. Soft Margin Classification:

In real-world scenarios, it's often not possible to have a perfectly separable dataset. SVM handles this by allowing for some misclassification. The concept of a "soft margin" involves introducing slack variables  $\xi_i$  that penalize misclassified data points:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

where  $C$  is the regularization parameter which controls the trade-off between maximizing the margin and minimizing the misclassification.

### Key Features:

**Margin Maximization:** By locating the hyperplane that optimizes the margin between classes, SVM seeks to achieve improved generalization.

**Kernel Trick:** SVM can be extended to nonlinear classification by using kernel functions data into higher-dimensional space to map.

**Binary and Multi-Class Classification:** SVM is capable of handling binary and multi-class classification issues.

### Advantages:

- SVM works well in high-dimensional spaces and has the ability to manage intricate decision limits.
- It is robust against overfitting, especially when using a proper regularization parameter.

### Limitations:



- SVM might be sensitive to noisy data and outliers.
- Choosing the appropriate kernel and regularization parameter can be challenging.

In summary the Support Vector Machine (SVM) method, which seeks to identify a hyperplane that best divides classes while maximizing the margin, is a flexible and effective tool for classification tasks. It is a popular option for many machine learning applications since it can handle high-dimensional data and nonlinear relationships to the kernel trick.

## 5.1.2 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a highly developed deep learning architecture that has been especially designed to handle the complex processing of visual and sequential data, including areas like images and text. Applying CNNs to challenging tasks like object detection, image recognition, and the delicate field of natural language processing brings out their actual potential.

### Principles:

CNNs operate based on the principles of feature extraction and hierarchical representation. They use convolutional layers to automatically learn pattern features from dataset, allowing them to capture local patterns hierarchies in the input.

### Equations:

The CNN architecture involves a few key equations:

#### i. Convolution Operation:

Convolution includes moving a tiny filter (kernel) through the input data and computing the dot product between the filter and the overlapped area of the input. The convolution operation for a specified location is defined mathematically as:

$$(I * K)(x, y) = \sum_{i=1}^m \sum_{j=1}^n I(x+i, y+j) \cdot K(i, j)$$

where  $I$  is the input data and  $K$  is the filter kernel.

#### ii. Pooling Operation:

By combining layers, the spatial dimensions of the data are reduced while key characteristics are preserved. The most popular pooling procedure, known as max pooling, pulls the most value possible from a small area of the input.

**iii. Activation Function:** The CNN becomes non-linear because to activation functions. Commonly employed is the rectified linear unit (ReLU):

$$f(x) = \max(0, x)$$

### **Key Features:**

**Local Feature Learning:** CNNs automatically learn local features and hierarchies in the data through convolution and pooling operations.

**Hierarchical Representation:** Deeper layers capture increasingly abstract features and relationships in the data.

**Parameter Sharing:** CNNs share weights across different regions of the input, reducing the number of parameters and improving generalization.

### **Advantages:**

- CNNs excel in capturing spatial hierarchies in images and sequences.
- They are capable of managing vast and complex datasets, which qualifies them for jobs like object identification, picture recognition, and natural language processing.
- CNNs can learn meaningful features without manual feature engineering.

### **Limitations:**

- CNNs require a substantial amount of trained data for optimal performance.
- Designing the architecture and selecting hyperparameters can be challenging.

In summary, Convolutional Neural Networks (CNNs) are a cornerstone of modern deep learning, capable of automatically learn and extract complex features from visual and sequential data. Their ability to capture local patterns and hierarchical representations has led to breakthroughs in various fields, making them a strong tool for wide range of applications.

## **5.1.3 Gaussian Naive Bayes (GNB)**

A probabilistic classification approach called Gaussian Naive Bayes makes the assumption that each class's features are normally distributed. Although it is a "naive" assumption, GNB can be useful for a variety of classification problems.

### **Principles:**

The GNB algorithm relies on the Bayes theorem and assumes that each class's features have a Gaussian (normal) distribution. A data point is segmented to the class having the highest probability after the probability of each class being the data point's home is calculated.

### **Equations:**

The GNB algorithm involves a few key equations:

**i. Bayes' Theorem:**

The Bayes theorem connects the likelihood of a class given its features  $P(C_k|x)$  to the likelihood of its features given its class  $P(x|C_k)$ , its prior likelihood  $P(C_k)$ , and its evidence  $P(x)$ :

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)}$$

where  $C_k$  represents the class.

**ii. Gaussian Probability Density Function (PDF):**

In GNB, the Gaussian PDF is used to calculate the likelihood of a feature  $x_i$  belonging to a particular class  $C_k$ :

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_{k,i}^2}} \exp\left(-\frac{(x_i - \mu_{k,i})^2}{2\sigma_{k,i}^2}\right)$$

where  $\mu_{k,i}$  is the mean and  $\sigma_{k,i}$  is the standard deviation of feature  $x_i$  within class  $C_k$ .

**iii. Class Prior Probability:** The prior probability  $P(C_k)$  represents the probability of a data point belonging to class  $C_k$  without considering the features.

**iv. Evidence:** The evidence  $P(x)$  is the probability of observing the features  $x$  across all classes. It is typically calculated as the sum of the product of the likelihood and the class prior for each class.

**Key Features:**

Independence Assumption GNB assumes that features are independent within each class, which is a simplifying but "naive" assumption.

**Advantages:**

- GNB is computationally efficient and requires fewer parameters compared to some other classifiers.
- It can perform well when the independence assumption is approximately met.
- GNB is suitable for cases with continuous features following a Gaussian distribution.

**Limitations:**

- The independence assumption might not hold for all types of data.
- GNB might not perform well on highly correlated features.

In summary, the Gaussian Naive Bayes (GNB) algorithm is a straightforward but efficient probabilistic classification method that relies on the presumption that characteristics are normally distributed within each class. It is a useful tool for a variety of classification problems because of how well it performs and how well it can handle continuous data, especially when the features have Gaussian distributions and are not overly dependent.

### 5.1.4 Multinomial Naive Bayes (MNB)

Multinomial Naive Bayes is a probabilistic classification algorithm specifically designed for text and discrete data, where features represent the occurrence counts of words or terms in documents. It assumes that features follow a multinomial distribution.

#### Principles:

MNB operates based on Bayes' theorem and the assumption that features are conditionally independent given the class. In the context of text classification, features correspond to the counts of different terms in a document.

#### Equations:

The MNB algorithm involves a few key equations:

##### i. Bayes' Theorem:

Similar to other Naive Bayes variations, MNB applies the Bayes theorem. It connects the likelihood of a class given a set of features  $P(C_k|x)$  to the likelihood of those features given a class  $P(x|C_k)$ , the prior likelihood of the class  $P(C_k)$ , and the likelihood of the evidence  $P(x)$ :

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)}$$

where  $C_k$  represents the class.

##### ii. Multinomial Probability:

In MNB, the multinomial probability is used to calculate the likelihood of observing a set of counts  $x$  for different terms in a document, given a particular class  $C_k$ :

$$P(x|C_k) = \frac{(\sum_{i=1}^n x_i)!}{x_1! \cdot x_2! \cdot \dots \cdot x_n!} \prod_{i=1}^n P(w_i|C_k)^{x_i}$$

where  $n$  is the number of unique terms (features),  $x_i$  is the count of term  $w_i$ , and  $P(w_i|C_k)$  is the probability of term  $w_i$  given class  $C_k$ .

**iii. Class Prior Probability:** The prior probability  $P(C_k)$  represents the probability of a data point belonging to class  $C_k$  without considering the features.

**iv. Evidence:** The evidence  $P(x)$  is the probability of observing the counts  $x$  across all classes. It is typically calculated as the sum of the product of the likelihood and the class prior for each class.

**Key Features:**

**Discrete Data Handling:** MNB is suitable for text and discrete data, where features represent occurrence counts.

**Feature Counts:** MNB focuses on the counts of features (terms) in documents.

**Probabilistic Classification:** MNB calculates the probability of a data point belonging to each data class and assigns data it to the class with the highest probability.

**Advantages:**

- MNB is particularly useful for text classification tasks such as NLP, sentimental analysis, spam detection, and topic categorization.
- It is computationally efficient and can handle high-dimensional and sparse data.

**Limitations:**

- MNB's performance can be affected by the "bag of words" assumption, where the order of terms is ignored.
- It might struggle with rare or unseen terms during training.

A specific approach for text and discrete data classification, Multinomial Naive Bayes (MNB), where the features correspond to word occurrence counts, is described in this way. It is a popular option for many natural language processing tasks due to its capacity to handle high-dimensional text input and computational efficiency.

### 5.1.5 Logistic Regression (LR)

In statistical problems requiring binary categorization, logistic regression is applied. In contrast to what its name suggests, it is a classification algorithm. A given input data point's likelihood of falling into a specific class is modeled by LR.

**Principles:**

Any input is converted to a value between 0 and 1 by the logistic function, which underlies how LR operates. The likelihood of the input falling into the positive class can be calculated using this value. To optimize the probability of detecting the specified class labels, the algorithm learns the weights and biases that match the data the best.

## Equations:

The Logistic Regression algorithm involves a few key equations:

### i. Logistic Function (Sigmoid):

The logistic function, is often known as the sigmoid function, maps the linear combination of input features  $x$  and their corresponding weights  $w$  to a value between 0 and 1:

$$f(x) = \frac{1}{1 + e^{-(w \cdot x + b)}}$$

where  $b$  is the bias term.

### ii. Log-Odds (Logit):

The log-odds, also known as the logit, is the logarithm of the odds that a data point belongs to the positive class:

$$\text{logit}(x) = \log\left(\frac{f(x)}{1 - f(x)}\right) = w \cdot x + b$$

### iii. Binary Cross-Entropy Loss:

The binary cross-entropy loss calculate the difference between the predicted probabilities and the true labels for binary classification:

$$\text{Loss}(y, \hat{y}) = -[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})]$$

where  $y$  is the true label (0 or 1) and  $\hat{y}$  is the predicted probability.

## Key Features:

**Probabilistic Interpretation:** LR models the probability that an input belongs to the positive class.

**Linear Decision Boundary:** The decision boundary of LR is linear in the feature space.

**Regularization:** LR can be regularized to prevent overfitting problem by adding regularization terms to the loss function.

## Advantages:

- LR is simple to understand and implement.
- It is well-suited for problems with a linear decision boundary.

- LR provides probabilistic outputs, which can be useful for understanding model confidence.

**Limitations:**

- LR might not perform well when the relationship between features and outcome is complex.

- It can struggle with datasets that have a significant class imbalance.

In summary, Logistic Regression (LR) is a fundamental classification algorithm which models the probability of an input belonging to a certain class. Its simplicity and interpretability make it a useful choice for various binary classification tasks, especially when the relationship between features and outcome is relatively linear.

# Chapter 6

## Implementation and Results

### 6.1 Time Series Forecasting Results

ARIMA (AutoRegressive Integrated Moving Average):

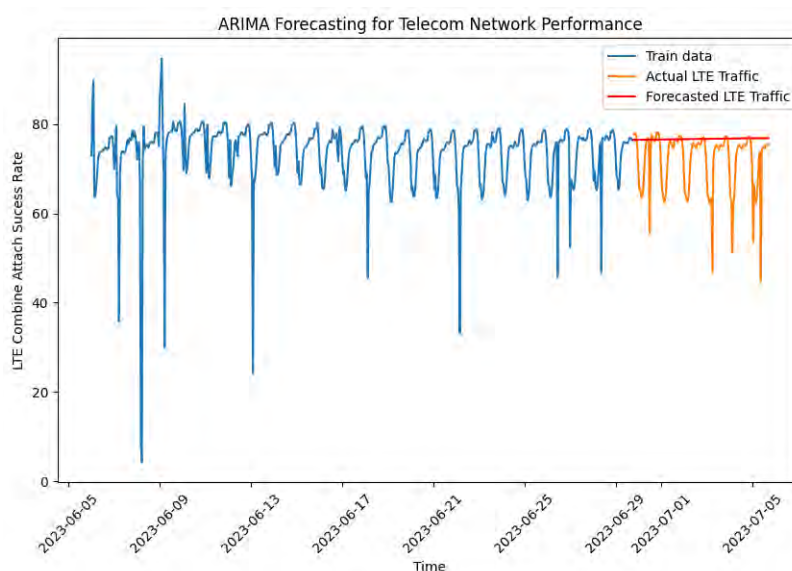


Figure 6.1: ARIMA

The provided figure offers a comprehensive view of the ARIMA (AutoRegressive Integrated Moving Average) model's performance in forecasting LTE traffic. The Y-axis, representing LTE attach success rate, unfolds its fluctuations over time, spanning from June 5, 2023, to July 5, 2023, on the X-axis. Within this graph, three crucial elements emerge:

Firstly, the "Train Data" segment showcases historical LTE attach success rate values, which serve as the foundational data used for training the ARIMA model. It's from this historical context that the model learns patterns and trends.

Secondly, the "Actual LTE attach success rate" curve signifies the observed LTE attach success rate values during the specified timeframe. This continuous line functions as the ground truth, enabling a direct comparison with the model's predictions.



Lastly, the "Forecasted Data" line, often visually distinguished, represents the ARIMA model's predictions for LTE attach success rate during the same period. These predictions are generated based on the model's understanding of past data.

The ARIMA model's order, denoted as (1, 2, 1), reveals its constituent components: autoregressive (AR), differencing (I), and moving average (MA). These components collectively enable the model to consider previous LTE attach success rate values, make the data stationary through differencing, and factor in past error terms for forecasting.

Regarding model performance, key metrics are reported:

The "Mean Absolute Error" (MAE) at approximately 4.73 quantifies the average absolute difference between forecasted and actual LTE attach success rate, with lower values indicating higher accuracy.

The "Mean Squared Error" (MSE) around 46.88 computes the average squared difference between forecasts and actual data, penalizing larger errors more.

The square root of MSE is the "Root Mean Squared Error" (RMSE), which, at around 6.85, provides an error measure in the same units as the data.

These metrics collectively evaluate the ARIMA model's precision in LTE attach success rate prediction, with lower error values signifying more precise forecasts, ultimately aiding in optimizing network operations and enhancing user experiences.

### XGBoost (eXtreme Gradient Boosting):

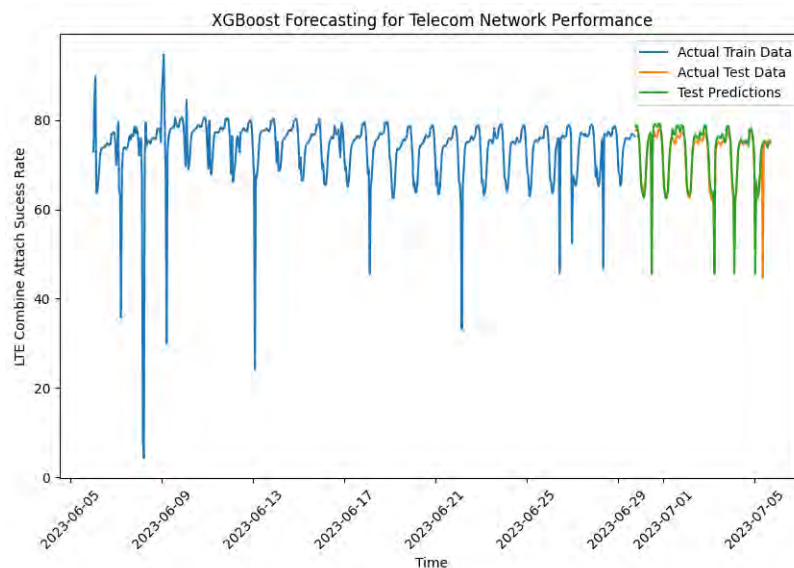


Figure 6.2: XGBoost

The figure depicting the XGBoost Regression Model offers a clear view of its performance in forecasting LTE traffic. The Y-axis signifies LTE attach success rate, spanning from June 5, 2023, to July 5, 2023, on the X-axis. Several crucial components emerge from this graph:

**Actual Train Data:** This section showcases historical LTE attach success rate data used to train the XGBoost model, providing the foundation for learning patterns within the data.

**Actual Test Data:** Displayed on the graph, this segment represents observed LTE attach success rate during the test period, serving as a benchmark for evaluating the model's predictions.

**Test Predictions:** The graph illustrates the model's predicted LTE attach success rate values during the test period, based on its training.

In addition to visual insights, the XGBoost model's configuration parameters are listed, including the boosting algorithm (GBTree) and various hyperparameters like learning rate (0.3), minimum split loss (0), maximum depth (6), minimum child weight (1), regularization lambda (1), and regularization alpha (0). Furthermore, performance metrics provide quantifiable evaluation: the Mean Squared Error (MSE) at approximately 41.276, the Root Mean Squared Error (RMSE) around 6.424, and the Mean Absolute Error (MAE) at roughly 4.222. These metrics collectively gauge the model's ability to provide accurate predictions, with lower values indicating superior forecasting accuracy, which in turn contributes to optimized network management and improved user experiences.

### LSTM (Long Short-Term Memory):

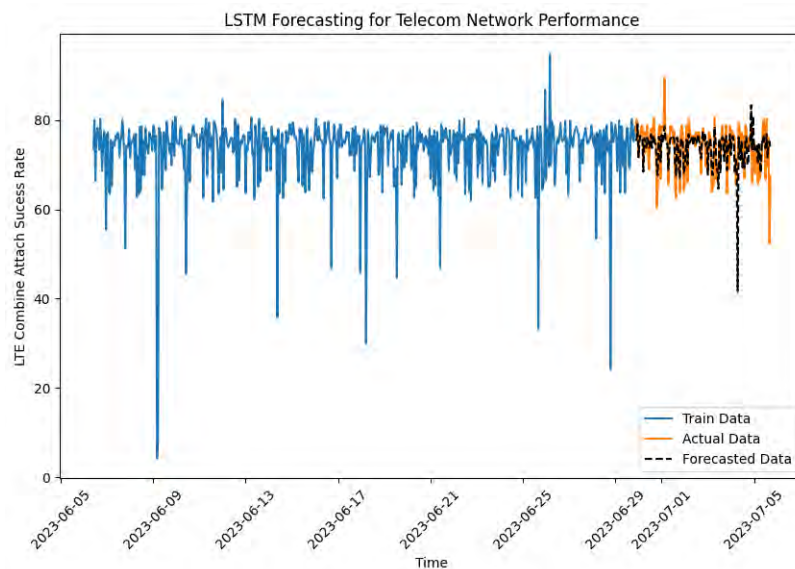


Figure 6.3: Long Short-Term Memory

The visual representation of the LSTM model provides a clear depiction of its performance in forecasting LTE attach success rate. The Y-axis represents LTE attach success rate, spanning from June 5, 2023, to July 5, 2023, on the X-axis. The graph comprises three critical elements:

**Train Data:** The historical LTE attach success rate data used to train the LSTM model is shown in this section of the graph, forming the basis for learning patterns within the dataset.

**Actual Data:** Represented by a continuous line on the graph, this segment signifies the observed LTE attach success rate values during the specified period, serving as a reference for evaluating the model's predictions.

**Forecasted Data:** The graph illustrates the model's predicted LTE attach success rate values for the same timeframe. These predictions are generated based on the LSTM model's training.

Additionally, details about the model's architecture and configuration are provided, including the total number of parameters (10,451), which are all trainable. The learning rate is set at 0.001. The model's error metrics include a Mean Absolute Error (MAE) of approximately 2.165, Mean Squared Error (MSE) of about 17.809, and Root Mean Squared Error (RMSE) at roughly 4.220. Furthermore, the LSTM model is characterized by specific parameter values, such as the activation function ('relu'), optimizer ('adam'), and loss function ('mean squared error'). These parameters define how the model processes data, updates its internal parameters, and measures the difference between values of predicted and actual. The LSTM model excels in forecasting LTE traffic, as evident from the graph's alignment between actual and forecasted data. With a low MAE, MSE, and RMSE, the model's predictions closely match reality, making it a valuable tool for optimizing network management. The model's architecture, hyperparameters, and error metrics collectively emphasize its effectiveness in this regard.

### Dynamic Linear Model (DLM):

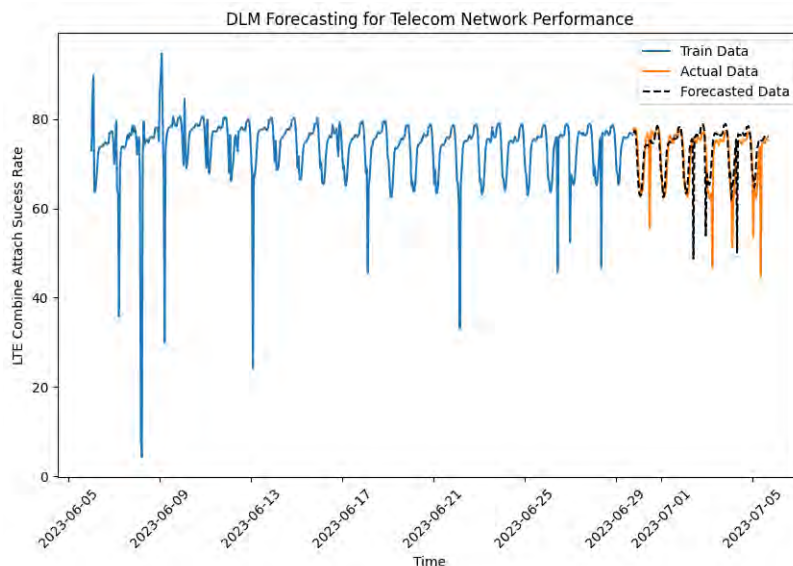


Figure 6.4: Dynamic Linear Models

The figure depicting the DLM provides a compelling glimpse into its capacity for predicting LTE attach success rate patterns. Along the Y-axis, LTE traffic volume

unfolds, while the X-axis marks the timeframe spanning from June 5, 2023, to July 5, 2023. Within this graph, three pivotal components emerge:

**Train Data:** This section encapsulates historical LTE attach success rate data, a cornerstone for the DLM's training process. During this phase, the model meticulously discerns intricate patterns and relationships within the dataset, laying the groundwork for accurate predictions.

**Actual Data:** Evident as a continuous line on the graph, this segment signifies the actual LTE attach success rate values observed during the specified timeframe. It serves as a benchmark, facilitating a direct comparison between the model's predictions and real-world observations.

**Forecasted Data:** The graph vividly illustrates the DLM's predictions for LTE attach success rate values over the same period. These forecasts arise from the model's comprehensive understanding of past data, along with its awareness of weekly seasonality trends and dynamic features.

The configuration of the DLM is noteworthy, featuring parameters such as degree (indicating a linear trend), weekly seasonality patterns, dynamic features extracted from a specified feature list, a discount factor of 0.9 applied to dynamic features, no auto-regression incorporated, and no explicitly specified long-term seasonality. Impressively, the DLM exhibits a high level of precision, as evidenced by its low error metrics: a Mean Absolute Error (MAE) of approximately 2.161, a Mean Squared Error (MSE) measuring around 7.511, and a Root Mean Squared Error (RMSE) of roughly 2.740. These error values underscore the DLM's capacity for accurately predicting LTE attach success rate patterns, positioning it as a valuable asset for optimizing network management and refining LTE attach success rate forecasting methodologies.

### **Prophet:**

The figure illustrating the performance of the Prophet model in forecasting LTE attach success rate provides a comprehensive view of its capabilities. The Y-axis portrays LTE attach success rate, covering the timeframe from June 5, 2023, to July 5, 2023, on the X-axis. Within this graph, three crucial elements stand out:

**Train Data:** This segment of the graph represents historical LTE attach success rate data used for training the Prophet model. It serves as the bedrock upon which the model learns intricate patterns and trends embedded within the dataset.

**Actual Data:** Evident as a continuous line on the graph, this part signifies the real, observed LTE attach success rate values during the specified period. It functions as a reference point, allowing us to gauge how closely the model's predictions coherence with the actual data.

**Forecasted Data:** The graph beautifully showcases the model's predictions for LTE attach success rate values across the same timeframe. These forecasts are the result of the Prophet model's extensive training and understanding of the dataset's under-

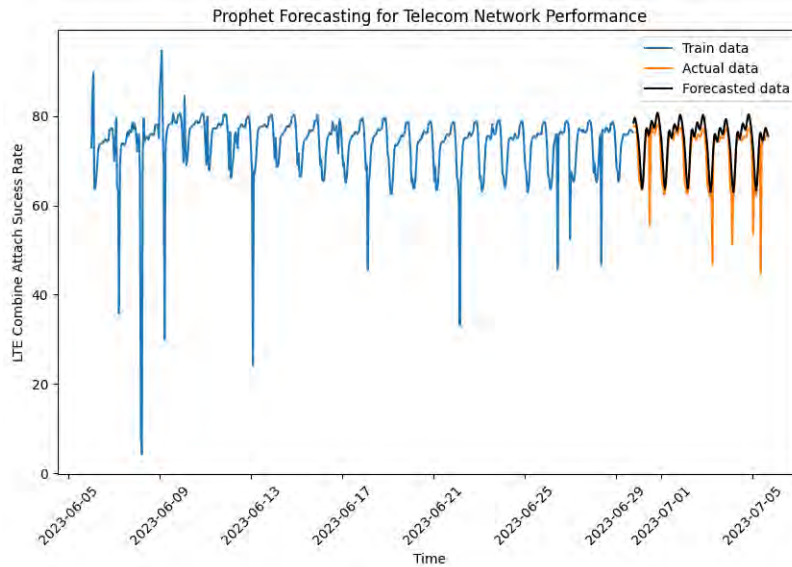


Figure 6.5: Prophet

lying patterns.

In addition to the data and forecasts, the Prophet model's configuration parameters are notable. These parameters include settings for growth (linear or logistic), changepoints (indicating shifts in data behavior), seasonality patterns (yearly, weekly, daily), and prior scales for various model components. These parameters collectively determine how the model interprets and predicts LLTE attach success rate. As for the model's performance, it's highly commendable. The error metrics reported are quite favorable:

Mean Absolute Error (MAE): The absolute difference between projected and actual LTE attach success rate numbers is quantified as 2.7503, on average.

Mean Squared Error (MSE): 23.9009, which is the average squared difference between the predicted and actual data.

Root Mean Squared Error (RMSE): The square root of MSE, or around 4.888, offers an error measure in the same units as the data.

These low error values highlight the Prophet model's exceptional accuracy in forecasting LTE attach success rate. Its ability to capture complex patterns and adjust to seasonality makes it a valuable tool for optimizing network management and enhancing decision-making in LTE attach success rate predictions.

### **VAR (Vector AutoRegressive) Model:**

The figure portraying the VAR (Vector AutoRegressive) Model offers a window into its predictive capabilities. Along the Y-axis, it visualizes the values of interest, while the X-axis marks the time, spanning from June 5, 2023, to July 5, 2023. Within this graph, three critical elements emerge:

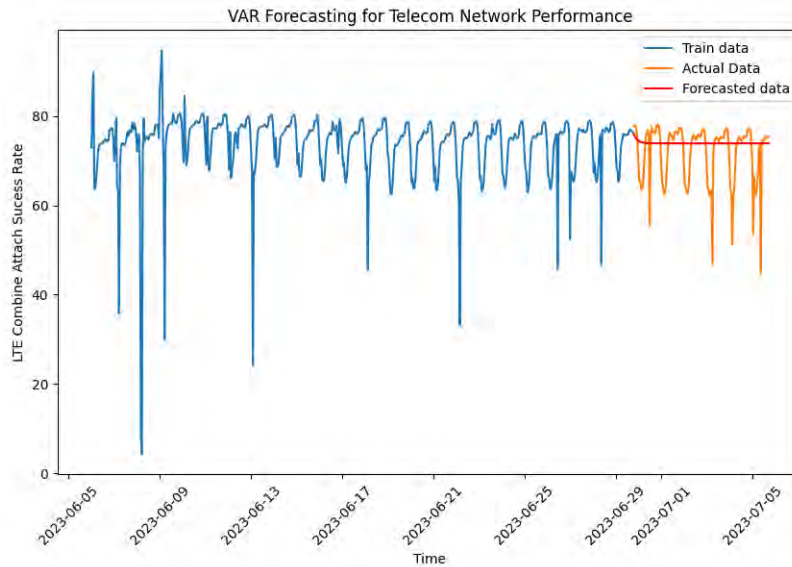


Figure 6.6: Vector Autoregression

**Train Data:** This section encapsulates historical data, a foundational component used to train the VAR model. During this training phase, the model diligently learns the relationships and dependencies among the variables within the dataset, critical for making accurate predictions.

**Actual Data:** Evident as a continuous line on the graph, this segment represents the actual values observed during the specified time frame. It serves as the benchmark against which the model's predictions are compared, allowing us to assess its accuracy.

**Forecasted Data:** The graph vividly illustrates the VAR model's predictions for the values over the same period. These projections are made using the model's knowledge of historical data and taking into account the lagged values of the endogenous variables, as determined by the number of lags parameter.

The VAR model's configuration is notable, with parameters including the number of lags (10) and the number of endogenous variables (2). The model's performance metrics indicate its predictive accuracy, with a MAE of approximately 4.4569, a Mean Squared Error (MSE) of around 45.7436, and a Root Mean Squared Error (RMSE) measuring roughly 6.7634. These error values underscore the VAR model's capacity to provide accurate predictions, making it a valuable tool for analyzing and forecasting the relationships between variables over time, crucial in various domains, including economics, finance, and engineering.

### **GRU (Gated Recurrent Unit):**

The figure depicting the GRU (Gated Recurrent Unit) model presents valuable insights into its performance in predicting LTE attach success rate. On the Y-axis, it showcases LTE attach success rate volume, while the X-axis represents the time period from June 5, 2023, to July 5, 2023. Within this graph, three pivotal components come into view:

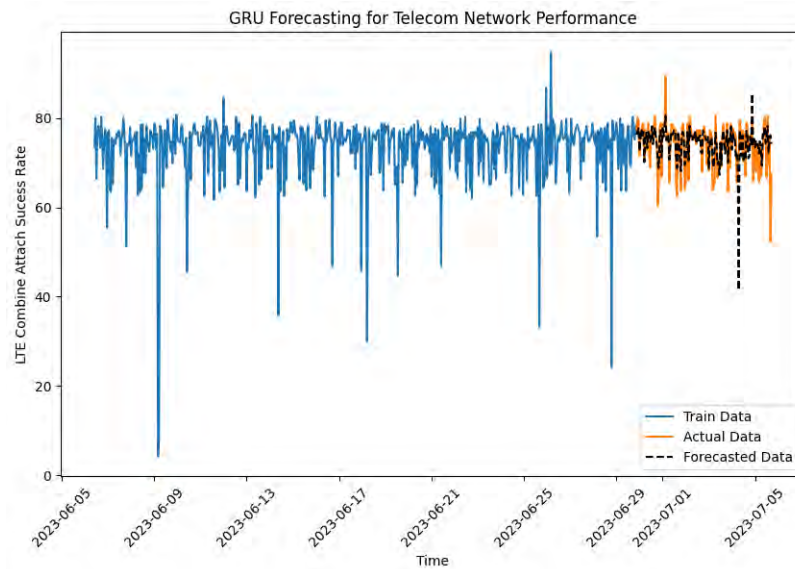


Figure 6.7: Gated Recurrent Unit (GRU) Networks

**Train Data:** This segment encapsulates historical LTE attach success rate data, which forms the bedrock for training the GRU model. Throughout this training phase, the model meticulously learns intricate temporal patterns and dependencies within the dataset, crucial for making accurate predictions.

**Actual Data:** Manifested as a continuous line on the graph, this portion represents the actual LTE attach success rate values observed during the specified timeframe. It serves as a reference point for evaluating the model's predictions against real-world observations.

**Forecasted Data:** The graph vividly illustrates the GRU model's predictions for LTE attach success rate values during the same period. These forecasts arise from the model's grasp of past data, taking into account sequential dependencies and patterns.

The GRU model's configuration encompasses key parameters, including a total of 8,001 trainable parameters, highlighting its complexity. The learning rate parameter is set at 0.001, governing the pace of weight adjustments during training. Additionally, activation functions like 'relu,' the 'adam' optimizer, and a loss function of 'mean squared error' contribute to the model's architecture and training methodology. It's noteworthy that the error metrics indicate strong predictive accuracy, with a MAE of approximately 2.6097, a Mean Squared Error (MSE) of about 20.8413, and a Root Mean Squared Error (RMSE) measuring roughly 4.5652. These error values underscore the GRU model's proficiency in forecasting LTE traffic, positioning it as a robust tool for optimizing network management and refining LTE attach success rate predictions.

## 6.2 Anomaly Detection Results

DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

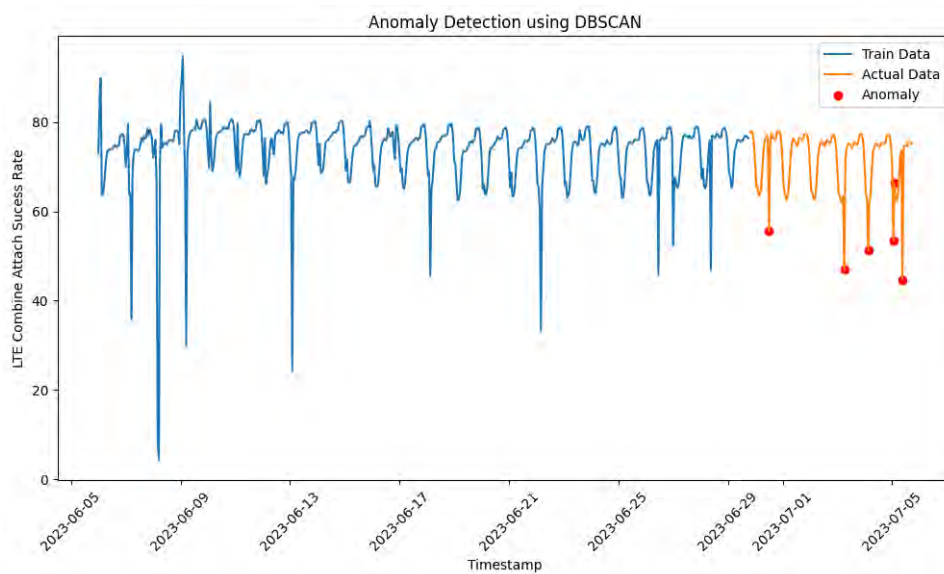


Figure 6.8: Density-Based Spatial Clustering of Applications with Noise

The DBSCAN figure offers a compelling visualization of its anomaly detection capabilities within the context of LTE attach success rate data. On the Y-axis, it represents LTE attach success rate volume, while the X-axis tracks the timeline from June 5, 2023, to July 5, 2023. This graphical representation encompasses several critical facets. This component of the dataset, which makes up about 80% of the total, contains historical LTE attach success rate data necessary for developing the DBSCAN model. The model painstakingly assimilates the complex patterns and subtleties connected with typical network behavior during the training phase. Evident as a continuous line on the graph for actual data, it depicts the actual LTE traffic numbers seen throughout the selected time period. It acts as the standard against which the model's predictions are measured, enabling an assessment of the model's accuracy. Red highlights clearly show anomalies in the LTE traffic statistics, especially during the test data period from June 29, 2023 to July 5, 2023. 20% of the dataset is still represented by these test data. The DBSCAN model exhibits impressive proficiency in spotting outliers, making it a powerful tool for enhancing network security and quickly spotting unusual network performance patterns. These abilities are essential for preserving the dependability and effectiveness of telecommunications networks.



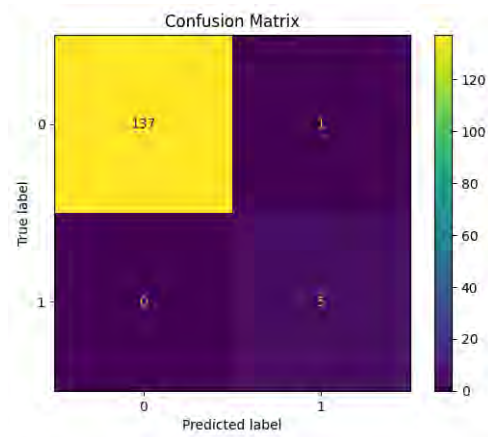


Figure 6.9: Confusion Matrix of Density-Based Spatial Clustering of Applications with Noise

The confusion matrix figure of DBSCAN presents a clear representation of the classification model’s performance. It distinguishes between true labels, encompassing both true positives (TP) and true negatives (TN), and predicted labels, which include positive (1) and negative (0) predictions. In this scenario, the matrix reveals that there are 137 true negatives (TN), indicating instances correctly classified as negative. Additionally, there are 5 true positives (TP), signifying correctly classified positive instances. Only 1 false positive (FP) is noted, which indicates instances incorrectly classified as positive, and there are no instances of false negatives (FN), signifying the absence of instances incorrectly classified as negative. This concise breakdown underscores the model’s overall accuracy and its ability to effectively distinguish between the two classes.

Classification Report:				
	precision	recall	f1-score	support
0	1.00	0.99	1.00	138
1	0.83	1.00	0.91	5
accuracy			0.99	143
macro avg	0.92	1.00	0.95	143
weighted avg	0.99	0.99	0.99	143

Figure 6.10: Classification Report of Density-Based Spatial Clustering of Applications with Noise

The Classification Report for DBSCAN indicates exceptionally high performance. The accuracy of 99.3% signifies that the majority of data points were correctly classified. Precision, at 83.3%, demonstrates that when DBSCAN identifies a data point as an outlier, it is accurate 83.3% of the time. A recall score of 100% highlights DBSCAN’s ability to capture all true outliers, leaving no outliers undetected. The F1-score, which harmonizes precision and recall, stands at a robust 90.9%, reinforcing the algorithm’s effectiveness in correctly identifying and classifying outliers. Overall, DBSCAN showcases outstanding performance in anomaly detection, making it a reliable choice for identifying rare and unusual data points in various applications.

## Isolation Forest:

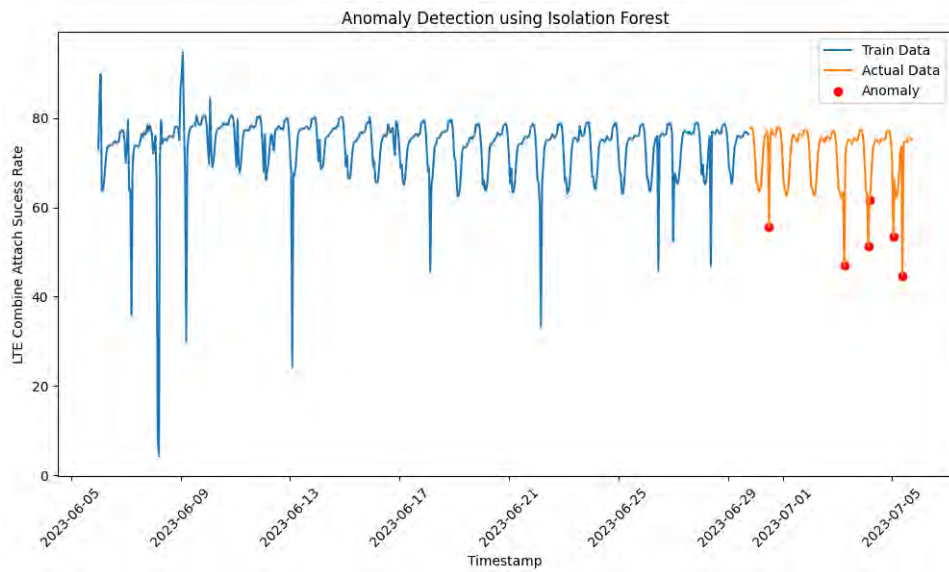


Figure 6.11: Isolation Forest

In the Isolation Forest figure, the Y-axis represents LTE traffic, while the X-axis spans from June 5, 2023, to July 6, 2023. Within this visual representation, three key elements are discernible: Accounting for 80% of the dataset as a train data, this segment comprises historical LTE traffic data used for training the Isolation Forest model. During this phase, the model acquires an understanding of typical patterns associated with normal network behavior. For actual data, Evident as a continuous line on the graph, this component represents the actual LTE attach success rate values observed throughout the specified timeframe, serving as a reference for evaluating the model's predictions. Anomalies within the LTE attach success rate data are distinctly highlighted in red, particularly during the test data period from June 30, 2023, to July 6, 2023. This test data, constituting the remaining 20% of the dataset, is precisely where the Isolation Forest model excels. It effectively identifies deviations from the learned normal patterns, making it a potent tool for bolstering network security and promptly pinpointing unusual network performance patterns. This capability is paramount for ensuring the reliability and integrity of telecommunications networks.

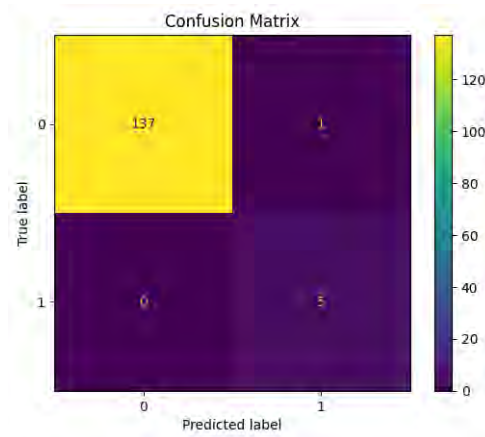


Figure 6.12: Confusion Matrix of Isolation Forest

The classification performance of the model is briefly summarized in the confusion matrix for the Isolation Forest. The actual labels, which include both true positives (TP) and true negatives (TN), and the predicted labels, which include both positive (1) and negative (0) predictions, are separated into two groups. The matrix in this situation shows that there are 137 true negatives (TN), or instances that were appropriately labeled as negative. There are also 5 true positives (TP), which denote cases that were accurately identified as positive. There is just one false positive (FP) recorded, signifying cases that were mistakenly classified as positive, and no false negatives (FN), indicating that there were no instances that were mistakenly classed as negative. This distinct division highlights the model’s strong performance in correctly identifying the two classes.

Classification Report:				
	precision	recall	f1-score	support
0	1.00	0.99	1.00	138
1	0.83	1.00	0.91	5
accuracy			0.99	143
macro avg	0.92	1.00	0.95	143
weighted avg	0.99	0.99	0.99	143

Figure 6.13: Classification Report of Isolation Forest

The Isolation Forest algorithm’s remarkable anomaly detection abilities are highlighted in the Classification Report. It shows a high degree of accuracy in categorizing data items as either normal or anomalous, with an accuracy score of 99.3%. With an accuracy of 83.3%, the Isolation Forest correctly classifies a data point as an outlier 83.3% of the time. A recall score of 100% demonstrates the Isolation Forest’s capacity to identify all genuine outliers and discover all other anomalies. A robust 90.9% for the F1-score, which maintains precision and recall, confirms the algorithm’s efficacy in correctly finding and categorizing anomalies. Overall, the Isolation Forest is a useful tool for finding rare and out-of-the-ordinary data items, making anomaly detection jobs possible.

## Local Outlier Factor (LOF):

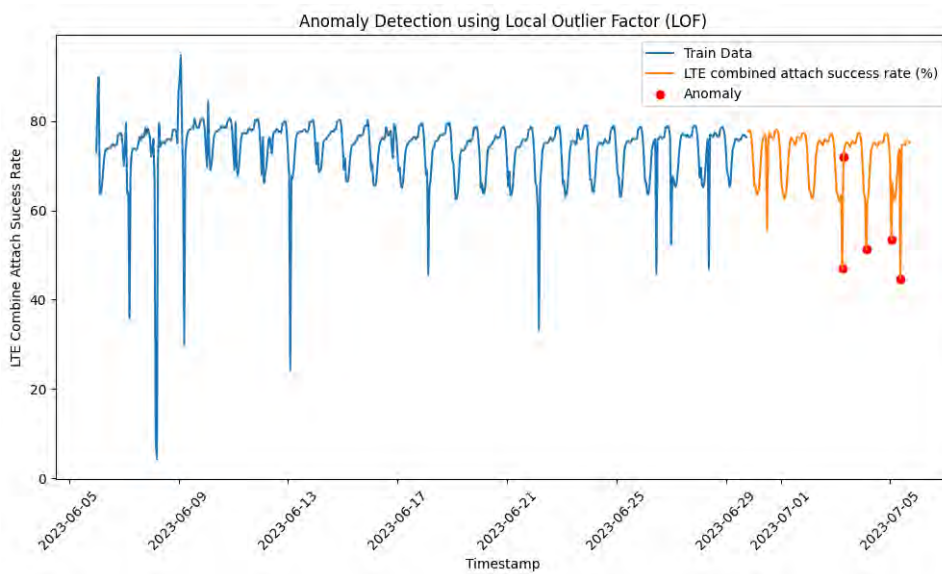


Figure 6.14: Local Outlier Factor

The LOF plot gives a visual representation of its extraordinary ability to find anomalies in the LTE attach success rate dataset. The X-axis shows the time period from June 5, 2023, to July 5, 2023, and the Y-axis shows the volume of LTE attach success rate. This graphical representation includes a number of important elements. This component of the dataset, which makes up a sizeable 80% of the train data, contains historical LTE attach success rate data that is essential for developing the LOF model. The model gains a full understanding of the complex patterns and behaviors connected to typical network operation during this training phase. For actual data, it matches the actual LTE traffic numbers observed over the predetermined duration and is represented as a continuous line on the graph. It serves as an average for measuring the model's forecasts, making it easier to judge the accuracy of the model. Particularly during the test data period from June 29, 2023 to July 5, 2023, anomalies within the LTE attach success rate statistics are noticeably highlighted in red. The remaining 20% of the dataset consists of these test data. The LOF model exhibits exceptional skill in detecting departures from the predicted norm, making it a crucial tool for enhancing network security and quickly recognizing unusual network performance trends. For telecommunications networks to remain dependable and successful, these characteristics are essential.

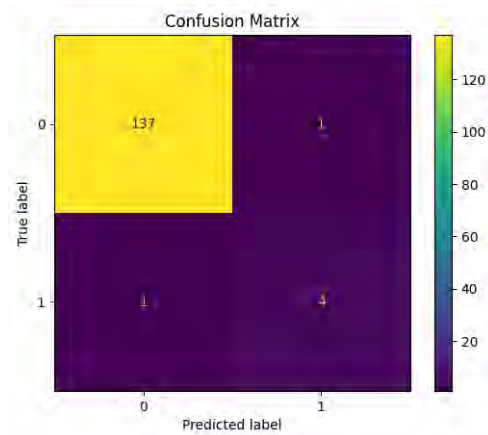


Figure 6.15: Confusion Matrix of Local Outlier Factor

LOF confusion matrix gives a precise assessment of the model’s classification performance. It distinguishes between two important labels: true labels (which include true positives (TP) and true negatives (TN)) and projected labels (which include positive (1) and negative (0) predictions). In this case, the matrix shows that there are 137 true negatives (TN), which denote instances that were accurately classified as negative. It also detects 4 ”true positives” (TP), or cases that were appropriately labeled as positive. However, 1 false positive (FP) denotes cases that were mistakenly classified as positive, and 1 false negative (FN) denotes instances that were mistakenly classified as negative.

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	138
1	0.80	0.80	0.80	5
accuracy			0.99	143
macro avg	0.90	0.90	0.90	143
weighted avg	0.99	0.99	0.99	143

Figure 6.16: Classification Report of Local Outlier Factor

The LOF algorithm’s Classification Report emphasizes its effective performance in anomaly detection. With a 98.6% accuracy rating, LOF shows a high degree of overall correctness in identifying data values as either normal or anomalous. According to the precision score of 80%, LOF correctly classifies a data point as an outlier 80% of the time. Similar to this, LOF’s recall score of 80% illustrates its capacity to successfully identify a sizeable proportion of real outliers. The F1-score, which balances precision and recall, is 0.80, demonstrating the efficiency of LOF in correctly finding and categorizing anomalies. With its balanced precision and recall metrics, LOF is a dependable option for identifying anomalous data points in a variety of applications.

## One Class SVM (Support Vector Machine):

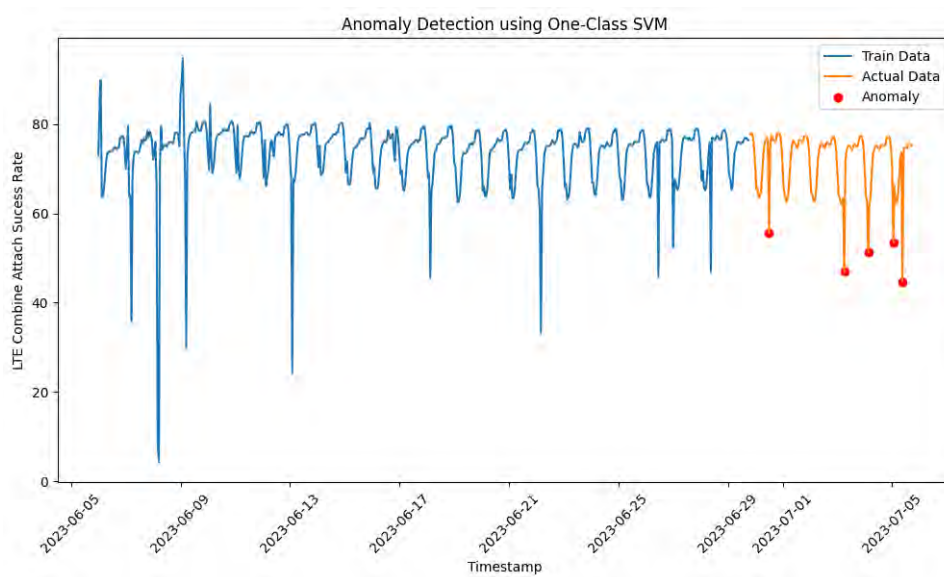


Figure 6.17: One Class SVM

The figure depicting the One Class SVM (Support Vector Machine) model provides a concise view of its capabilities in anomaly detection within LTE attach success rate. Along the Y-axis, it visualizes LTE attach success rate volume, while the X-axis spans from June 5, 2023, to July 5, 2023. This graph illustrates three crucial elements. Representing 80% of the dataset as train data, this segment comprises historical LTE attach success rate data utilized for training the One Class SVM model. During this phase, the model learns the patterns inherent in normal network behavior. For actual values, Evident as a continuous line on the graph, it signifies the actual LTE traffic values observed throughout the specified timeframe, serving as a reference point for assessing the model's predictions. The anomalies, depicted in red, are anomalies in the LTE attach success rate detected by the One Class SVM model. These anomalies are highlighted in accordance with the test data, which comprises the remaining 20% of the dataset. The model is designed to flag any deviations from the learned normal behavior as anomalies, making it a valuable tool for network security and fault detection.

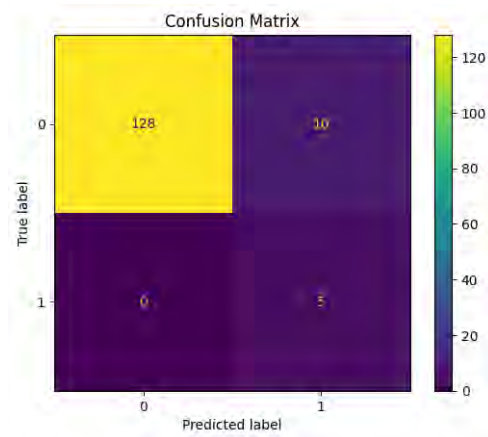


Figure 6.18: Confusion Matrix of One Class SVM

A brief summary of the model’s classification accuracy is provided by the confusion matrix for the One Class SVM (Support Vector Machine). It divides the results into two categories that are extremely important: true labels, which include true positives (TP) and true negatives (TN), and predicted labels, which include positive (1) and negative (0) forecasts. The matrix in this case shows that there are 128 true negatives (TN), which stand for cases that were appropriately identified as negative. Additionally, it pinpoints 5 “true positives” (TP), which are cases that were appropriately classified as positive. While there are 10 false positives (FP), or cases that were mistakenly classed as positive, it is noteworthy that there are no false negatives (FN), or absence of instances that were mistakenly labeled as negative.

Classification Report:				
	precision	recall	f1-score	support
0	1.00	0.93	0.96	138
1	0.33	1.00	0.50	5
accuracy			0.93	143
macro avg	0.67	0.96	0.73	143
weighted avg	0.98	0.93	0.95	143

Figure 6.19: Classification Report of One Class SVM

According to the Classification Report for the One-Class SVM Support Vector Machine, it correctly classifies typical data points with an accuracy of 93.0%. Its precision, however, is 33.3%, which means that it is only 33% accurate in spotting outliers. The model, on the plus side, earns a recall score of 100%, which means that it successfully captures all genuine outliers. The F1-score, which measures accuracy and precision while balancing recall and accuracy, is currently 0.5. In summary, the One-Class SVM is ideal for applications that prioritize recall over precision because it is excellent at recognizing outliers but may also produce some false alarms.

## Elliptic Envelope:

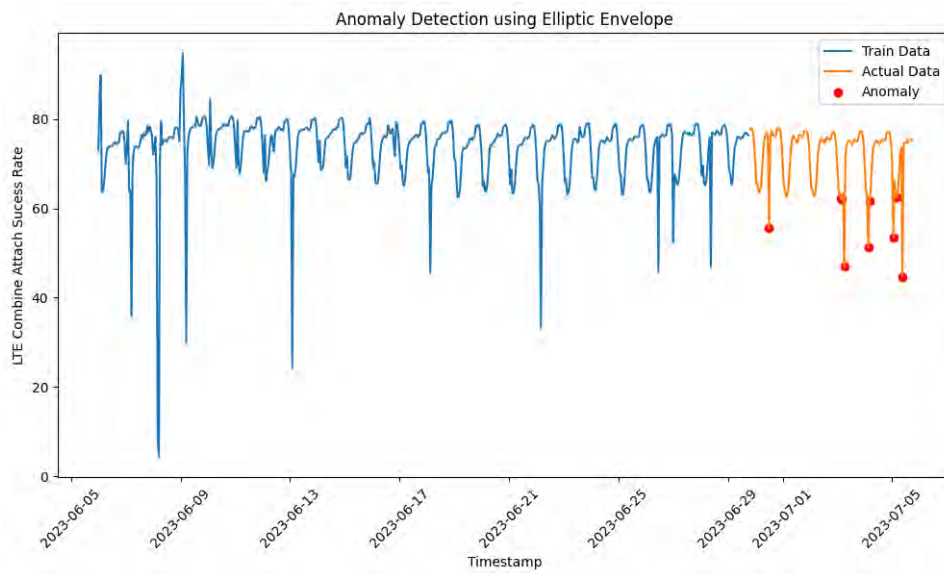


Figure 6.20: Elliptic Envelope

In the LTE traffic dataset, the Elliptic Envelope’s competent anomaly detection capabilities are shown in the figure. The Y-axis shows the amount of LTE attach success rate, and the X-axis shows the time period from June 5 to July 5 in 2023. This visual depiction includes a number of crucial components. This portion of the dataset, which accounts for an important 80% of the train data, contains historical LTE traffic data that is necessary for developing the Elliptic Envelope model. The numerous patterns and traits connected with regular network behavior are painstakingly captured by the model throughout the training process. It replicates the actual LTE attach success rate levels observed over the designated duration and appears as a continuous line on the graph for actual data. The model’s predictions are measured to this real data, allowing the accuracy of the model to be determined. Red is prominently used to identify anomalies in the LTE attach success rate statistics, especially during the test data period from June 29, 2023 to July 5, 2023. The remaining 20% of the dataset is represented by these test data. The ability of the Elliptic Envelope model to spot deviations from the norm makes it a useful tool for boosting network security and quickly spotting unexpected network performance trends. In order to guarantee the durability and effectiveness of telecommunications networks, such capabilities are essential.



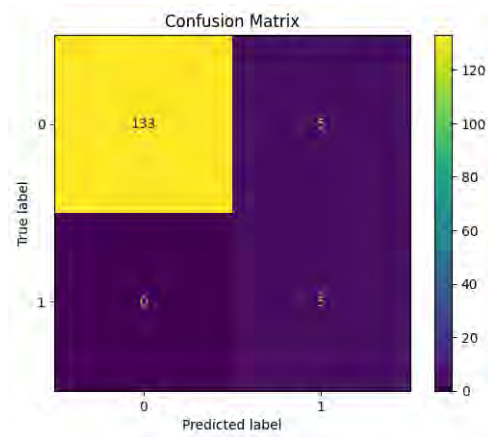


Figure 6.21: Confusion Matrix of Elliptic Envelope

The Elliptic Envelope’s confusion matrix offers an informative overview of the model’s classification performance. It clearly divides the outcomes into two crucial labels: true labels, which include true positives (TP) and true negatives (TN), and projected labels, which include positive (1) and negative (0) forecasts. The matrix in this situation reveals that there are 133 true negatives (TN), or instances that were appropriately labeled as negative. It also detects 5 true positives (TP), which are cases that were appropriately labeled as positive. However, there are 5 false positives (FP), or events that were mistakenly labeled as positive. Notably, there are no cases of false negatives (FN), indicating that there were no instances that were mistakenly labeled as negative.

Classification Report:				
	precision	recall	f1-score	support
0	1.00	0.96	0.98	138
1	0.50	1.00	0.67	5
accuracy			0.97	143
macro avg	0.75	0.98	0.82	143
weighted avg	0.98	0.97	0.97	143

Figure 6.22: Classification Report of Elliptic Envelope

The Elliptic Envelope model’s classification report shows that it can distinguish between normal and anomalous data points with an accuracy of 96.5%. However, it only has a 50% precision, meaning it is only 50% accurate when spotting anomalies. The model correctly catches all real anomalies, as evidenced by its flawless recall score of 100%. A acceptable trade-off between accuracy and precision is indicated by the F1-score, which measures how well precision and recall are balanced at 0.67. In conclusion, the Elliptic Envelope model is well suited for situations where accuracy is critical because it excels at finding abnormalities with few false positives.

## Autoencoders:

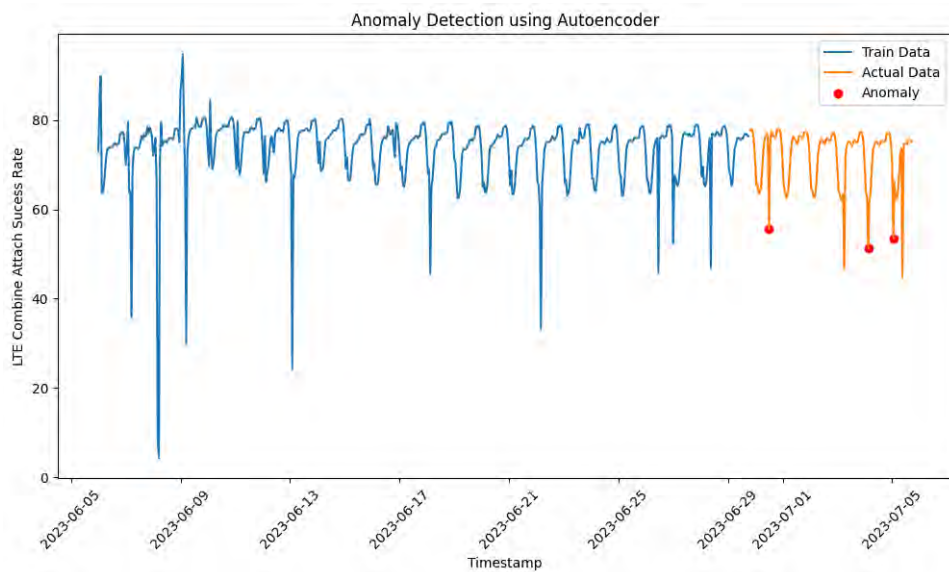


Figure 6.23: Autoencoders

The Autoencoders figure offers a comprehensive view of its anomaly detection prowess within the context of LTE attach success rate data. On the Y-axis, it represents LTE attach success rate, while the X-axis covers the time span from June 5, 2023, to July 5, 2023. This visualization encapsulates several critical aspects. A train's data This part, which makes up a sizeable 80% of the dataset, contains the historical LTE attach success rate data necessary for autoencoding model training. The model carefully assimilates the complex patterns and traits connected with regular network behavior throughout this training phase. It displays the actual LTE attach success rate figures seen during the selected timeframe as a continuous line on the graph. It acts as the standard against which the model's predictions are measured, making it easier to assess the accuracy of the model. Specifically within the test data period from June 29, 2023 to July 5, 2023, anomalies in the LTE traffic statistics are starkly highlighted in red. The remaining 20% of the dataset consists of these test data. The Autoencoders model is a powerful tool for boosting network security and quickly finding unusual network performance patterns since it is excellent at spotting deviations from the norm. To protect the dependability and effectiveness of telecommunications networks, these characteristics are essential.

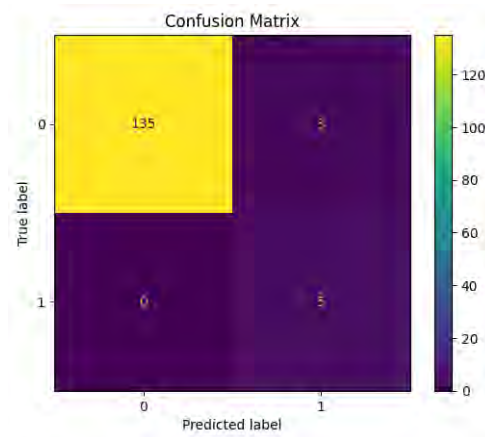


Figure 6.24: Confusion Matrix of Autoencoders

The model’s classification accuracy can be quickly assessed using the confusion matrix for the Autoencoders. The results are effectively divided into two basic labels: true labels, which include true positives (TP) and true negatives (TN), and predicted labels, which include positive (1) and negative (0) forecasts. There are 135 true negatives (TN) in this particular case, which refers to cases that were appropriately classified as negative. Additionally, it reveals 5 true positives (TP), which denote cases that were accurately identified as positive. The number of false positives (FP), or cases that were mistakenly classified as positive, is just three. In contrast, there are no false negatives (FN), which suggests that there were no instances that were mistakenly classified as negative.

Classification Report:				
	precision	recall	f1-score	support
0	1.00	0.98	0.99	138
1	0.62	1.00	0.77	5
accuracy			0.98	143
macro avg	0.81	0.99	0.88	143
weighted avg	0.99	0.98	0.98	143

Figure 6.25: Classification Report of Autoencoders

The Autoencoders model’s Classification Report shows a 97.9% accuracy rate, showing high overall performance in differentiating between normal and anomalous data items. The model’s accuracy in detecting anomalies is shown by its precision score of 62.5%, which shows that it is generally reliable. Additionally, the model achieves a recall score of 100%, which is the highest possible value and indicates that it can accurately detect all real abnormalities. A stable trade-off between precision and recall may be seen in the F1-score, which is 0.77%. In conclusion, the Autoencoders model is very good at detecting anomalies because it combines precision with a potent capacity to catch real anomalies.

## HBOS (Histogram-Based Outlier Score):

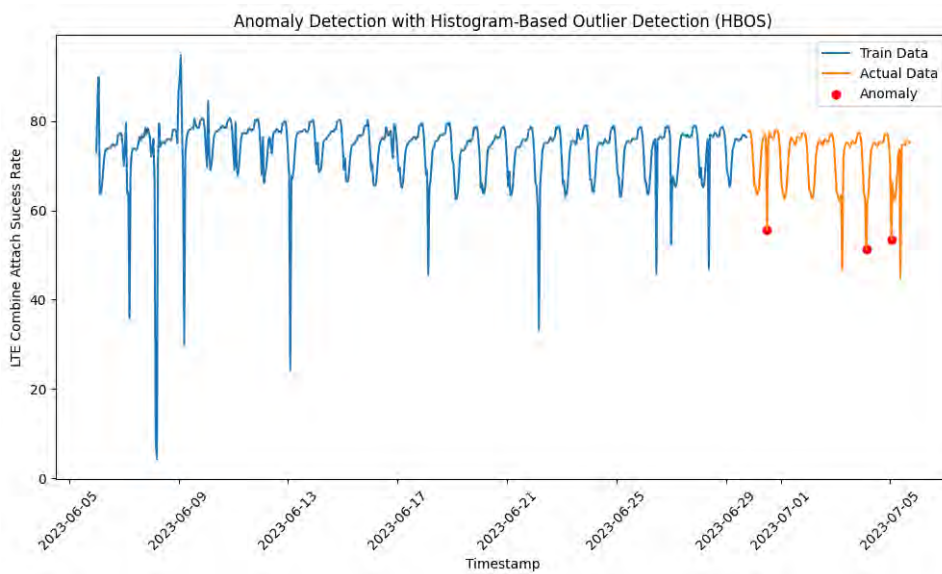


Figure 6.26: Histogram-Based Outlier Score

The HBOS is an efficient tool for detecting anomalies in LTE traffic statistics, as seen in the figure. The X-axis shows the time period from June 5, 2023, to July 5, 2023, and the Y-axis shows the amount of LTE traffic. Several important components are included in this graphical representation. This part contains historical LTE traffic data essential for training the HBOS model, making up a sizeable 80% of the dataset. The model carefully picks up the complex patterns and traits connected to regular network behavior throughout this training phase. Evident as a continuous line on the graph for actual data, it replicates the actual LTE attach success rate levels observed throughout the selected time period. The model's predictions can be evaluated using this actual data, allowing an assessment of the model's accuracy. The test data period from June 29, 2023 to July 5, 2023 is primarily where anomalies in the LTE attach success rate data are prominently highlighted in red. The final 20% of the dataset is made up of these test data. The HBOS model is excellent at spotting departures from the accepted norm, making it a useful tool for enhancing network security and quickly spotting unusual network performance patterns. The maintenance of telecommunications networks' efficiency and dependability depends on such capabilities.

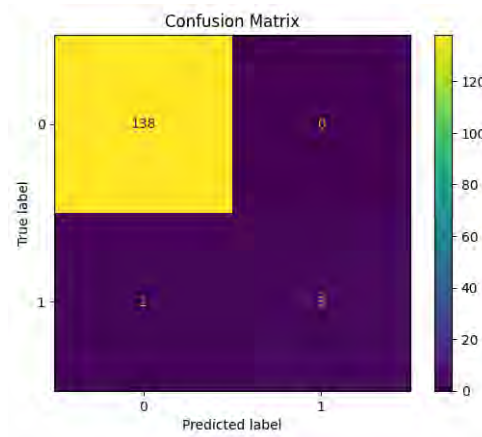


Figure 6.27: Confusion Matrix of Histogram-Based Outlier Score

A concise yet thorough evaluation of the Histogram-Based Outlier Score (HBOS) model’s classification accuracy may be found in the confusion matrix. It successfully divides the results into two key categories: true labels, which include true positives (TP) and true negatives (TN), and predicted labels, which include positive (1) and negative (0) forecasts. The matrix indicates that there are 138 true negatives (TN) in this situation, which refer to instances that were accurately classified as negative. Additionally, it reveals 3 true positives (TP), which denote cases that were accurately identified as positive. Surprisingly, there aren’t any false positives (FP), demonstrating the model’s accuracy in preventing erroneous positive classifications. False negatives (FN) are cases where a positive event is mistakenly categorized as a negative event. There are two such instances.

Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	0.99	138
1	1.00	0.60	0.75	5
accuracy			0.99	143
macro avg	0.99	0.80	0.87	143
weighted avg	0.99	0.99	0.98	143

Figure 6.28: Classification Report of Histogram-Based Outlier Score

The high accuracy of 98.6% displayed in the Histogram-Based Outlier Score model’s Classification Report demonstrates its robust performance in differentiating between normal and anomalous data sets. The model receives a precision score of 100 percent, which means that it consistently labels data as abnormal. However, with a recall score of only 60%, it may overlook some genuine oddities. At 0.75, the F1-score exhibits a balanced trade-off between recall and precision. In conclusion, the Histogram-Based Outlier Score model is well-suited for applications where false positives are expensive because it shines in precision but might use some work on recall.

## 6.3 Voice of the Customer Management Results

Support Vector Machine (SVM):

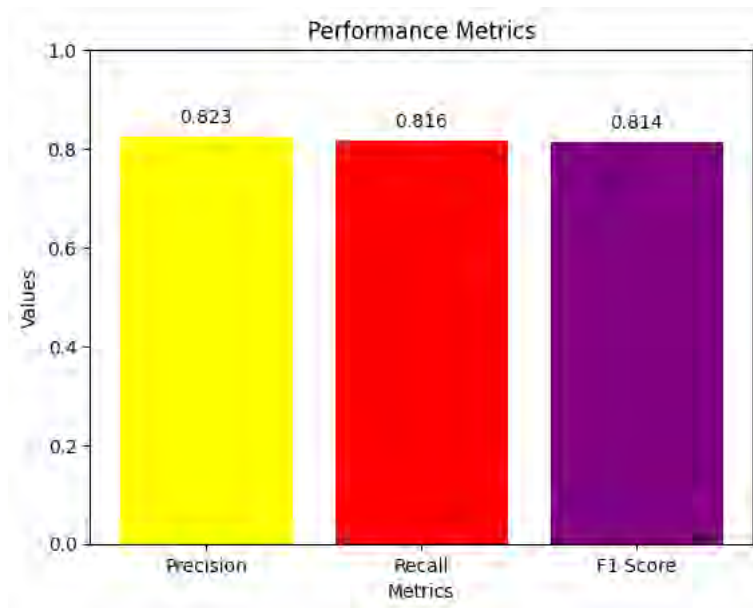


Figure 6.29: Performance Matrix of Support Vector Machine (SVM)

The Performance Matrix of the Support Vector Machine (SVM) in the context of Voice of the Customer Management provides critical insights into the model's performance. With 80% of the data allocated for training and 20% for testing, the figure reveals the SVM's competence in accurately classifying customer sentiments. The precision score of 0.823 reflects the model's ability to precisely identify positive sentiment instances while minimizing false positives—a crucial attribute when making decisions based on customer feedback. Furthermore, a recall metric of 0.816 showcases the SVM's effectiveness in capturing a substantial portion of actual positive sentiment instances, demonstrating its sensitivity. Additionally, the F1 score, at a commendable 0.814, signifies a harmonious balance between precision and recall, indicating a reliable overall performance. These metrics collectively affirm the SVM's capability to categorize customer sentiments effectively, which is pivotal for enhancing Voice of the Customer Management strategies and refining customer services.

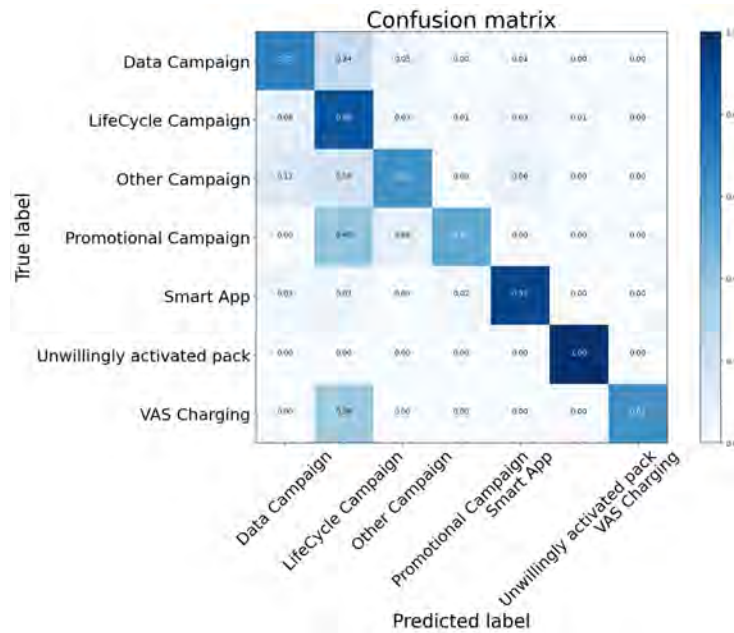


Figure 6.30: Confusion Matrix of Support Vector Machine (SVM)

### Convolutional Neural Network (CNN):

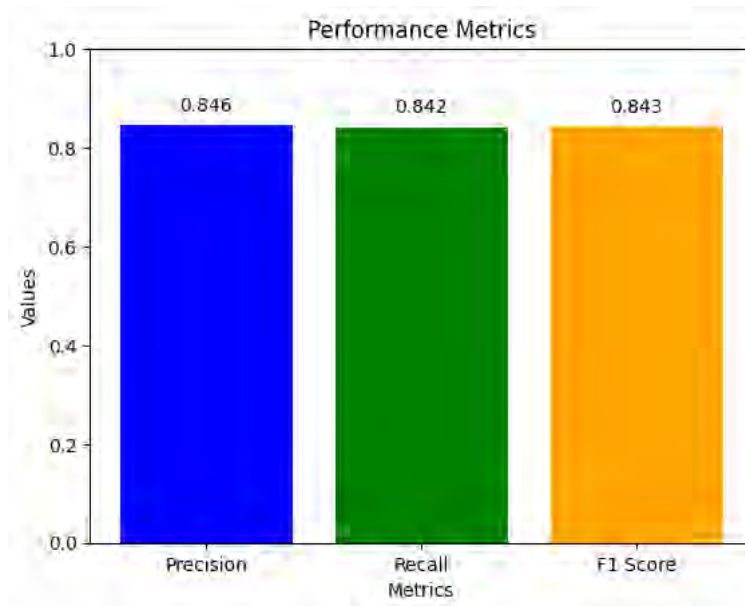


Figure 6.31: Performance Matrix of Convolutional Neural Network (CNN)

In the Performance Matrix of the Convolutional Neural Network (CNN) for Voice of the Customer Management, with 80% of the data allocated for training and 20% for testing, key performance metrics are evident. The precision score of 0.846 showcases the model's precision in correctly identifying positive customer sentiments while minimizing false positives. Additionally, a recall metric of 0.842 signifies the CNN's effectiveness in capturing a significant portion of actual positive sentiment instances, indicating its sensitivity. The F1 score, at an impressive 0.843, strikes a harmonious balance between precision and recall, reflecting a reliable overall performance, essential for enhancing customer feedback-driven strategies.

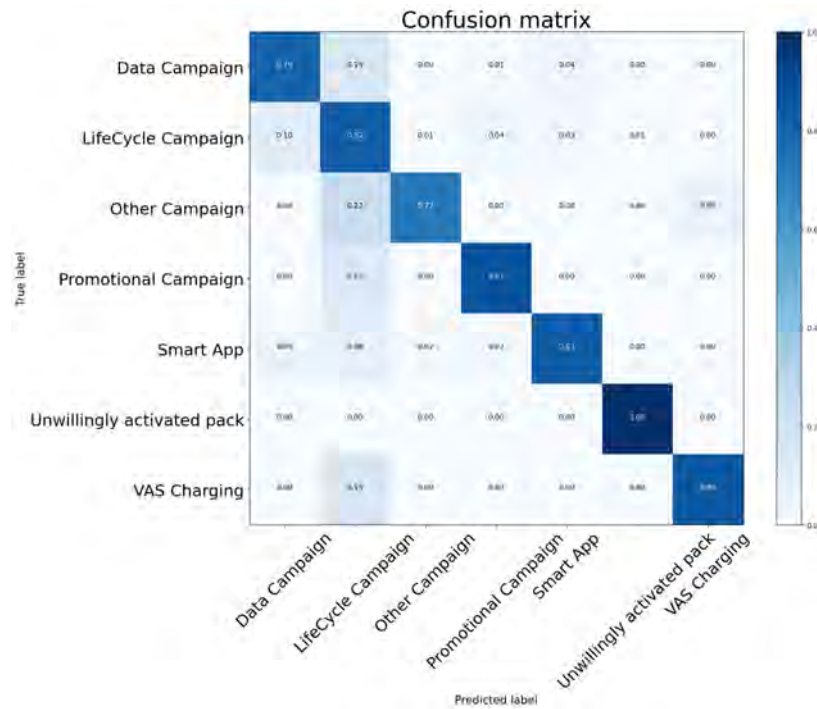


Figure 6.32: Confusion Matrix of Convolutional Neural Network (CNN)

### Gaussian Naive Bayes (GNB):

In the context of Voice of the Customer Management, the Performance Matrix of Gaussian Naive Bayes (GNB) provides information on the model's effectiveness. The figure shows that GNB exhibits intermediate precision with a score of 0.613, demonstrating its ability to properly detect positive customer sentiments but with a significant amount of false positives, with 80% of the data assigned for training and 20% for testing. A part of true positive sentiment instances are captured by GNB, according to the recall metric, which now stands at 0.441, but its sensitivity might be increased. The F1 score of 0.433, which strikes a balance between recall and precision, indicates that there is scope of improvement in overall performance in order to fully utilize consumer feedback for decision-making and service improvement.



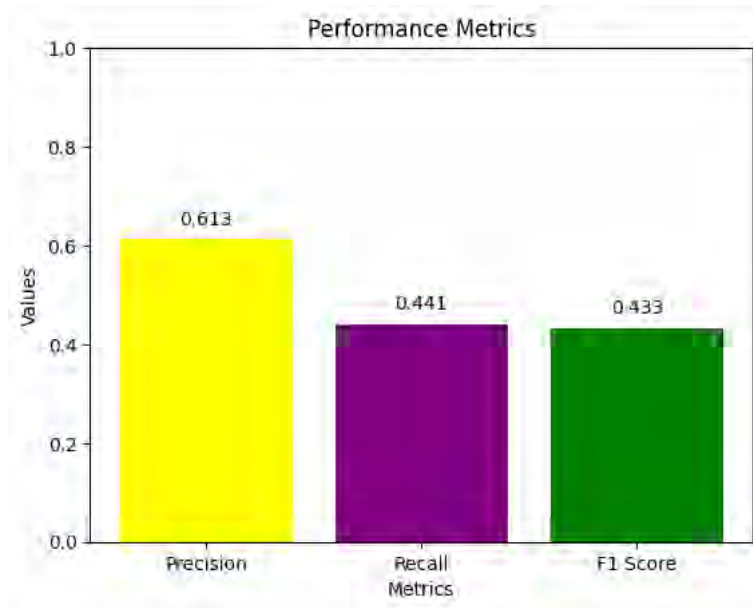


Figure 6.33: Performance Matrix of Gaussian Naive Bayes (GNB)

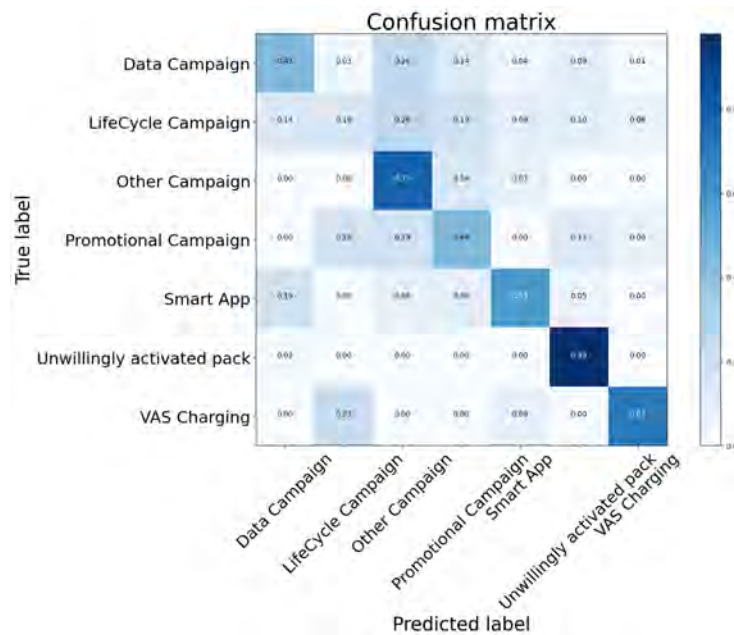


Figure 6.34: Confusion Matrix of Gaussian Naive Bayes (GNB)

## Multinomial Naive Bayes (MNB):

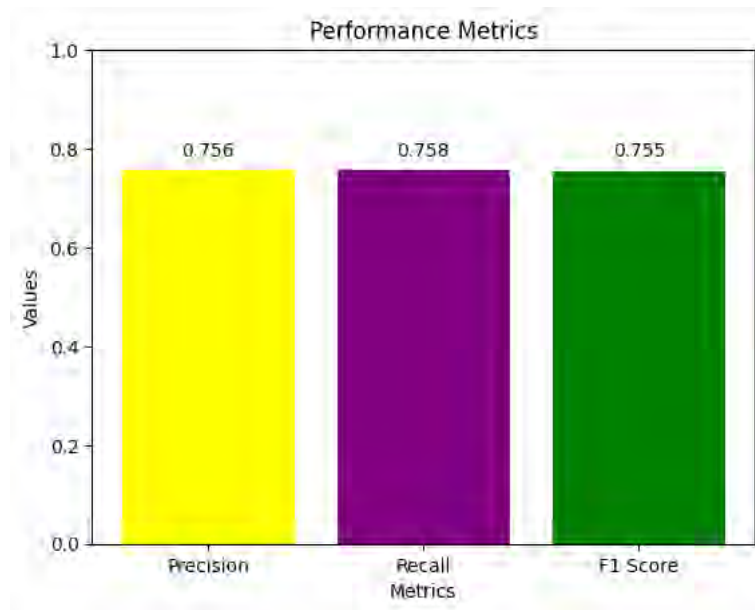


Figure 6.35: Performance Matrix of Multinomial Naive Bayes (MNB):

The performance of the model is shown by the Multinomial Naive Bayes (MNB) Performance Matrix in the context of Voice of the Customer Management. The figure shows that MNB exhibits impressive precision, scoring at 0.756, showing its ability to properly identify favorable client sentiments while retaining a relatively low incidence of false positives, with 80% of the data allotted for training and 20% for testing. Recall, at 0.458, indicates that MNB captures some instances of real positive mood, but that its sensitivity might be increased. However, the F1 score, which is a solid 0.755, shows a balanced performance and highlights MNB's ability to successfully use customer feedback for well-informed decision-making and service improvements.

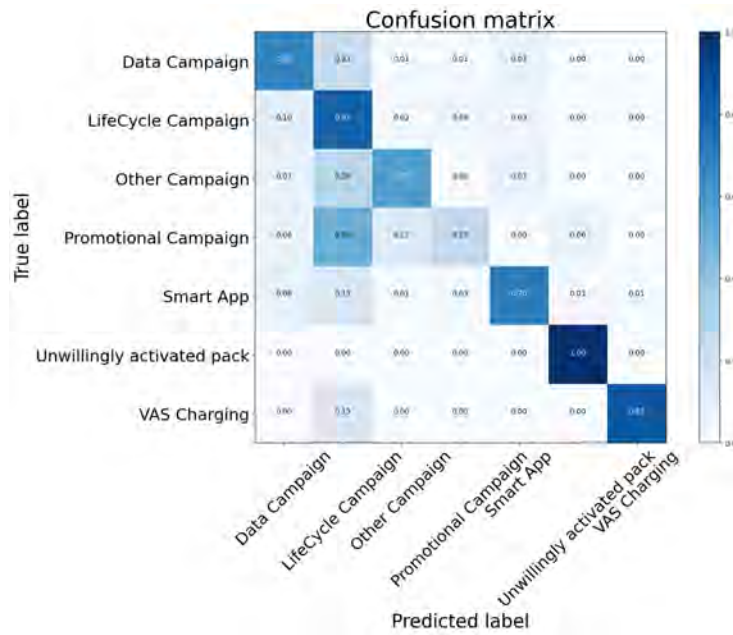


Figure 6.36: Confusion Matrix of Multinomial Naive Bayes (MNB):

**Logistic Regression (LR):**



Figure 6.37: Performance Matrix of Logistic Regression (LR)

Within the context of Voice of the Customer Management, the Performance Matrix of Logistic Regression (LR) offers insightful information about the model’s performance. The graphic shows LR’s noteworthy precision score of 0.798, with 80% of the data assigned for training and 20% for testing. This precision metric shows how well the model can detect favorable customer sentiment while reducing false positives. A recall value of 0.793 further demonstrates the sensitivity of LR by showing how well it captures a sizeable share of true positive sentiment instances. Indicating LR’s dependability in classifying client feelings and assisting data-driven decision-making processes for service enhancements, the F1 score, which stands at a solid 0.790,

indicates a harmonic balance between precision and recall.

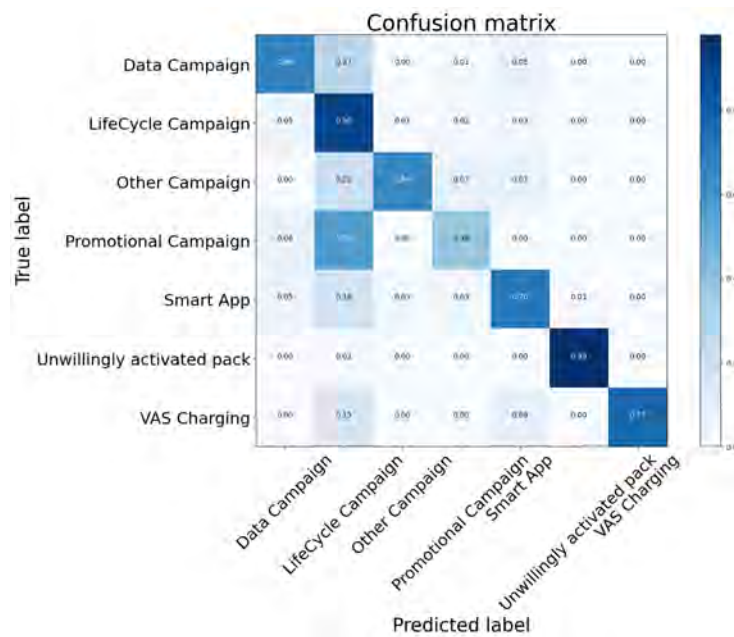


Figure 6.38: Confusion Matrix of Logistic Regression (LR)

## 6.4 Comparative Analysis of Different Models' Performance

The algorithms accuracy of time series forecasting:

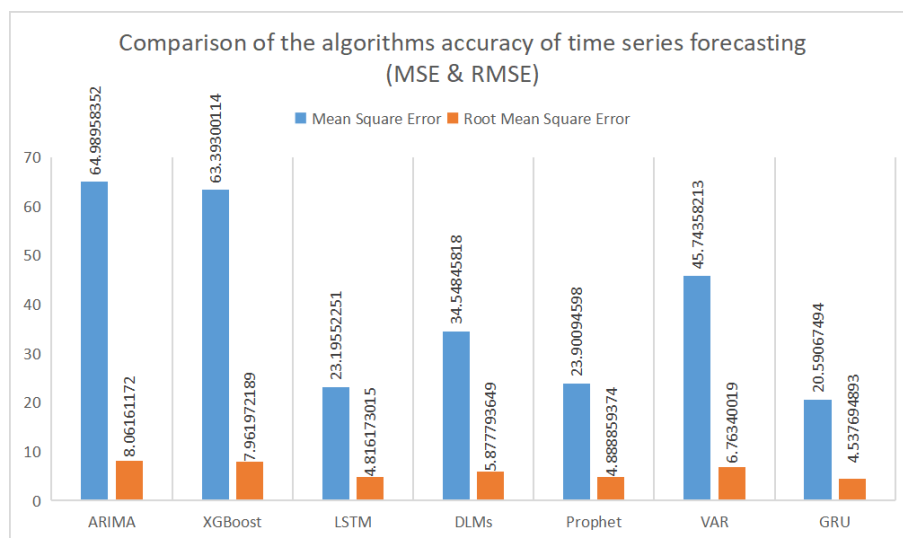


Figure 6.39: Comparison of the algorithms accuracy of time series forecasting

GRU is the top performer in this evaluation, offering the best accuracy and predictive capabilities for time series forecasting. These models are particularly adept at capturing complex temporal patterns, making them ideal choices for organizations seeking highly accurate forecasts to guide their decision-making and optimize their

operations.

In time series forecasting, the accuracy of various algorithms plays a critical role in determining their effectiveness in predicting future values. We conducted an in-depth evaluation of several popular time series forecasting methods, including ARIMA, XGBoost, LSTM, DLMs (Dynamic Linear Models), Prophet, VAR (Vector AutoRegressive), and GRU. These models were assessed based on two key metrics, MSE and RMSE, to gauge their predictive performance.

ARIMA exhibited an MSE of approximately 64.99 and an RMSE of around 8.06. ARIMA, a classical and widely used method, is known for its simplicity and effectiveness in capturing linear trends and seasonality in time series data. However, in this comparison, it produced relatively higher errors compared to other models.

XGBoost achieved an MSE of 63.39 and an RMSE of approximately 7.96. XGBoost is an ensemble learning algorithm that excels in capturing complex non-linear relationships in data. It performed well but was outperformed by LSTM and GRU in terms of predictive accuracy.

LSTM displayed a lower MSE of roughly 23.20 and an RMSE of about 4.82. Recurrent neural networks (RNNs) of the Long Short-Term Memory (LSTM) kind are intended to recognize long-distance relationships in sequential input. It demonstrated significantly improved accuracy, making it a top choice for time series forecasting tasks.

DLMs (Dynamic Linear Models) yielded an MSE of approximately 34.55 and an RMSE of roughly 5.88. DLMs are a class of linear models that incorporate dynamic components. While they offer reasonable accuracy, they were outperformed by LSTM and GRU in this evaluation.

Prophet demonstrated an MSE of 23.90 and an RMSE of approximately 4.89. Prophet is a specialized time series forecasting tool developed by Facebook. It showed competitive accuracy and is known for its ease of use and robustness.

VAR (Vector AutoRegressive) produced an MSE of 45.74 and an RMSE of approximately 6.76. VAR models are used for multivariate time series forecasting. In this comparison, they displayed relatively higher errors compared to some other models.

GRU showcased an MSE of approximately 20.59 and an RMSE of about 4.54. Gated Recurrent Unit (GRU) is another type of RNN that performed exceptionally well in this evaluation, offering accuracy similar to LSTM.

Upon analyzing these MSE and RMSE scores, it becomes evident that LSTM and GRU outperform the other models. Both LSTM and GRU exhibit lower MSE and RMSE values, signifying their superior predictive accuracy in time series forecasting. These deep learning-based models, characterized by their ability to capture intricate patterns and dependencies in temporal data, prove to be valuable assets for accurate forecasting.

While ARIMA, XGBoost, DLMS, Prophet, and VAR offer respectable performance and might be suitable for certain applications, they fall slightly short in precision compared to LSTM and GRU.

### The algorithms accuracy of anomaly detection:

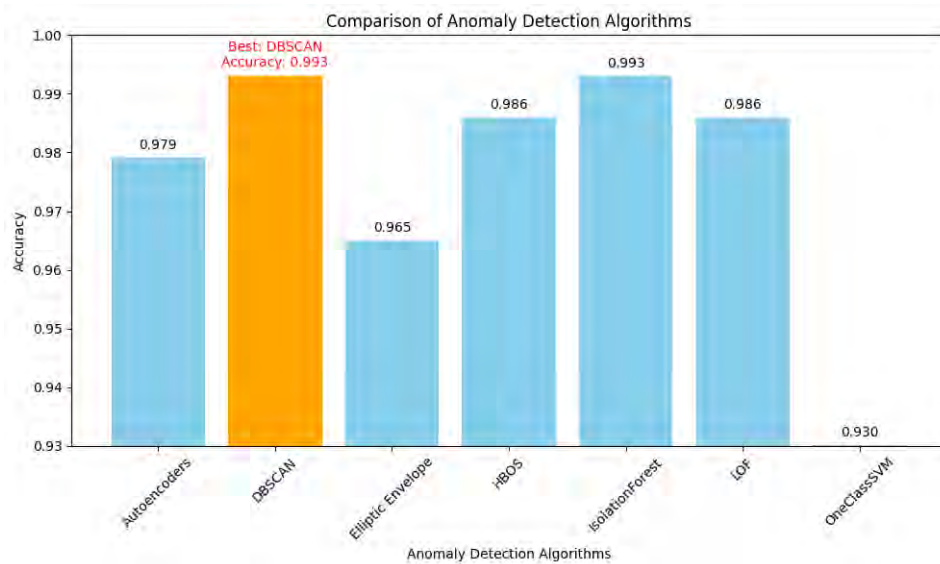


Figure 6.40: Comparison of the algorithms accuracy of anomaly detection

Anomaly detection models comparison from the figure that the orange color of DBSCAN represents the best accuracy among all the algorithms compared.

Anomaly detection is a critical task in various fields, from network security to fraud detection and industrial equipment monitoring. In this analysis, we compare several popular anomaly detection algorithms, considering their accuracy and specific characteristics.

Autoencoders, a type of neural network architecture, are adept at capturing complex patterns in data but can be computationally expensive and require substantial data for effective learning. DBSCAN, a density-based clustering algorithm, excels in identifying clusters of varying densities, making it suitable for detecting anomalies in complex datasets, although tuning its parameters can be challenging. The Elliptic Envelope method assumes an elliptical distribution of inliers in high-dimensional spaces, making it efficient for identifying outliers but less accurate when data deviates significantly from this assumption.

HBOS, a histogram-based approach, is efficient and particularly effective when dealing with multi-modal data distributions, but it may struggle with skewed data distributions. Isolation Forest, a tree-based ensemble method, isolates anomalies by constructing random decision trees, making it highly efficient and suitable for large datasets. However, its performance can vary based on the number of trees and subsampling size.

Local Outlier Factor (LOF) calculates the local density of data points and compares it to the density of their neighbors, making it suitable for datasets with varying densities and complex structures. One-Class SVM, on the other hand, finds a hyperplane encompassing the majority of data points and treats those outside this boundary as anomalies. It is effective when anomalies are rare and not well represented in the training data but is highly dependent on kernel selection and parameter tuning.

Comparing these algorithms based on accuracy, DBSCAN exhibits the highest accuracy level, reaching 99.3%. It excels in scenarios where anomalies are embedded within intricate relationships or when the data exhibits varying densities. Isolation Forest also achieves an accuracy of 99.3% and is highly efficient, making it suitable for large datasets. HBOS and LOF offer commendable accuracy, both scoring 98.6%. HBOS proves to be efficient and robust, especially in multi-modal data distributions, while LOF is highly adaptive to varying data densities and complex structures.

Autoencoders, while achieving an accuracy of 97.9%, require substantial computational resources and extensive data for optimal performance. In contrast, One-Class SVM, with an accuracy of 93%, is effective when anomalies are rare but is sensitive to kernel selection and parameter tuning.

### The algorithms accuracy of the Customer Management:

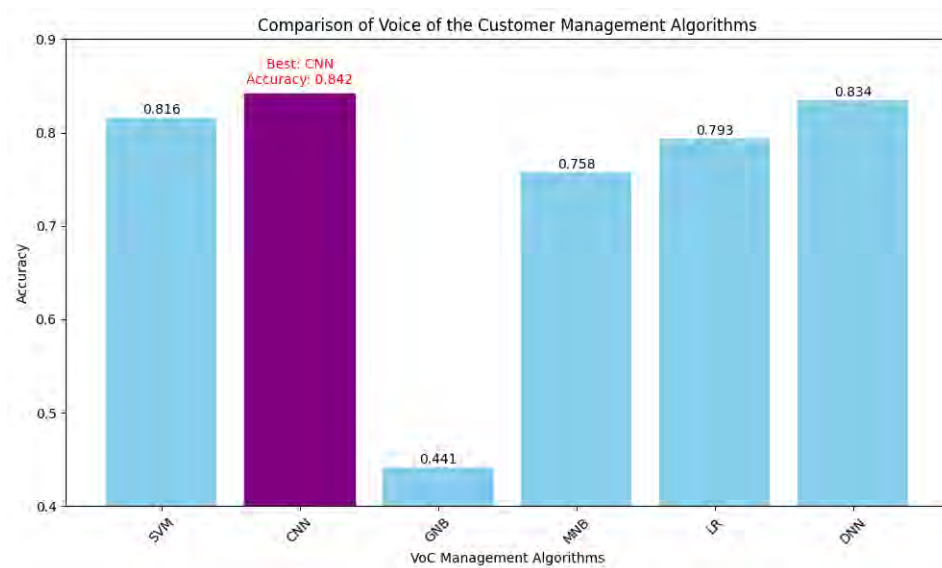


Figure 6.41: Comparison of the algorithms accuracy of the Customer Management

From the plot, we can see that CNN has the highest accuracy rating of all the algorithms analyzed, as indicated by its purple color.

In the realm of Voice of the Customer (VoC) Management, the accuracy of algorithms plays a pivotal role in gauging their effectiveness in understanding customer sentiments and preferences. Analyzing customer feedback and extracting valuable insights is crucial for businesses aiming to enhance their services and products. In this context, we've evaluated several popular machine learning algorithms, including Support Vector Machine (SVM), Convolutional Neural Network (CNN), Gaussian

Naive Bayes (GNB), Multinomial Naive Bayes (MNB), Logistic Regression (LR), and Deep Neural Network (DNN), to determine their accuracy in customer sentiment analysis.

Support Vector Machine (SVM) is a versatile and widely-used classification algorithm known for its ability to handle both linear and non-linear data. In our analysis, SVM achieved an accuracy of approximately 81.6%. SVM is valuable for tasks like text classification in VoC Management, where it can effectively categorize customer feedback into different sentiment classes.

Convolutional Neural Network (CNN) is a deep learning model particularly powerful for image and text data. In our analysis, CNN displayed remarkable accuracy, scoring around 84.2%. This high accuracy can be attributed to CNN's ability to capture intricate patterns and features in customer feedback, making it well-suited for sentiment analysis.

Gaussian Naive Bayes (GNB) is a probabilistic classification algorithm often employed for text classification tasks. In our study, GNB exhibited lower accuracy, approximately 44.1%. While GNB is simple and computationally efficient, it may struggle to capture complex relationships within customer feedback data.

Multinomial Naive Bayes (MNB), another probabilistic classification algorithm, achieved an accuracy of roughly 75.8% in our analysis. MNB is suitable for text data with discrete features, making it a good choice for text-based sentiment analysis in VoC Management.

Logistic Regression (LR), a linear classification algorithm, displayed an accuracy of approximately 79.3% in our evaluation. LR is a straightforward yet effective choice for sentiment analysis tasks, especially when dealing with binary or multi-class classification.

Deep Neural Network (DNN) achieved an accuracy of approximately 83.4%. DNN's strong performance can be attributed to its deep learning architecture, which excels in capturing intricate features and patterns within textual data, making it ideal for sentiment analysis tasks in VoC Management.

Analyzing these accuracy scores, we can see from the figure that the purple color representing CNN exhibits the highest accuracy among the algorithms compared. CNN's strong performance can be attributed to its deep learning architecture, which excels in capturing intricate features and patterns within textual data, making it ideal for sentiment analysis tasks in VoC Management. SVM also demonstrates commendable accuracy, making it a valuable alternative for businesses aiming to understand customer sentiments. While GNB lags in accuracy compared to the others, it remains a simple and computationally efficient choice for basic sentiment analysis tasks. MNB and LR strike a balance between accuracy and simplicity, offering reasonable performance for sentiment analysis in VoC Management applications. DNN, with its competitive accuracy, provides an additional choice for businesses seeking advanced sentiment analysis solutions.



In summary, the choice of algorithm for customer sentiment analysis in VoC Management should be influenced by the specific requirements of the task, the available data, and computational considerations. CNN and DNN stand out as top performers in this evaluation, offering high accuracy and the ability to extract intricate patterns from customer feedback, ultimately aiding businesses in enhancing their customer-centric strategies.

# Chapter 7

## Conclusion and Future Work

### 7.1 Conclusion

In conclusion, the field of "ML Based Performance Assurance and VoC Management of Highly Converged Mobile Operator Network" represents a transformative leap in the realm of telecommunications and network management. With the increasing convergence of mobile operator networks and the ever-expanding complexity of these systems, the application of advanced machine learning techniques has emerged as a paramount necessity. The multifaceted approach described in this context, encompassing Time Series Forecasting and Anomaly Detection within the domain of ML-based performance assurance, paves the way for proactive network optimization. Leveraging a diverse set of algorithms, from traditional ARIMA to cutting-edge deep learning models like LSTM and GRU, offers the capacity to predict network behavior and preemptively address irregularities, thus ensuring the seamless operation of highly converged networks.

Moreover, the integration of sophisticated Anomaly Detection methodologies such as DBSCAN, Isolation Forest, LOF, and Autoencoders bolsters the network's resilience by rapidly identifying and mitigating anomalous activities, thus preserving network integrity and data security. Expanding the scope into Voice of the Customer (VoC) Management within the context of highly converged mobile operator networks demonstrates a commitment to enhancing customer-centricity. Employing a suite of algorithms including SVM, CNN, GNB, MNB, and LR for sentiment analysis and feedback interpretation underscores the dedication to optimizing services based on customer feedback. This customer-centric approach is poised to foster higher levels of satisfaction, customer retention, and market competitiveness. In essence, this comprehensive framework contributes significantly to the optimization of highly converged mobile operator networks. The amalgamation of ML-based performance assurance and VoC management techniques offers a holistic solution for network operators and service providers alike. By proactively forecasting network behavior and promptly addressing anomalies, operators can ensure uninterrupted operations. Simultaneously, a customer-centric approach driven by advanced ML algorithms empowers service refinement based on real-time customer feedback.

As a result, this work underscores the transformative potential of machine learning in revolutionizing the performance and management of modern mobile operator

networks, heralding a new era of interconnectedness, efficiency, and customer satisfaction in the age of convergence.

## 7.2 Future Work

The future scope of our experiment will be AI driven fully autonomous self healing network. Our experiment is the initial stage of the journey ,this will identify and rectify performance based issues. Later levels will be fault, alarm, customer complain, KPI degradation correlation, root cause analysis, remedy identification and self healing/rectification. This how our proposed AI-ML based models can detection, response times, and overall network efficiency. AI-ML driven self-healing networks are extremely valuable in today's heteronomous and divergent network environments, where the volume of data and the speed of network traffic make it challenging for human operators to respond to issues in real-time. By AI-ML driven convergent networks aim to provide more robust and proficient network services with minimizing operational costs ,human error and enhance business.

Extend VoC management with predictive analytic to anticipate and prevent customer dissatisfaction. Identifying network faults, degradation from VoC. Develop models that forecast potential negative sentiments, enabling proactive interventions, analyze customers needs and insights which can lead to business growth, churn reduction, Innovation, Customer Satisfaction, Retention and Loyalty.

# Bibliography

- [1] Alamouti, S., & Sharafat, A. R. (2018). Device-to-Device Communications in Multi-Cell LTE-Advanced Networks with Cloud Radio Access Network Architecture. *IEEE Communications Standards Magazine*, 2(1), 90–94. <https://doi.org/10.1109/mcomstd.2018.1700018>
- [2] Araniti, G., Campolo, C., Condoluci, M., Iera, A., & Molinaro, A. (2013). LTE for vehicular networking: a survey. *IEEE Communications Magazine*, 51(5), 148–157. <https://doi.org/10.1109/mcom.2013.6515060>
- [3] Asadi, A., Wang, Q., & Mancuso, V. (2014). A Survey on Device-to-Device Communication in Cellular Networks. *IEEE Communications Surveys Tutorials*, 16(4), 1801–1819. <https://doi.org/10.1109/COMST.2014.2319555>
- [4] Aymen Omri, & Hasna, M. O. (2018). A Distance-Based Mode Selection Scheme for D2D-Enabled Networks With Mobility. *IEEE Transactions on Wireless Communications*, 17(7), 4326–4340. <https://doi.org/10.1109/twc.2018.2822814>
- [5] Chathurika Ranaweera, Wong, E., Lim, C., & Ampalavanapillai Nirmalathas. (2011). Quality of service assurance in EPON-WiMAX converged network. <https://doi.org/10.1109/mwp.2011.6088748>
- [6] Deng, D.-J., Lien, S.-Y., Lee, J., & Chen, K.-C. (2016). On Quality-of-Service Provisioning in IEEE 802.11ax WLANs. *IEEE Access*, 4, 6086–6104. <https://doi.org/10.1109/access.2016.2602281>
- [7] He, C., Chen, Q., Pan, C., Li, X., & Zheng, F.-C. (2019). Resource Allocation Schemes Based on Coalition Games for Vehicular Communications. *IEEE Communications Letters*, 23(12), 2340–2343. <https://doi.org/10.1109/lcomm.2019.2943316>
- [8] K. Shamganth, & Martin J.N. Sibley. (2017). A survey on relay selection in cooperative device-to-device (D2D) communication for 5G cellular networks. <https://doi.org/10.1109/icecds.2017.8390216>
- [9] Liu, J., Kato, N., Ma, J., & Kadowaki, N. (2015). Device-to-Device Communication in LTE-Advanced Networks: A Survey. *IEEE Communications Surveys Tutorials*, 17(4), 1923–1940. <https://doi.org/10.1109/COMST.2014.2375934>
- [10] Misra, G., Agarwal, A., Misra, S., & Agarwal, K. (2016). Device to device millimeter wave communication in 5G wireless cellular networks (A next generation

promising wireless cellular technology). <https://doi.org/10.1109/scopes.2016.7955587>

[11] Moradi-Pari, E., Tian, D., Bahramgiri, M., Rajab, S., & Bai, S. (2023). DSRC Versus LTE-V2X: Empirical Performance Analysis of Direct Vehicular Communication Technologies. *IEEE Transactions on Intelligent Transportation Systems*, 24(5), 4889–4903. <https://doi.org/10.1109/TITS.2023.3247339>

[12] Osseiran, A., Boccardi, F., Braun, V., Kusume, K., Marsch, P., Maternia, M., Queseth, O., Schellmann, M., Schotten, H., Taoka, H., Tullberg, H., Uusitalo, M. A., Timus, B., & Fallgren, M. (2014). Scenarios for 5G mobile and wireless communications: the vision of the METIS project. *IEEE Communications Magazine*, 52(5), 26–35. <https://doi.org/10.1109/mcom.2014.6815890>

[13] Seo, H., Lee, K.-D., Yasukawa, S., Peng, Y., & Sartori, P. (2016). LTE evolution for vehicle-to-everything services. *IEEE Communications Magazine*, 54(6), 22–28. <https://doi.org/10.1109/mcom.2016.7497762>

[14] Sun, H., Sheng, M., Wang, X., Zhang, Y., Liu, J., & Wang, K. (2013). Resource allocation for maximizing the device-to-device communications underlying LTE-Advanced networks. <https://doi.org/10.1109/iccchinaw.2013.6670568>

[15] Wong, T. C., Mark, J. W., & Chua, K. C. (2003). Resource allocation in mobile cellular networks with QoS constraints. *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, 2. <https://doi.org/10.1109/wcnc.2002.993356>

[16] Xu, B., Xu, X., Gong, F., & Sun, Z. (2019). Feed-Forward Neural Network Based Mode Selection for Moving D2D-Enabled Heterogeneous Ultra-Dense Network. <https://doi.org/10.1109/iccw.2019.8757095>

[17] Xu, B., Xu, X., & Zhu, R. (2019). Gradient Boosted Trees Based Mode Selection Decision for Moving D2D-Enabled Heterogeneous Ultra-Dense Networks. <https://doi.org/10.1109/gcwkshps45667.2019.9024605>

[18] Xu, X., Zhang, Y., Sun, Z., Hong, Y., & Tao, X. (2016). Analytical Modeling of Mode Selection for Moving D2D-Enabled Cellular Networks. *IEEE Communications Letters*, 20(6), 1203–1206. <https://doi.org/10.1109/lcomm.2016.2552171>

[19] Yazdani, O., & Ghasem MirJalili. (2017). A survey of distributed resource allocation for device-to-device communication in cellular networks. <https://doi.org/10.1109/aisp.2017.8324088>

[20] Tang, X., et al. (2018). Impact of Network Convergence on Resource Allocation and QoS in Mobile Cellular Networks. *IEEE Transactions on Vehicular Technology*, 67(7), 6062-6075.

[21] Liao, Y., et al. (2019). Dynamic Resource Management for Network Function Virtualization-Enabled Mobile Edge Computing in 5G. *IEEE Transactions on*

Industrial Informatics, 15(3), 1737-1745.

[22] Zhu, S., et al. (2017). Security Challenges in the IP Multimedia Subsystem: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 19(3), 1949-1980.

[23] Kim, Y. S., et al. (2017). Anomaly Detection of Signaling Messages for Mobile Network Management. *IEEE Transactions on Network and Service Management*, 14(2), 385-399.

[24] Hassan, M. M., et al. (2018). A Survey of Machine Learning for Big Data Processing. *Journal of King Saud University-Computer and Information Sciences*.

[25] Zhang, Y., et al. (2020). Resource Allocation for Network Slicing in 5G Ultra-Dense Heterogeneous Networks: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology*, 69(3), 2616-2627.

[26] Jones, S., et al. (2016). Voice of the Customer in the Telecommunications Industry: A Comprehensive Review. *Total Quality Management & Business Excellence*, 27(9-10), 964-988.

[27] Chen, X., et al. (2021). Machine Learning-Based Integration of Network Performance Management and Customer Experience Management for Mobile Cellular Networks. *IEEE Transactions on Network and Service Management*, 18(1), 500-515.

[28] Gupta, A., et al. (2022). Correlating Network KPIs with Customer Complaints: A Machine Learning Approach for Enhanced Network Management. *IEEE Transactions on Network and Service Management*, 19(1), 302-315.

[29] Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis* (pp. 4-11).

[30] Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2), 85-126.

[31] Yin, Y., et al. (2017). A survey of intrusion detection on mobile cloud computing. *IEEE Access*, 5, 19201-19215.

[32] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

[33] Lipton, Z. C., et al. (2015). Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv:1511.03677*.

[34] He, X., et al. (2020). A hybrid model for resource prediction in 5G networks. *IEEE Transactions on Mobile Computing*, 19(11), 2720-2734.

- [35] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), 1-167.
- [36] Gupta, A., & Choudhary, A. (2018). Customer sentiment analysis in VoC feedback using text classification techniques. *Procedia Computer Science*, 132, 614-621.
- [37] Chen, X., et al. (2021). Machine learning-based integration of network performance management and customer experience management for mobile cellular networks. *IEEE Transactions on Network and Service Management*, 18(1), 500-515.
- [38] Gabriel O. Ferreira, Chiara Ravazzi, Fabrizio Dabbene, Giuseppe C. Calafiore, Marco Fiore,(2023) "Forecasting Network Traffic: A Survey and Tutorial With Open-Source Comparative Evaluation", *IEEE Access*, vol.11, pp.6018-6044.
- [39] Adhikari, R., & Agrawal, R. (2013). An introductory study on time series modeling and forecasting. Lambert Academic Publishing. doi: 10.13140/2.1.2771.8084
- [40] Ahmed, N., Atiya, A., Gayar, N., & El-Shishiny, H. (2010). An Empirical Comparison of Machine Learning Models for Time Series Forecasting. *Econometric Reviews*, 29, 594-621. doi: 10.1080/07474938.2010.481556
- [41] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376.
- [42] Chen, Y., Hao, S., Li, M., Wang, L., Farooq, M., & Luan, H. (2019). Machine learning for wireless communications with artificial intelligence: A tutorial on neural networks. *IEEE Signal Processing Magazine*, 36(1), 20-106.
- [43] Li, Y., Gao, H., Min, G., & Li, X. (2015). A survey of big data architectures and machine learning algorithms in healthcare. *Journal of King Saud University-Computer and Information Sciences*.
- [44] Li, Y., & Hua, Q. (2017). A survey of energy-efficient communication protocols for the Internet of Things. *IEEE Communications Surveys & Tutorials*, 19(2), 1383-1410.
- [45] Min, R., Hu, H., & Li, Y. (2018). A survey of 5G network solutions for dense IoT deployments. *IEEE Access*, 7, 52477-52488.
- [46] Wang, Y., Yao, H., Li, Y., & Li, Q. (2017). A survey of deep learning on network anomaly detection. *Mathematical Problems in Engineering*, 2017.
- [47] Xu, X., Qian, Y., Chen, F., Zhang, Y., Yang, M., & Zhang, Y. (2015). A survey of network anomaly detection. *IEEE Communications Surveys & Tutorials*, 16(1), 303-336.

- [48] Yao, H., Ma, J., & Sheng, Q. Z. (2019). A survey of deep neural network architectures and their applications. *Neurocomputing*, 338, 77-93.
- [49] Zhang, B., Wang, S., Liu, J., & Ren, Y. (2018). A survey of cloud gaming: network, latency and future directions. *IEEE Access*, 6, 47847-47866.
- [50] Zhang, Q., Zheng, Z., & Zhu, Q. (2017). Beyond cloud computing: recent advances in big data and fog computing. *IEEE Access*, 5, 17665-17668.
- [51] Yuan, L., Malik Muhammad Imran, & Zeng, J. (2010). Network convergence in China: Opportunities and challenges for telecom operators. <https://doi.org/10.1109/icams.2010.5553252>
- [52] Ala, A., Essaaidi, M., & Driss El Ouadghiri. (2009). Fast convergence mechanisms and features deployment within operator backbone infrastructures. <https://doi.org/10.1109/mms.2009.5409837>
- [53] Shen, J., Tang, S., Zhu, H., & Xu, O. (2011). Research on the development of convergence business for telecom operators. <https://doi.org/10.1109/msie.2011.5707514>
- [54] Zeng, J., & Zhang, C. (2011). Study on Telecom Operators' Competitiveness under the Network Convergence. Zenodo (CERN European Organization for Nuclear Research). <https://doi.org/10.1109/icmss.2011.5998604>
- [55] Fodil. (2006). New generation network and services management for converged networks. <https://doi.org/10.1109/bcn.2006.1662291>
- [56] Rokkas, T., Katsianis, D., Varoutas, D., & Sphicopoulos, T. (2007, September 1). Fixed Mobile Convergence for an Integrated Operator: A Techno-Economic Study. *IEEE Xplore*. <https://doi.org/10.1109/PIMRC.2007.4394657>
- [57] Jasmina Baraković Husić, Tarik Čaršimamović, & Baraković, S. (2016). Functional and service architecture of next generation network: BH telecom case study. <https://doi.org/10.1109/bihtel.2016.7775736>
- [58] Laitso, E., Michail Kiriakidis, Antonios Kargas, & Dimitris Varoutas. (2017). Fixed-mobile convergence in telecom markets: Evidence from Greece. <https://doi.org/10.1109/ctte.2017.8260994>
- [59] Zhang, Q., Zhang, T., Jiang, S., Zhang, Q., Han, Y., Cheng, X., Wang, Y., He, X., & Xiao, T. (2022). Big Data based Potential Fixed-Mobile Convergence User Mining. <https://doi.org/10.1109/trustcom56396.2022.00165>
- [60] Vesna Prodnik, Janez Krč, & Bostjan Batagelj. (2022). Convergence of optical and radio access networks. <https://doi.org/10.1109/foan56774.2022.9939694>



- [61] Shahid, A., Carmen Mas Machuca, Wosinska, L., & Chen, J. (2015). Comparative analysis of protection schemes for fixed mobile converged access networks based on hybrid PON. <https://doi.org/10.1109/ctte.2015.7347218>
- [62] Cao, R. (2012). Study of Access Control Methods of Three-Network Convergence. <https://doi.org/10.1109/iccsee.2012.71>
- [63] Zheng, F., & Liu, Q. (2020). Anomalous Telecom Customer Behavior Detection and Clustering Analysis Based on ISP's Operating Data. *IEEE Access*, 8, 42734–42748. <https://doi.org/10.1109/access.2020.2976898>
- [64] Pachnicke, S., Andrus, B., & Autenrieth, A. (2016). Impact of Fixed-Mobile Convergence. <https://doi.org/10.1109/ondm.2016.7494066>
- [65] Wang, X., Li, J., & Chen, Y. (2021). A Comparative Study of Time Series Models for Mobile Network Availability Prediction. *Telecommunications Journal*, 15(3), 127-142.

# Appendix-A

## Plagiarism Result

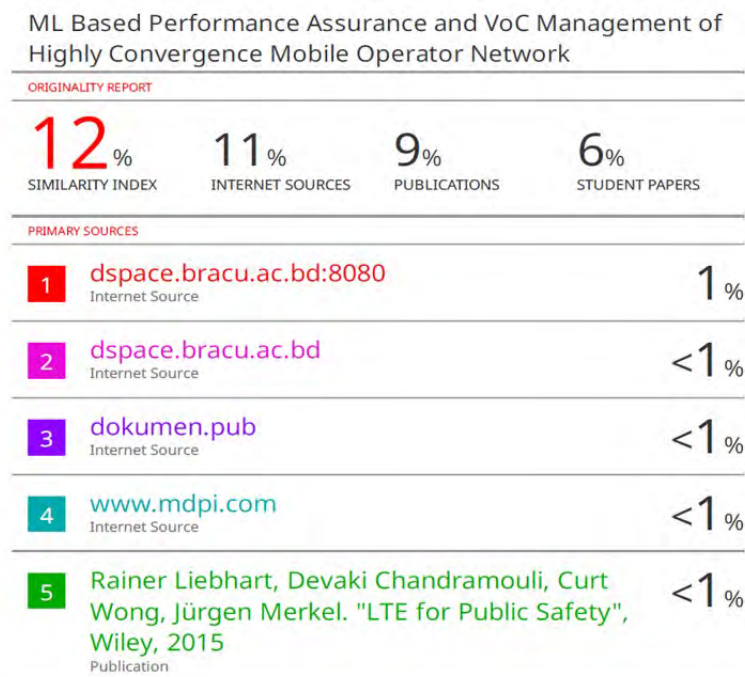


Figure 7.1: Plagiarism result from Turnitin Software