CUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Forecasting Arrivals to a Call Centre Using Machine Learning and Deep Learning

Rukia Kasauli NAKKAZI

Computer Engineering Department

October, 2024

CUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS APPROVAL

Forecasting Arrivals to a Call Centre Using Machine Learning and Deep Learning

Rukia Kasauli NAKKAZI

Computer Engineering Department

This Doctorate Thesis was evaluated by the following Jury Members on 11/10/2024 and was approved by unanimity / majority of votes.

Jury	: Prof. Dr. .M. Fatih AKAY	(Advisor)	…………………
	: Prof. Dr. Mehmet Uğraş CUMA		…………………
	: Prof. Dr. Ibrahim Taner OKUMUŞ		…………………
	: Doç. Dr.Serkan KARTAL		…………………
	: Doç. Dr. Fatih KILIÇ		…………………

This Thesis was written in the Department of Computer Engineering, Institute of Natural and Applied Sciences.

Thesis Number:

Prof. Dr. Sadık DİNÇER
Director
Institute of Natural and Applied Sciences

CONTENTS

CUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Forecasting Arrivals to a Call Centre Using Machine Learning and Deep Learning

Rukia Kasauli NAKKAZI

Advisor: Prof. Dr. M. Fatih AKAY

Department of Computer Engineering

ABSTRACT

With the increasing complexity of call center operations, accurate call arrival prediction has become a crucial research area, attracting significant attention from both academia and industry. Forecasting call arrivals is essential for effective resource allocation, staffing decisions, and service level planning, ultimately improving operational efficiency and customer satisfaction.

This study investigates the effectiveness of combining Bayesian optimization (BO) with feature selection to enhance call arrival prediction accuracy. We analyzed various machine learning (ML) and deep learning (DL) models, including Random Forest (RF), Multi-layer Perceptron (MLP), Support Vector Machine (SVM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM).

Using real-world call center data, we employed three datasets with daily, hourly, and half-hourly observations to predict call volume and Average Handling Time (AHT). Autocorrelation functions revealed patterns in these datasets. Both multivariate and univariate prediction approaches were evaluated.

Using the Mean Absolute Error (MAE) metric and Root Mean Squared Error metric (RMSE), we demonstrated that optimized models with selected features consistently outperformed baseline models and optimized models with all features. Specifically, DL models, notably CNN and LSTM, showed robust responsiveness to combined BO and feature selection. ML models, especially MLP and RF, had strong baseline performances and also benefited from our approach. Depending on the dataset, performance gains from our approach ranged from 14% to 88% for ML models and 92% to 97% for DL models for both call volume and AHT predictions for multivariate models. For univariate models with only BO, percentage improvements ranged from 20% to 71% for ML models and 50% to 96.6% for DL models. These results highlight the effectiveness of combining feature selection with BO for more accurate predictions of both call volume and AHT.

Keywords: Call Arrivals forecasting, machine learning, Hyperparameter Optimization, Feature Selection.

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

Makine Öğrenimi ve Derin Öğrenme Kullanarak Bir Çağrı Merkezine Gelenlerin Tahmin Edilmesi

Rukia Kasauli NAKKAZI

Danışman: Prof. Dr. M. Fatih AKAY

Bilgisayar Mühendisliği Anabilim Dalı

ÖZ

Çağrı merkezi operasyonlarının artan karmaşıklığı ile birlikte, çağrı gelişlerini doğru bir şekilde tahmin etmek önemli bir araştırma alanı haline gelmiş ve hem akademinin hem de sanayinin büyük ilgisini çekmiştir. Çağrı gelişlerini tahmin etmek, etkili kaynak tahsisi, personel kararları ve hizmet seviyesi planlaması için gereklidir; bu da nihayetinde operasyonel verimliliği ve müşteri memnuniyetini artırmaktadır.

Bu çalışma, çağrı geliş tahmin doğruluğunu artırmak için Bayes optimizasyonunu (BO) özellik seçimi ile birleştirmenin etkinliğini araştırmaktadır. Rastgele Orman (RF), Çok Katmanlı Algılayıcı (MLP), Destek Vektör Makineleri (SVM), Tekrarlayan Sinir Ağı (RNN), Evrişimli Sinir Ağı (CNN) ve Uzun Kısa Süreli Bellek (LSTM) dahil olmak üzere çeşitli makine öğrenimi (ML) ve derin öğrenme (DL) modelleri analiz edilmiştir.

Gerçek dünyadaki çağrı merkezi verilerini kullanarak, çağrı hacmi ve Ortalama İşlem Süresi (AHT) tahmini yapmak için günlük, saatlik ve yarım saatlik gözlemlere sahip üç veri seti kullanılmıştır. Otokorelasyon fonksiyonları, bu veri setlerindeki kalıpları ortaya çıkarmıştır. Hem çok değişkenli hem de tek değişkenli tahmin yaklaşımları değerlendirilmiştir.

Ortalama Mutlak Hata (MAE) metriği ve Kök Ortalama Kare Hata (RMSE) metriği kullanılarak, seçilmiş özelliklerle optimize edilmiş modellerin, tüm özelliklere sahip temel modelleri ve optimize edilmiş modelleri sürekli olarak geride bıraktığı gösterilmiştir. Özellikle, DL modelleri, özellikle CNN ve LSTM, Bayes optimizasyonu ve özellik seçimi kombinasyonuna karşı güçlü bir yanıt vermiş olup. ML modelleri, özellikle MLP ve RF, güçlü temel performanslar sergilemiştir ve bu yaklaşımdan fayda sağlamıştır. Veri setine bağlı olarak, çağrı hacmi ve AHT tahminleri için çok değişkenli modellerde yaklaşımımızdan sağlanan performans artışı, ML modelleri için %14 ila %88, DL modelleri için ise %92 ila %97 arasında değişmiştir. Sadece BO içeren tek değişkenli modellerde ise, ML modelleri için iyileşme oranı %20 ila %71, DL modelleri için %50 ila %96,6 arasında değişmiştir. Bu sonuçlar, hem çağrı hacmi hem de AHT tahminlerinde daha doğru tahminler elde etmek için özellik seçiminin Bayes optimizasyonu ile birleştirilmesinin etkinliğini ortaya koymaktadır.

Anahtar Kelimeler: Çağrı Geliş Tahmini, makine öğrenimi, Hiperparametre Optimizasyonu, Özellik Seçimi

GENİŞLETİLMİŞ ÖZET

Çağrı merkezleri, modern iş dünyasında müşteri etkileşimleri ve destek hizmetleri için birincil temas noktası olarak kritik bir rol oynar. Çeşitli müşterilerin dinamik ihtiyaçlarını karşılamak amacıyla stratejik olarak konumlandırılan bu merkezler, eğitimli çalışanların çeşitli kanallar aracılığıyla yardım sağladığı, sorunları çözdüğü ve işlemleri yürüttüğü iletişim merkezleri olarak işlev görür. Çağrı merkezi operasyonlarının artan karmaşıklığı ve dinamik yapısı nedeniyle, çağrı tahmini önemli bir araştırma alanı olarak ortaya çıkmış ve hem akademinin hem de sanayinin dikkatini çekmiştir.

Çağrı merkezlerini etkin bir şekilde yönetmek, özellikle iş gücü maliyetleri ve kapasite planlamasıyla ilgili büyük zorluklar içermektedir. Ballouch ve ark. (2021)'e göre, bir çağrı merkezinin bütçesinin %60-80'i iş gücü maliyetlerine ve kapasite planlamasına ayrılmaktadır. Kapasite planlaması, beklenen iş yüküne dayalı olarak hedefleri karşılayacak optimum ajan sayısını belirlemeyi içerir. Bu iş yükü büyük ölçüde çağrı gelişlerinden—yani bir çağrı merkezinin aldığı çağrı sayısından—kaynaklanır. Çağrı merkezi hizmetlerine olan talep arttıkça, çağrı geliş modellerini doğru tahmin etmek, kaynak tahsisi, personel kararları ve hizmet seviyesi planlaması için kritik hale gelir ve bu da operasyonel verimliliğin artmasına katkı sağlar. Ne yazık ki, birçok çağrı merkezi hala iş yükü tahmini için denetçilerin deneyimlerine dayanmaktadır (Kumwilaisak ve ark., 2022). Bu insan yargısına dayalı yaklaşım, yanlış tahminler, aşırı yüklenen denetçiler ve insan hatalarının artma olasılığı gibi sorunlara yol açar. Bir çağrı merkezi birden fazla hizmeti ve çok sayıda ajanı yönettiğinde, denetçiler aşırı yük altında kalabilir ve bu da verimsizlikleri daha da artırır.

Çağrı merkezi operasyonlarının yönetimindeki pratik sınırlamalar, yalnızca insan yargısına dayanmadan çağrı tahmini yapmak için çeşitli modellerin geliştirilmesine yol açmıştır. Son on yılda, literatürde geleneksel istatistiksel yöntemler, makine öğrenimi (ML) ve derin öğrenme (DL) tabanlı birçok çağrı geliş tahmini modeli önerilmiştir. Ancak, bildiğimiz kadarıyla, çağrı gelişlerinin en iyi model parametrelerini ve temel belirleyicilerini belirlemek için hiperparametre optimizasyonu ve özellik seçimi algoritmalarının birleşimini sistematik olarak değerlendiren bir çalışma bulunmamaktadır. Bu yaklaşım, daha doğru ve güvenilir tahmin modelleri oluşturmayı hedefleyerek çağrı merkezlerinin verimliliğini ve kaynak tahsisini iyileştirmeyi amaçlamaktadır.

Bu tezin amacı, çağrı tahmininde kullanılan çeşitli modellerin performansına özellik seçimi ve Bayes optimizasyon tekniklerinin entegrasyonunun etkisini araştırmaktır. Veri setleri, zaman serisi verilerinden giriş-çıkış çiftleri (X, y) oluşturmak için sequence oluşturma fonksiyonuyla zaman serisi problemine dönüştürülmüştür. Otokorelasyon fonksiyonları, tekrarlayan kalıpları belirlemek ve anlamlı gecikme değerlerine dayalı olarak en uygun zaman gecikmesi/sequance seçimini bilgilendirmek için kullanılmıştır. Farklı modeller için, Bayes optimizasyonu (BO) algoritmasına farklı hiperparametreler sunulmuş ve en uygun hiperparametreleri belirlemek için 10 kez yineleme yapılmıştır. Bu optimum hiperparametreler, veri setlerinin tüm özellikleri ile test edilmiş ve tatmin edici sonuçlar elde edilene kadar süreç tekrarlanmıştır. Random Forest (RF) özellik seçimi kullanılarak önemli özellikler belirlenmiş ve Pearson korelasyon analizi, gereksiz özellikleri elemek ve en etkili özellikleri korumak için kullanılmıştır. Bu seçilen özellikler, BO ile optimize edilmiş modellere aktarılmıştır. Model geliştirme sürecinde, Rastgele Orman (RF), Destek Vektör Makineleri (SVM), Çok Katmanlı Algılayıcı (MLP), Evrişimli Sinir Ağı (CNN), Uzun Kısa Süreli Bellek (LSTM) ve Tekrarlayan Sinir Ağı (RNN) gibi çeşitli modeller kullanılmıştır. Araştırma, günlük, saatlik ve yarım saatlik veri kümeleri üzerinde hem standart hem de optimize edilmiş modellerin performansını, Ortalama Mutlak Hata (MAE) metriği ve Kök Ortalama Kare Hata (RMSE) kullanarak değerlendirmiştir. Toplamda 18 sütun içeren veri kümelerinin her birinde ortak tahmin değişkenleri yer almıştır. Özellikle, günlük veri seti 1400 giriş, saatlik veri seti 69073 giriş ve yarım saatlik veri seti 138145 girişten oluşmaktadır.

Üç veri seti üzerinde RF özellik seçimi algoritmalarının uygulanmasıyla, her tahmin değişkeninin önemi hesaplanmış ve önem sırasına göre artan sırayla sıralanmıştır. Pearson korelasyon analizi, özellikler arasındaki ilişkileri, hem pozitif hem de negatif korelasyonları incelemiştir. Tahmin modellerine dahil edilecek özellikleri önceliklendirirken, yüksek etki derecesi ve hedef değişkenlerle olan güçlü pozitif korelasyonlar gibi faktörler dikkate alınmıştır.

Bu tezin sonuçları, literatürde optimal gecikme süreleri ve gradient boosting gibi diğer özellik seçimi yöntemlerini kullanan modellere kıyasla rapor edilen Ortalama Mutlak Hata (MAE) değerlerinin oldukça daha doğru olduğunu ortaya koymaktadır. Bayes optimizasyonu ile optimize edilen modeller, tüm veri setlerinde temel modellere kıyasla sürekli olarak daha iyi performans göstermiş ve tahmin doğruluğunda önemli iyileştirmeler sergilemiştir. Özellikle, CNN ve LSTM gibi derin öğrenme modelleri, Bayes optimizasyonuna karşı güçlü bir yanıt vermiştir. MLP ve RF gibi makine öğrenimi modelleri de güçlü temel performans göstermiş ve Bayes optimizasyonunun entegrasyonundan önemli ölçüde fayda sağlamıştır. Veri setine bağlı olarak, makine öğrenimi modelleri için Bayes optimizasyonundan sağlanan performans artışı %12 ila %83, derin öğrenme modelleri için %40 ila %96 arasında değişmiştir. Tek değişkenli modellerde, makine öğrenimi modelleri için iyileşme oranı %20 ila %71, derin öğrenme modelleri için ise %50 ila %96.6 arasında değişmiştir.

Özellik seçimi öncesinde, BO optimizasyonundan sonra tüm veri seti değişkenleri kullanıldığında, MLP tüm modeller arasında en düşük MAE değerine sahipken, RF onu takip etmiş ve SVR düşük performans göstermiştir. Özellikle LSTM, CNN, RF ve MLP modelleri için önemli performans iyileştirmeleri elde edilmiş ve bu modeller rekabetçi sonuçlara ulaşmıştır. Çok değişkenli SVR modelleri, çağrı gelişlerini tahmin etmede en düşük performansı sergilemiştir. Temel modellerden seçilmiş özelliklere sahip modellere geçiş, hata oranlarını önemli ölçüde azaltmış ve ML modelleri için %14 ila %88, DL modelleri için ise %92 ila %97 arasında hata oranı azalması sağlamıştır. Özellik seçimi sayesinde, çok değişkenli modeller genel olarak tek değişkenli modellere göre daha iyi performans göstermiştir. Çalışma ayrıca, daha kısa zaman aralıklarının daha doğru tahminler sağladığını ve saatlik ve yarım saatlik aralıkların daha tutarlı tahminler sunduğunu vurgulamıştır.

Genel olarak, MLP tabanlı tahmin modelleri, kullanılan veri setlerinden bağımsız olarak ML yöntemleri arasında daha iyi performans (yani, daha düşük MAE) sergilemiştir ve RF bu performansı izlemiştir. DL modelleri arasında ise LSTM ve CNN en düşük MAE değerine sahip olmuş, RNN bu modelleri takip etmiştir.

Her modelin optimal parametreleri, her model için çeşitli hiperparametreler göz önünde bulundurularak Bayes optimizasyonu aracılığıyla belirlenmiştir. Uzun Kısa Süreli Bellek (LSTM) modeli için keşfedilen hiperparametreler arasında 0.0001, 0.001, 0.002, 0.003, 0.01 ve 0.02 öğrenme oranları; relu, tanh ve linear aktivasyon fonksiyonları; Adam, RMSprop ve SGD optimizatörleri; ve 50 ila 1000 arasında değişen nöron sayıları bulunmaktadır. Optimal parametreler, 0.003 öğrenme oranı, relu aktivasyon fonksiyonu, Adam optimizatörü ve 850 nöron olarak belirlenmiştir.

Evrişimli Sinir Ağı (CNN) modeli için incelenen hiperparametreler, 1. katman için 32'den 512'ye kadar (32'lik adımlarla) nöronlar, 2. katman için 32'den 512'ye kadar (32'lik adımlarla) nöronlar, kernel boyutu (3-5 arası), havuz boyutu (2-8 arası), 0.0001, 0.001, 0.002, 0.003, 0.01 ve 0.02 öğrenme oranları; relu, tanh ve linear aktivasyon fonksiyonları; ve Adam, RMSprop ve SGD optimizatörlerini içermektedir. Optimal parametreler, 1. katman için 128 nöron, 2. katman için 64 nöron, kernel boyutu 5, havuz boyutu 6, 0.003 öğrenme oranı, tanh aktivasyon fonksiyonu ve Adam optimizatörü olarak belirlenmiştir.

Random Forest (RF) modeli için değerlendirilen hiperparametreler, n_estimators (20, 50, 100, 150), max_depth (None, 10, 20), min_samples_split (2, 5, 10) ve min_samples_leaf (1, 2, 4) olarak sıralanmıştır. Optimal parametreler, 20 n_estimators, 10 max_depth, 2 min_samples_split ve 1 min_samples_leaf olarak bulunmuştur.

Destek Vektör Makineleri (SVM) için keşfedilen hiperparametreler, çekirdek türleri (linear, rbf, poly), C değerleri (10, 20, 60, 80, 100, 120, 150) ve epsilon değerleri (0.02, 0.04, 0.06, 0.08, 0.5) olmuştur. Optimal parametreler, rbf çekirdek, 100 C değeri ve 0.06 epsilon değeri olarak belirlenmiştir.

Çok Katmanlı Algılayıcı (MLP) modeli için hiperparametreler, yoğun katman boyutu (32'den 512'ye kadar 32'lik adımlarla), öğrenme oranları (0.0001, 0.001, 0.002, 0.003, 0.01, 0.02), aktivasyon fonksiyonları (relu, tanh, linear) ve optimizatörler (Adam, RMSprop, SGD) olarak sıralanmıştır. Optimal parametreler, 128 nöronlu yoğun katman, 0.003 öğrenme oranı, relu aktivasyon fonksiyonu ve Adam optimizatörü olarak belirlenmiştir.

Son olarak, Tekrarlayan Sinir Ağı (RNN) modeli için değerlendirilen hiperparametreler, yoğun katman boyutu (32'den 512'ye kadar 32'lik adımlarla), öğrenme oranları (0.0001, 0.001, 0.002, 0.003, 0.01, 0.02), aktivasyon fonksiyonları (relu, tanh, linear) ve optimizatörler (Adam, RMSprop, SGD) olarak sıralanmıştır. Optimal parametreler, 128 nöronlu yoğun katman, 0.003 öğrenme oranı, relu aktivasyon fonksiyonu ve Adam optimizatörü olarak belirlenmiştir.

Tahmin değişkenleri arasında CallCount, TalkCount ve TalkTimeSec, çağrı hacmini tahmin etmek için birincil belirleyiciler olarak öne çıkmıştır. Ortalama İşlem Süresi (AHT) tahmininde ise Acht, TalkTimeSec ve Answered30PSec gibi özellikler önemli faktörler olarak belirlenmiştir. Bu değişkenler, çağrı tahmin performansını önemli ölçüde artırmıştır.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Dr. M. Fatih AKAY, for his exceptional guidance, constant support, and insightful feedback throughout the course of my PhD journey. His expertise and mentorship have been crucial in shaping my research and academic growth.

I would also like to extend my sincere thanks to my committee members, Prof. Dr. Mehmet Uğraş CUMA, Prof. Dr. Ibrahim Taner OKUMUŞ, Doç.Dr. Serkan KARTAL and Doç. Dr. Fatih KILIÇ for their valuable insights and thoughtful critiques, which have greatly contributed to the development and refinement of this thesis.

Lastly, my deepest gratitude goes to my father, Dr. Kasauli MZ, my mother, Ms. Aisha Nabanjala, my lovely husband, Muniir Idriis, and my children. Their patience, unwavering support, and encouragement have been my strength throughout this journey. I am also grateful to my siblings, especially Dr. Rashidah Kasauli and Dr. Yasin Nsubuga, as well as friends Safina, Yasin, Sacho, and Mumbejja, who have been great company along the way.

LIST OF TABLES

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

ACHT : Average Call Handling Time
AHL : Agent Heterogeneity and Learning Effect model
AHT :Average Handling Time
BME1 : Bivariate Mixed-Effects
BME2 : Bivariate Mixed-Effects
CNN : Convolutional neural network
DL : Deep Learning
DLMs : Dynamic Linear Models
DNN : Deep Neural Network
ED : Euclidean Distance functions
ESN : Echo State Network
GBDT : Gradient Boosting Decision Trees
GBR : Gradient Boosting Regression
HA : Historical Average
HW : Holt Winters
HWT : Holt-Winters Triple exponential smoothing
KNN : K-Nearest Neighbors
LSTM : Long-Short Term Memory
MAE : Mean Absolute Error
ME : Univariate Mixed-Effects
MHF : MEDIC Hourly Forecasting
ML : Machine Learning
MLP : Multilayer Perceptrons
MU : Multiplicative Univariate
OPT-RF : Optimized Random Forest
PD : Pearson correlation distance function
RF : Random Forest
RMAE : Relative Mean Absolute Error
RMSE : Root Mean Squared Error
RNN : Recurrent Neural Networks
RPS : Ranked probability score
RPS : Ranked probability score
SLFN : Single Layer Feedforward Networks
SMA : Simple Moving Average Model
SN : Seasonal Naïve

SVD **:** Singular Value decomposition

SVM **:** Support Vector Machine

1. INTRODUCTION

Call centers play a pivotal role in modern business operations, serving as the forefront for customer interactions and support services across a diverse array of industries (Gans et al., 2003;Ahmed et al., 2024; Shen & Huang, 2008b). Strategically deployed to cater to the dynamic needs of diverse clientele, they act as hubs for communication where trained agents provide assistance, resolve issues, and facilitate transactions through multiple channels (Weinberg et al., 2007;(Chacón et al., 2023b; Ibrahim et al., 2016a; Kadioglu & Alatas, 2023).

With the increasing complexity and dynamic nature of call center operations, call arrival prediction has emerged as a significant research area, attracting attention from both academia and industry. Managing call centers effectively involves addressing significant challenges, particularly related to labor costs and capacity planning. According to Ballouch et al. (2021), 60-80% of a call center's budget is allocated to labor costs and capacity planning**.** Capacity planning involves determining the optimal number of agents needed to meet targets based on the expected workload. This workload is primarily driven by call arrivals - the number of calls a call center receives. As demand for call center services grows, accurately predicting call arrival patterns becomes crucial for effective resource allocation, staffing decisions, and service level planning, ultimately contributing to improved operational efficiency and customer satisfaction. Unfortunately, many call centers still rely on supervisors' experience for workload forecasting (Kumwilaisak et al., 2022). This reliance on human judgment leads to several issues, including inaccurate forecasting, overloaded supervisors,overloaded agents and increased likelihood of human errors. When a call center manages multiple services and numerous agents, supervisors can become overwhelmed, further worsening inefficiencies.

The practical limitations of managing call center operations have driven the development of various models for predicting call arrivals, rather than relying solely on human judgment. In recent years, researchers have explored various methodologies and techniques to enhance call arrival prediction accuracy. These approaches included classical time series methods (Gans et al., 2003; Bhulai et al., 2008; Bastianin et al.,2012), neural networks and machine learning (ML) algorithms(Ibrahim et al., 2016; Martin et al.,2021; Kanthanathan et al., 2020, Manno et al., 2023; Chacón et al., 2023) and their hybridization (Baskaran et al.,2023; D. K. Barrow, 2016). Studies have also investigated the integration optimization techniques(BALLOUCH et al., 2021) and feature selection methods (Martin et al., 2021).

Despite the potential of ML algorithms for precise forecasts, (Albrecht et al., 2021) noted their underutilization in call center forecasting. Ibrahim et al. (2016) and Manno et al. (2023) also emphasized the value of integrating Deep Learning (DL) methods for enhanced prediction accuracy in data-rich environments. However, DL models while effective with complex data pose challenges in tuning due to their intricate structures (Victoria & Maragatham, 2021).

A comprehensive review of the literature reveals that prediction accuracy is significantly influenced by the choice of models, input features, and the specific optimization techniques applied. Improved accuracy can be achieved by carefully selecting prediction models, optimizing their parameters, and using feature selection methods that highlight key predictors. The combination of these approaches supports better resource allocation and operational efficiency in call centers, making them more adaptable to varying conditions. A detailed survey of prior studies on call arrival prediction and their key findings is available in Chapter 2.

To date, there remains a significant gap in the comprehensive exploration of more effective predictive models and the application of advanced optimization techniques specifically designed for call arrival forecasting. This includes the systematic evaluation that combines hyperparameter optimization and feature selection algorithms to identify optimal model parameters and key predictors.

One promising optimization technique is Bayesian Optimizationc(BO). It is particularly well-suited for optimizing complex and costly functions, relying on a probabilistic surrogate model, often a Gaussian process, to explore the parameter space intelligently (Turner et al., 2020). By systematically exploring hyperparameters and leveraging probabilistic models to guide the search towards promising regions, BO efficiently identifies optimal solutions thus enhancing model performance and predictive accuracy (Garnett ,2023).

Additionally, Feature selection methods are commonly employed in related studies (Niu et al., 2020; Peng et al., 2005) to identify the most relevant variables that contribute to accurate predictions (Martin et al., 2021). However, their application in call arrival forecasting to enhance accuracy remains largely unexplored.

In light of these considerations, this study aims to investigate the impact of integration of feature selection and BO techniques on the performance of various models employed in call arrival prediction. Utilizing models such as Random Forest (RF), Support Vector Machines (SVM), and Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM) and Recurrent Neural Network (RNN), this research comprehensively evaluates the performance of standard and optimized models for short-term call arrival forecasting across daily, hourly, and half-hourly datasets, utilizing Mean Absolute Error (MAE) as the primary performance metric. Additionally, Root Mean Squared Error (RMSE) values are provided in the tables for reference, offering supplementary insights into model reliability and performance variations. Relevant features were extracted through Random Forest’s Mean Decrease Impurity (MDI) and Pearson correlation analysis, while autocorrelation analysis uncovered temporal dependencies and estimate significant lag values. This study also investigates the impact of feature selection on the predictive accuracy of optimized models, aiming to demonstrate the superiority of the proposed methodologies and contribute to advancements in call center management practices and operational efficiency. Insights gained from this research will empower organizations to make informed decisions on resource planning and operational optimization. Notably, this work represents a pioneering effort in providing such insights into the effectiveness of these approaches compared to baseline methods

The thesis contributes significantly to the field of call arrival forecasting:

1. Novelty in Model Development: This thesis is the first to develop call arrival prediction models that integrate BO and feature selection algorithms with ML methods. It employs a diverse array of ML and DL algorithms, including MLP, SVM, RF, RNN, LSTM, and CNN, effectively enhancing prediction accuracy through the application of techniques like Random Forest's Mean Decrease Impurity (MDI) and Pearson correlation analysis.
2. Diverse Dataset Evaluation: Extensive experiments were conducted using a variety of datasets, including daily, hourly, and half-hourly intervals. This rigorous testing ensures the reliability and generalizability of the results.
3. Hyperparameter Optimization: This research uniquely focuses on optimizing hyperparameters for ML and DL models specifically in the context of call arrival prediction. It provides insights into the optimal settings for each algorithm, showcasing how these adjustments enhance predictive performance.
4. Identification of Key Predictors: The study emphasizes the identification of key predictors that significantly impact call arrivals. This focus enhances the reproducibility and reliability of the models by pinpointing critical variables for effective forecasting.
5. Performance Metrics for Comparison: The thesis presents comprehensive results using performance metrics such as MAE and RMSE, facilitating comparisons with prior studies and validating the effectiveness of the proposed models.
6. Enhanced Temporal Granularity: The predictions extend beyond daily intervals to include hourly and half-hourly forecasts, projecting up to seven time steps ahead. These advancements lead to a more accurate and reliable prediction model, ultimately improving call center efficiency and resource allocation.

This study is structured as follows: Section 2 presents preliminary work, section 3 provides description of the materials and methods. Section 4 presents the results and discussions. Section 5 concludes the study.

2. LITERATURE REVIEW

This section reviews relevant studies across various domains, including machine learning, deep learning, call arrival forecasting, feature selection, and hyperparameter optimization. It begins with the historical evolution and advancements in time series forecasting, particularly in call arrival prediction, followed by an in-depth exploration of related methodologies. This comprehensive examination highlights state-of-the-art methods and insights, offering valuable context for our research. It then focuses on techniques similar to those proposed in this study, emphasizing the adaptability and superior performance of optimized DL and ML models.

Historically, traditional time series models found extensive application across various fields, including call center management (Gans et al., 2003; Shen & Huang, 2008). Majorly used classical time series methods included Holt–Winters, exponential smoothing, SARIMA and ARIMA. However Hochreiter & Schmidhuber (1997) noted the superiority of deep learning methods especially LSTM over classical methods especially ARIMA during call arrival forecasting.

Another notable study by (Gans et al., 2003) reviewed literature on the application of traditional and statistical methods emphasizing the need for sophisticated analysis techniques and statistical tools to better understand call volumes, customer behavior, and operational challenges highlighting interdisciplinary research and future technological advancements to predict call arrival patterns in call centers.

Weinberg et al. (2007) employed Markov chain Monte Carlo sampling algorithm to estimate both the latent states and model parameters in their Bayesian forecasting model (M1) for call center data for predicting within-day arrival rates at a U.S. commercial bank's call center. Other models were compared with their approach i.e. M2, a linear additive model applied to transformed data incorporating day-of-week and time-of-day as covariates and M3, an extension of Model 2, adding interaction terms between day-of-week and time-of-day effects. These results indicated that Model 1 generally outperformed Models 2 and 3 with RMSE of 6.86.

Bastianin et al.(2012) proposed a strategy for selecting time series models for daily call arrival forecasting, emphasizing flexible loss functions, statistical evaluation, and economic assessment. They identified SARMAX-GARCH and approximate Bayesian model averaging (ABMA) with RPS of 415.75 as top performers.

Additionally, S. Bhulai explored the application of nearest neighbour algorithms i.e. Pearson correlation distance PD and Euclidean Distance (ED) functions for predicting call center traffic. The main findings of the study are as follows: Firstly, the algorithm does not require a predefined model for arrival rates and can be directly applied to historical data without pre-processing. Secondly, it provides a more accurate forecast compared to conventional methods that utilize simple averages. Thirdly, the nearest neighbour algorithm, particularly when employing the Pearson correlation distance function with an optimized service level of 0.869, can effectively account for correlation structures commonly present in call center data. Lastly, numerical experiments demonstrated that this algorithm yields smaller forecasting errors and improves staffing levels in call centers. Overall, the study contributed to the field by introducing a flexible workforce management tool capable of adapting to varying call volumes, thereby playing a crucial role in maintaining customer satisfaction and effectively managing operational costs(Bhulai, 2015).

In the study by Ibrahim et al. (2016) ,the Univariate Mixed-Effects (ME) model was found to produce the most accurate point forecasts in most scenarios for both call types evaluated. Specifically, the ME model had the lowest Root Mean Squared Error (RMSE) for one-day-ahead (16.31), one-week-ahead (17.95), and two-week-ahead forecasts(20.55) . The coverage probability of the 95% prediction intervals also generally favored the ME model for shorter lead times, although the BME1 and BME2 models had better coverage probabilities for one-day and one-week lead times, and the MU model had better coverage probability for two-week lead times.

Moreover, other studies indicate a dynamic research landscape with the use of ML and DL in forecasting time series. Research advancements in stock price prediction; (Abdoli et al., 2020; Ebrahimpour et al., 2011; Gans et al., 2003; Habtemariam et al., 2023; Khashei & Hajirahimi, 2019; Samal & Kumar, 2023; Shen & Huang, 2008a; Torres et al., 2021), time series analysis; (Xu et al., 2019;Abut et al., 2016; Bastianin et al., 2012; Chakrabarty, 2020; Kervanci & Akay, 2023; Kumwilaisak et al., 2022; Liao et al., 2018; Scheidt & Chung, 2019),tourism (Acbbasimehr et al., 2020; Chakrabarty, 2020; C. Li et al., 2020; Yu et al., 2021), solar energy (Habtemariam et al., 2023; Ma et al., 2020; Niu et al., 2020), health (Abut et al., 2016; Martin et al., 2021; Scheidt & Chung, 2019; Sprock & McGinnis, 2014, 2014; Yu et al., 2021; Zeroual et al., 2020), pollution (Freeman et al., 2018; Qin et al., 2017).

Hybrid architectures have been also used in this research field, for instance, a study conducted by Baskaran et al.,(2023) introduced a novel hybrid forecasting model that combines Interacted-ARIMA (INTARIMA) and Artificial Neural Network (ANN) models for improved forecasting accuracy. The INTARIMA model outperformed traditional ARIMA when interactions among lagged variables existed, as validated through simulation studies. (Baskaran et al., 2023).

Devon K. Barrow examined the effectiveness of the Seasonal Moving Average (SMA) method for forecasting intraday call arrivals in call centers. It highlights that while SMA is generally outperformed by more complex methods for short to medium-term forecasting, it excels at longer forecast horizons crucial for capacity planning. The study introduces a hybrid approach combining SMA with artificial neural networks (ANNs) to enhance short-term accuracy without compromising long-term performance.(D. K. Barrow, 2016).

However, recent strides in machine learning (ML) and deep learning (DL) have introduced alternative approaches boasting potentially heightened accuracy and flexibility. For example, Torres et al. (2021) provided a comprehensive overview of the most commonly used deep learning architectures for time series data, such as feedforward networks, recurrent neural networks (RNN), gated recurrent units (GRU), long short-term memory (LSTM), bidirectional RNN (BRNN), convolutional neural networks (CNN), and temporal convolutional networks (TCN). They emphasized the importance of setting hyperparameters, selecting appropriate frameworks, and utilizing hardware efficiently to enhance the performance of deep neural networks in time series prediction.

Similarly Andrade & Moazeni (2023) developed an empirical model to analyze features that exert significant impact on the likelihood that a call arrived to the platform would eventually be transferred to a live agent.

Ballouch et al. (2021), incorporated time lags with Multilayer Perceptron (MLP) and Long-Short Term Memory (LSTM) networks to predict call arrivals in a call center. Forecasting was performed for 12, 24, 36, and 48 time steps ahead. The results showed that the MLP-based model achieved MAE values ranging from 1.50 to 13.58, while the LSTM-based model had MAE values between 19.99 and 66.74 (BALLOUCH et al., 2021).

Additionally, Martin et al.(2021) conducted feature selection using an ensemble-based decision tree model and used a multi-layer perceptron (MLP) to generate daily, hourly, and spatially distributed call volume predictions for emergency medical service (EMS). Additionally, K-Means clustering was applied to produce heterogeneous spatial clusters based on call location and associated call volume densities for spatially distributed predictions. The authors compared the predictive performance of the MLP model to traditional time-series forecasting techniques (Holt-Winter's (HW) Triple Exponential Smoothing, ARIMA) and a common industry method (MEDIC Hourly Forecasting (MHF)). Their findings highlighted the superior performance of MLP models over both time-series methods and industry approaches, particularly at finer spatial granularity levels, where precise call volume forecasts are essential

Moreover, Kanthanathan et al. (2020) employed deep learning models in their study. The findings suggest that LSTM ably captured the underlying patterns and trends in the time series data with MAE of 6.6996, MSE 96.4373 and predicted the AHT with MAE of 13.9535, MSE 312.3277. This was in comparision with Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Bi-Directional LSTM (BiLSTM). Though all methods effectively forecasted the overall incoming call count for each service supported by the contact center. This approach empowers administrators and operations managers to anticipate service demand, efficiently manage incoming call peaks, optimize utilization of Customer Service Agents (CSAs), and minimize CSA idle time (Kanthanathan et al., 2020)

Similarly, Albrecht et al. (2021) conducted a comprehensive investigation comparing ML approaches with traditional time series models including Gradient Boosting Decision Trees (GBD), Gradient Boosting Regression (GBR),K-Nearest Neighbors (KNN),Random Forest

(RF),Support Vector Regression (SVR), STL + ARIMA, STL + ETS, STL + RW.Through cross-validation with an expanding rolling window, their study revealed that the random forest (RF) algorithm outperformed other models in predicting call arrival patterns with MAE of RF: 11.7544 for No lead time, 11.8129 for One week, 12.0648 for Two weeks and 12.8134 for Three weeks.This finding underscores the importance of leveraging ML algorithms for accurate forecasting in call centers(Albrecht et al., 2021).

Manno et al. (2023), compared deep and shallow neural networks for predicting incoming call volumes in call centers 24 hours ahead. They included Echo State Network (ESN),Single Layer Feedforward Networks (SLFN) and Seasonal Naïve(SN). SLFN achieved better NRMSE of 0.3691, dMAPE of 16.75% and MMDE 50.65%. the experimentconducted on three real-world datasets, revealed better predictive performance for the shallow approach. Additionally, the shallow strategy demonstrated greater robustness and efficiency, reducing inference time by a factor of 2.5 to 4.5 compared ESN and SN (Manno et al., 2023).

The study by Tartuk et al.(2022) investigated the utilization of the XGBoost algorithm, augmented with consecutive and periodic lookback methods, for predicting call center arrivals. Key findings of the research include the following: The integration of lookback techniques enhances the XGBoost model's forecasting capabilities, surpassing the performance of traditional moving average models.

The study by Chacón et al., (2023a) compares various methods for forecasting call center arrivals, including classical time series techniques such as Holt-Winters, exponential smoothing, and SARIMA, with machine/deep learning approaches like , LSTM, GRU and XGBoost. Real-life call logs from a national US insurance company spanning ten months were used for testing. Short and long-term forecasts were evaluated using half-hourly and daily aggregations. Results show that for short-term forecasts with ample seasonal periods, deep learning models, particularly RNN, reported minimal errors, followed by the boosting approach. However, for long-term forecasts with less than a year's worth of data, boosting outperformed all other models, including deep learning. For Half-hourly intervals, it was LSTM with MAPE of 27.28 and daily interval it was XGBoost with MAPE of 10.63. This suggests that while deep learning models excel in capturing short-term volatility fluctuations, boosting is more effective for long-term forecasts, especially in scenarios with limited seasonal periods. Therefore, the study recommends the use of boosting as a benchmark approach for forecasting call arrivals, particularly for inhomogeneous Poisson processes with constrained seasonal data.

The study by Kiwamu & Goro, (2019) presents compelling findings regarding the efficacy of dynamic linear models (DLMs) in call center operations. Firstly, DLMs are shown to effectively forecast call volumes by accounting for gradual changes and the influence of special days. Secondly, when applied to actual call center data, the DLM approach results in a significant reduction in operator over-arrangement, with reductions of up to 39.3% observed. Thirdly, beyond accurate volume forecasts, DLMs provide probability distributions, aiding in optimal staffing decisions. These findings highlight the value of DLMs as a valuable tool for call centers to optimize staffing levels, ultimately enhancing service quality while managing operational costs efficiently.

Another study by Millán-Ruiz & Hidalgo (2013) created a novel approach based on ANNs (uRprop) to the intra-day forecasting of call arrivals in call centers and compared other forecasting methods i.e. single exponential smoothing, simple time series, stationary time series, damped trend time series, several ARIMA models, regression models, logistic regression and artificial neural networks (seven different learning algorithms). Results showed ANNs outperforming better than the other models with lowest MAE for example 4.533 for call group 1, 3.244 for group 2, 2.378 for group 3 and 0.976 for group 5.

Taylor (2012) focused on density forecasting of intraday call center arrivals using models based on exponential smoothing. The study primarily examines high and low volume series of half-hourly call arrivals at call centers. A new Holt-Winters exponential smoothing (HWT) Poisson count data model was developed with gamma-distributed stochastic arrival rate to enable density forecasting of call arrival volumes and rates for intraday data. The study compares different forecasting methods, including the basic HWT model, HWT with stationary level, ARMA model, and singular value decomposition (SVD-type) approaches, to evaluate forecast accuracy and operational decision-making in setting staffing levels. The results indicate that the HWT models with stationary level perform well up to five days ahead, while the SVD-type approach with exponential smoothing factor models is more accurate beyond this timeframe mean RPS of 60 for US bank data and 21 for NHS data. The research highlighted the importance of accurate density forecasting for call center operations and suggests areas for further research, such as incorporating covariates and developing quantile models for arrival volumes and rates.

In their study, Barrow and Kourentzes investigated the impact of outlying periods on forecasting call center arrivals.They identified artificial neural networks (ANNs) as the most accurate overall, while simpler methods like double seasonal exponential smoothing and seasonal weekly moving averages performed better during outlying periods. Explicitly modeling outlying periods with ANNs led to a significant improvement in accuracy with Relative Mean Absolute Error (RMAE) of 1, with gains exceeding 5 percent compared to other methods. The authors advocated for the use of multiple binary dummies, emphasizing the effectiveness of ANN Bin S-1 over ANN BinS due to its simpler structure and slightly superior accuracy. However, they noted the necessity for further research, suggesting the development of methods for automatically identifying and validating outlying patterns in call center data. Furthermore, they proposed future investigations into density forecast models for call center arrival rates, integrating their findings on outlier modeling to enhance forecasting accuracy and support staffing decisions effectively (D. Barrow & Kourentzes, 2018).

Kumwilaisak et al. (2022) utilized deep neural networks, and reinforcement learning (RL), to predict traffic data in a call center. Specifically combined Long-Short Term Memory (LSTM) and Deep Neural Network (DNN) to model and forecast call center traffic and attained an MAE of 0.832, Erlang-A Model to predicted traffic parameters to calculate essential service metrics, such as call abandonment probability and average response time and Reinforcement Learning (RL) to implement Q-learning to dynamically manage workforce allocation. This was evaluated based on various metrics such as Service Level Agreement (SLA) and Average Speed of Answer (ASA).

Furthermore, Surasai & Sa-ing (2023)investigated the effectiveness of sophisticated time series models (SARIMAX, SVR, Gradient Boosting, RNN, GRU, LSTM) in forecasting call center contact volume, a critical aspect of successful Workforce Management implementation. SVR achieved the lowest MAE of 25.13, MAPE of 6.15%, RMSE of 34.66, and highest R^2 of 90.56% in the selected feature group. The study emphasized the importance of algorithm selection and feature engineering in optimizing time series forecasting models for workforce management in call centers.

Shen & Huang (2008) developed methods for accurately forecasting call arrivals in call centers both on a daily (interday) and within-day (intraday) basis using Historical Average (HA),Additive (ADD),Multiplicative (MUL),Time Series (TS) Models i.e Multiple variants including ARIMA models and Bayesian Gaussian (BG) specifically used for intraday updating. For Interday, TS4 had the lowest mean RMSE 56.88, intraday Bayesian Gaussian (BG) and Time Series (TS) methods provided the most accurate updates with mean RMSE of Mean 18.28 and 18.16 respectively. These statistics suggest that the TS methods generally outperformed the HA method.

In call center forecasting, Kadioglu & Alatas (2023) used a two-layered approach combining integer programming techniques and LSTM model to optimize the allocation of available staff resources according to the forecasted workload. LSTM model performed reasonably well in capturing the underlying patterns and trends in the time series data, yielding predictions that were relatively close to the true values whereas integer programming techniques were used to optimize the allocation of staff based on the forecasted workload. The call volume prediction score was Mean Squared Error (MSE) of 96.4373 and Mean Absolute Error (MAE) of 6.6996. For AHT prediction the score was MSE of 312.3277 and MAE of 13.9535.

Another study proposed a data-driven method to solve the call center service level prediction problem. in some cases the AHT prediction in advance was necessary incase of the Erlang-C method. By using feature extraction based on empirical analysis, they proposed the use of decision tree based ensemble methods, like random forest and GBDT, to model the relationship between service level and input features which tremendously improved the prediction accuracy. GBDT and Xgboost models provided the best predictive accuracy for call center service levels, outperforming traditional methods and other machine learning models. With GBDT, without date features, the MAE was 7.16, MAPE = 9.74% and with date features the MAE was 6.81% and MAPE of 9.11%. As for Xgboost, without date features, the MAE was 6.95% and MAPE of 9.57% and with date features, the MAE was 6.85% with MAPE of 9.26% (Hou et al., 2022).

Another study focused on Linear Regression and Simple Moving Average Models (SMA) for forecasting Average Handling Time (AHT) in call centers. AHT is a critical metric in call center performance as it reflects the average duration of customer interactions, including call duration and any subsequent administrative work. SMA The achieved a performance improvement of MAE 6.81% and standard deviation of 0.53%, indicating a consistent and significant enhancement over the baseline (Fu et al., n.d.,2019)

Ding (2016) evaluated multiple models for their accuracy in predicting AHT in terms of service levels (SL1), abandonment rates (r), and average speed of answer (ASA).The models that incorporated agent heterogeneity and learning effects showed significantly improved accuracy. The best-performing model (Model 3) assumed an Inhomogeneous Poisson Process (IPP) for call arrivals, an exponential distribution for handling times, and incorporated empirical data for other factors. This model had the lowest error rates of Weighted Absolute Error for Service Level (WAESL1) of 0.12%, WAEr of 0.743% and WAEASA of 1.73.

Table 2.1. Overview of Recent Studies Predicting Call Arrivals and Their MAE and RMSE

Year	Study	Method	Optimal method	MAE	RMSE
2007	Weinberg et al. (2007)	BMM, MCMC	BMM,MCMC	_	6.86.
2008	Shen & Huang (2008)	HA, ADD,MUL, TS, BG	TS	_	56.88
2013	Millán-Ruiz & Hidalgo(2013)	ANNs, ARIMA, regression models	ANNs	0.976-4.533	_
2015	Bhulai (2015)	NNA with PD And ED functions.	NNA with PD	0.869	_
2016	D. K. Barrow(2016)	ANNs, ARIMA, ETS, seasonal NAÏVE, SMA	ANNs	1.00	_
2019	Kiwamu & Goro (2019)	DLMs	DLMs	_	
2021	Albrecht et al. (2021)	GBD, GBR,KNN,RF,SVR, STL + ARIMA, STL + ETS, STL + RW	RF	11.75 - 12.81	_
2021	BALLOUCH et al. (2021)	MLP, LSTM	MLP	1.50-13.58	_
2022	Tartuk et al. (2022)	XGBoost, Moving average	XGBoost	4.11- 8.39	_
2023	Surasai & Sa-ing (2023)	SARIMAX, SVR, LightGBM, GBR, XGBoost, Simple RNN, GRU,LSTM	SVR	25.13	_
2023	Kadioglu & Alatas (2023)	Integer programming, LSTM	Integer programming, LSTM	6.69	_
2022	Kumwilaisa et al. (2022)	Combined approach DNN & LSTM, Erlang - A , RL	Combined approach	_	0.832
2023	Surasai & Sa-ing (2023)	SARIMAX, SVR, LightGBM, GBR, XGBoost, Simple RNN,GRU, LSTM	SVR	_	34.66

Table 2.2. Overview of Recent Studies Predicting Call Arrivals showing various performance metrics

Year	Study	Method	Optimal method	MAPE	RPS	NRMSE	RMAE
2018	Barrow & Kourentzes (2018)	ANNs, Naïve, MA, ETS, ARIMA	ANNs	–	–	–	1
2012	Taylor (2012)	HWT, Poisson count data model, ARMA, SVD	SVD-type approach	–	21- 60	–	–
2012	Bastianin et al. (2012)	ARMAX , SARMAX , SARMAX-GARCH	SARMAX-GARCH	_	415.75.	–	–
2021	Martin et al. (2021)	MLP and Ensemble-based decision tree model ,HWT, ARIMA, MHF	MLP	5.98%,	–	–	–
2023	Chacón et al. (2023a)	HW, SARIMA, XGBoost, LSTM, GRU	LSTM XGBoost	27.28 10.63	–	–	–
2023	Manno et al. (2023)	ESN, SLFN, SN	SLFN	–	–	0.3691	–

Table 2.3. Overview of related studies for AHT predictions showing various performance metrics

Year	Study	Method	Optimal method	RMSE	MAE	WAESL1
2013	Millán-Ruiz & Hidalgo(2013)	ANNs, ARIMA, regression models	ANNs		0.976-4.533	
2016	Ibrahim et al. (2016)	MU, ME, BME1, BME2	ME	16.31-20.55	–	–
2016	Ding (2016)	Log-Normal Distribution, AHL,Daily AHT Model	AHL	–	–	0.12%
2019	Li et al. (2019)	GBDT	GBDT	–	0.013 - 0.035	–
2019	Fu et al.(2019)	Linear Regression Models, SMA	SMA	–	6.81%	–
2020	Kanthanathan et al. (2020)	RNN, LSTM, GRU, BiLSTM)	LSTM	–	6.69-13.95	–
2022	Hou et al. (2022a)	Decision trees, RF, GBDT	GBDT and Xgboost	–	6.81-7.16	–
2023	Kadioglu & Alatas (2023)	Integer programming, LSTM	Integer programming, LSTM	–	13.95	–

Meanwhile the studies by Elshewey et al.(2023) and (Kervanci et al., 2024) focused on different applications—classification of Parkinson's disease and Bitcoin price prediction, respectively—their utilization of Bayesian Optimization to optimize hyperparameters in machine learning models holds relevance to our study of call arrival forecasting. Elshewey et al. demonstrated notable enhancements in model accuracy, particularly with Support Vector Machine (SVM), after hyperparameter tuning using Bayesian Optimization. Similarly, Kervancı and Akay's exploration of hyperparameter optimization for Long Short-Term Memory (LSTM) models offers valuable insights into the significance of optimizing model parameters for improved predictive performance. While the specific contexts differ, the methodologies and techniques presented in these studies, including Bayesian Optimization, can be adapted and applied to optimize models for call arrival forecasting, potentially leading to enhanced predictive accuracy in our research.

This literature review details studies on call arrivals and their methodologies, emphasizing the varied approaches and their respective performance metrics. It highlights the efficacy of different methods and underscores the need for advanced modeling and optimization techniques in predictive modeling. While existing studies in call arrival forecasting have compared various forecasting methods, there remains a need for further exploration of a wider range of DL and ML models to address diverse call arrival patterns effectively. Additionally, the underutilization of optimization techniques specifically tailored for call arrival forecasting presents a notable gap. Moreover, while feature selection methods are commonly employed in other studies, their potential application to enhance accuracy in call arrival forecasting remains largely unexplored.

In this thesis, we address these gaps by boosting six time-series models with Bayesian Optimization and feature selection and comparing them with their baseline versions using three different datasets at daily, hourly, and half-hourly intervals. We propose the most appropriate ML and DL models for forecasting call arrivals at different time intervals. By integrating Bayesian Optimization and feature selection, we aim to refine the model selection process, enhance predictive accuracy, and provide a robust framework for effective call arrival forecasting.

3. MATERIAL AND METHOD

This section covers dataset generation, implementation, data preprocessing, exploratory analysis, problem formulation, performance evaluation, and the Bayesian Optimization (BO) and feature selection methods used in the study. It also provides an overview of the ML and DL architectures and the theoretical framework guiding their application

3.1. Materıal

3.1.1. Dataset generation

In this study, we utilized three real datasets collected at intervals of 1 day, 1 hour, and 30 minutes, obtained from a telecom company in Turkey. The datasets comprised call arrival data from January 1, 2018, to October 31, 2021. Temporal features were engineered to enhance the original dataset with additional time-related information, including hour, day, weekday, weekend, month, and quarter. Specifically, the daily dataset consisted of 1400 entries, the hourly dataset contained 69073 entries, and the half-hourly dataset comprised 138145 entries each with 18 columns representing various attributes. The forecasting task aimed to predict both the call volume and average handling time (AHT), as illustrated in Figure 3.1.

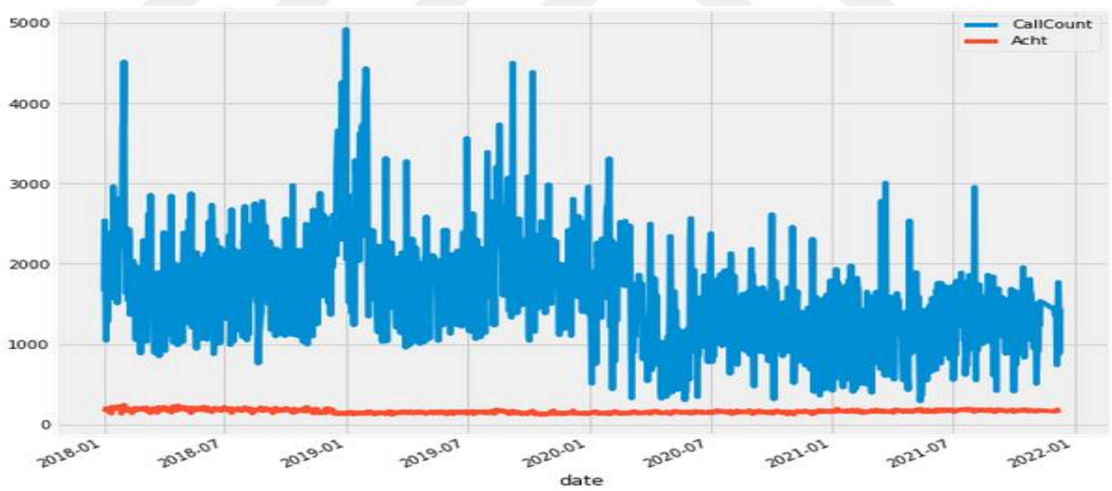

Figure 3.1. Preview of Call Volume and AHT data

3.1.2. Implementation

Implementation of this study was executed in Python (version 3.11.5), utilizing the Keras library (version 2.15.0) with TensorFlow (version 2.15.0) as the backend. The TensorBoard (version 2.15.2) tool was also employed for visualization and monitoring during model training. Keras provided two primary model types: The Sequential model and the Model class. For this study, we employed the Sequential model as the basis for our ML and DL architectures.

3.1.3. Data Pre-processing

In the preprocessing phase, the dataset underwent cleaning, which involved addressing missing values by removing rows containing non-values. Additionally, sorting procedures were applied to enhance the organization and structure of the data. The dataset was split, with the training set excluding the last three weeks for daily observation and the last 1 week for both hourly and half-hourly. Consequently, the testing set incorporated the most recent 21 observations for daily data, 168 and 336 observations for hourly and half-hourly data, respectively. To ensure consistent numerical ranges across sets, Min-Max scaling normalized the data. The data sets were then turned into a time series problem by utilizing the create sequences function to generate input-output pairs (X, y) from the time series data.

3.1.4. Autocorrelation analysis

In this section, the researcher examined the autocorrelation in the data to understand characteristics of data which shows the degree of similarity between the values of the same variables over successive time intervals (Site & Birant, 2019). Autocorrelation analysis was employed for both Call volume and Average handling time predictors. Autocorrelation function defined in Equation 3.1.

$$ACF\ (i) = \frac{\sum_{t=1}^{n-i}(y_t - \bar{y})(y_{t+1} - \bar{y})}{\sum_{t=1}^{n}(y_t - \bar{y})^2}$$

Equation 3.1

Where y_t represents the value of the time series at time t, $\bar{y}$ is the mean of the time series, n is the total number of observations in the time series, i is the lag order.

3.1.5. Problem Formulation

The study utilized the create sequences function to generate input-output pairs (X, y) from the time series data. This function created sequences of call arrivals (X) and their corresponding target values (y) for forecasting. The sequence length parameter determines the number of past observations considered for each input sequence, while the n_steps_ahead parameter sets the number of future steps to predict. The output of the create sequences function is a set of input-output pairs (X, y) suitable for training a forecasting model. X contains sequences of historical Call arrivals, and y contained the corresponding target values representing call arrivals at future time steps as shown in below

Given a time series dataset of Call arrivals $D = \{x_1, x_2, \ldots, x_t\}$
We defined the function create_sequences (D, sequence_length, n_steps_ahead) as follows: create_sequences $(D,\ sequence\_length,\ n\_steps\_ahead) \rightarrow (X, y)$

Where: $X = \{(x\_1, x\_(1 = 1), x\_(1 + sequence\_length - 1)\ ) \mid i = 1,2, \ldots, \mid D \mid -sequence\_length - n\_(steps\_ahead\ ) - +1\}$.
$y = \{(x\_(1 + sequence\_length), x\_(1 + sequence\_lenght + 1\ ), \ldots., x\_(i + sequence\_length + n\_step\_ahead - 1) \mid i = 1,2, \ldots, |D| - sequence\_length - n\_steps\_ahead + 1\}$.

The objective is to develop a forecasting model
$\hat{x}^{t+1} = f\left(x_t, x_{t-1}\, , \ldots .\, x_{t-sequence\_length+1}\right)$ to predict call arrival values at future time steps $t + 1, t + 2, \ldots, t + n\_steps\_ahead$ based on historical observations.

where x_t is the call arrival measurement at time t,X represents input sequences and y is the corresponding target values.

Note: The function f represents the forecasting model to be learned, $sequence\_length$ determines how many past observations are considered for each input sequence and n_steps_ahead specifies the number of steps into the future we want to predict.

3.1.6. Performance Evaluation

The accuracy of the proposed models has been assessed by comparing its predictions against real data using regression metrics. In this study, statistical MAE and RMSE, are employed and defined in Equation 3.2 and 3.3. MAE calculates average absolute differences between predicted and actual values and is less influenced by noisy data. RMSE quantifies the difference between predicted and actual values by squaring the differences, averaging them across all samples, and then taking the square root of the result.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}\,| \qquad \text{Equation 3.2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y_i})^2} \qquad \text{Equation 3.3}$$

- $\boldsymbol{n}$ is the number of observations
- $\boldsymbol{y_i}$ represents the actual (ground truth) value for observation
- $\boldsymbol{\hat{y_i}}$ represents the predicted value for observation
- $|.|$ denotes the absolute value

3.2. Methods

3.2.1. Random Forest

RF is an integrated ML method (Mitchell et al., 1990) that uses the random resampling (bootstrap) technique and node stochastic classification technique to construct multiple unrelated decision trees, and produces the final regression results by voting as shown in Figure 3.2. The RF method has better robustness to noise and missing data, and has a faster learning speed (Niu et al., 2020).

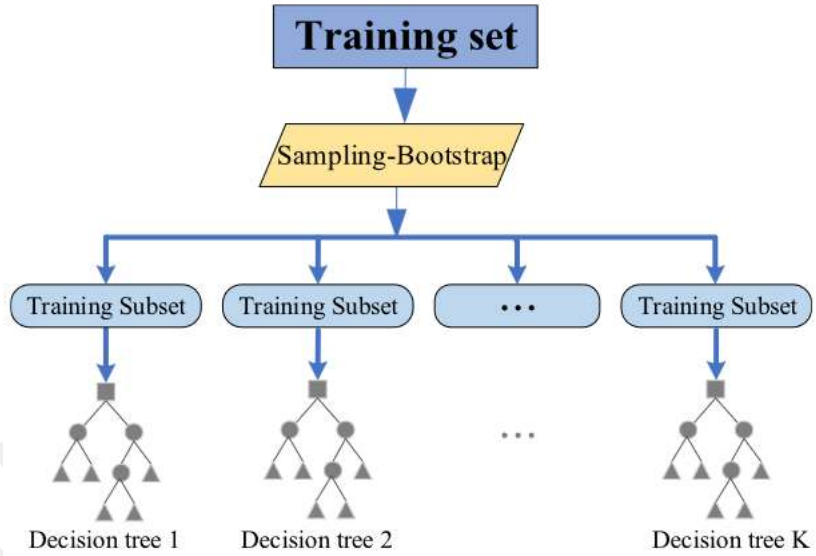

Figure 3.2. Random forest generation process (Niu et al., 2020).

Klusowski (2018) explains that a random forest is a prediction model composed of multiple randomized base regression trees $\{r_m(x, \Theta_m, D_n), m \geq 1\}$, where Θ_1, Θ_2,...are independent and identically distributed outputs of a randomizing variable Θ. These random trees are aggregated to form the overall regression estimate as seen in Equation 3.4.

$$\bar{r}_n(x) = \mathbb{E}_\Theta[r_m(x, \Theta, D_n)] \qquad \text{Equation 3.4}$$

Where x is the input vector, $\mathbb{E}_\Theta$ represents the expectation with respect to the random parameter, conditional on x and the dataset D_n. In practical applications, this expectation is approximated by generating M (usually large) random trees and averaging their predictions shown in Equation 3.5.

$$\bar{r}_n(x) = \frac{1}{M}\sum_{m=1}^{M} r_m\,(x, \Theta_\mathrm{m}, D_n) \qquad \text{Equation 3.5}$$

This method is supported by the law of large numbers. The variable Θ_m introduces randomness by determining how the splits are made in each tree, including the selection of the feature to split and the split point. Each tree is trained on a bootstrapped sample of the dataset and uses a random subset of features at each split, ensuring diversity among the trees. This randomness helps to reduce overfitting and enhances the model's generalization ability.

By using the predictions of multiple trees, the random forest reduces variance and improves the robustness of the model's predictions, making it a powerful ensemble learning method.

3.2.2. Support Vector Regression

Support Vector Regression (SVR) is a regression tool used to analyze the relationship between a continuous dependent variable and one or more predictor variables. Its optimization process is based on support vectors, making its efficiency independent of the input data's dimensionality and instead reliant on the number of support vectors. This characteristic makes SVR particularly effective for handling high-dimensional datasets. Unlike traditional regression methods that rely on specific model assumptions, SVR, trains a model to highlight the significance of variables in defining the relationship between input and output (Zhang & O'Donnell, 2019). However, Cervantes et al. (2020) note that while Support Vector Machines (SVMs), including SVR, have strong generalization abilities and several advantages, they also have notable weaknesses. These include challenges with parameter selection, algorithmic complexity, particularly concerning training time with large datasets, and reduced performance on imbalanced datasets. To enable SVR to manage nonlinear data, a kernel function was introduced that alters the original input data into a higher-dimensional space known as the kernel space (see Figure 3.3).

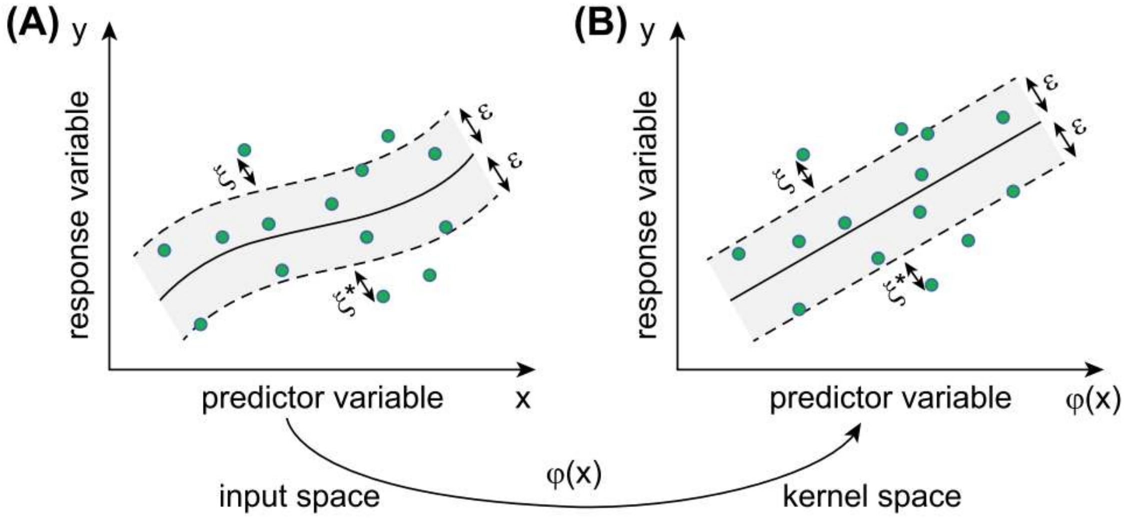

Figure 3.3. A visual depiction of nonlinear ε-SVR.

None linear SVR employs a mapping function, denoted as φ, to convert the data from the input space (A), where linear separation is not feasible, to a higher-dimensional kernel space (B). In this kernel space, the data becomes separable by a linear hyperplane (Zhang & O'Donnell, 2019). In this context, a mapping function is used to convert the input feature R_d (Equation 3.6 and 3.7) into a kernel space F. Employing kernels is a prevalent strategy in SVM (for both regression and classification) because it obviates the necessity of solving for a high-order separating hypersurface in the input space, which is notably complex compared to solving a linear optimization problem in the kernel space.

Given $\varphi(\cdot): R_d \longrightarrow F$, we write linear function $f(x)$ in terms of $\varphi(x)$ in Equation 3.6.

$$y = f(x) = \langle w, \varphi|(x) \rangle + b = w^T \varphi(x) + b$$ Equation 3.6

Then following dual optimization problem is written in Equation 3.7:

$$\max_{\mathrm{a},a^*} -\epsilon \sum_{l}^{l_{sv}} (a_i + a_i^*) + \sum_{l}^{l_{sv}} y_i (a_i + a_i^*) - \frac{1}{2} \sum_{i}^{l_{sv}} \sum_{j}^{l_{sv}} (a_i + a_i^*)$$

Equation 3.7

$$x(a_i + a_i^*) k(x_i, x_j)$$

Where $k(x_i, x_j) = \varphi(x_i)\varphi(x_j)$, sunject to

$$\sum_{l}^{l_{sv}} y_i (a_i + a_i^*) = 0, a_i, a_i^* \epsilon [0, \mathrm{C}]$$

When, $k(\cdot)$ is the kernel function. Then, the expansion of w and f can be written as seen in Equation 3.8 and Equation 3.9.

$$\sum_{l}^{l_{sv}} y_i (a_i + a_i^*) \varphi(x_i)$$ Equation 3.8

and

$$f(x) = \sum_{l}^{l_{sv}} y_i (a_i + a_i^*) k(x_i, x) + b$$ Equation 3.9

According to Awad et al.(2015), various kernel functions are available for use in SVR, each catering to different data distributions. These include linear kernels, polynomial kernels, radial basis function (RBF) kernels, and ANOVA RB kernels. The choice of kernel function depends on the characteristics of the input data. For instance, the linear kernel, being the simplest, is suitable for large sparse data vectors. Polynomial kernels find extensive application in image processing tasks. RBF kernels serve as a versatile choice applicable when prior knowledge about the data distribution is lacking. ANOVA RB kernels are typically employed in regression tasks. For this study we used RBF.

3.2.3. Multilayer Perceptrons

Multi-Layer Perceptron is a feed-forward artificial neural network which consists of interconnected layers: an input layer, one or more hidden layers, and an output layer. Neurons in each layer are connected to those in the next linking weights, allowing information to flow forward. Nonlinear activation functions, such as ReLU in hidden layers and linear in the output layer, enable the network to learn complex patterns. Training involves adjusting connection weights through backpropagation to minimize a specified loss function, like mean squared error.The structure of MLP is illustrated below in Figure 3.4.

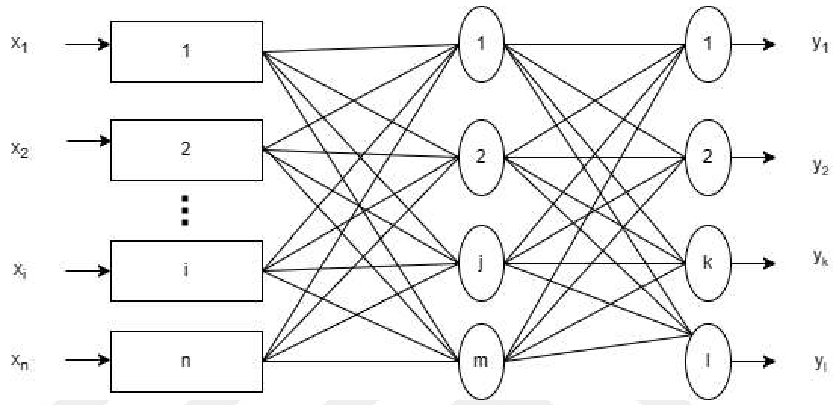

Figure 3.4. Multilayer perceptron (Widiasari et al., 2017)

In Equation each input x_i to a neuron in the hidden layer is multiplied by W_{ji} and a bias term is added. The resulting value, H_j, is then passed through an activation function and subsequently appears in the output layer, as described in Equation 3.10 and 3.11.

$$H_j = f\left(\sum\nolimits_{i=1}^{n} W_{ji}X_i + b_i\right) \qquad \text{Equation 3.10}$$

$$y = f(\sum_{j=1}^{m} W_{kj}H_i + b_o) \qquad \text{Equation 3.11}$$

3.2.4. Long Short-Term Memory Network

The LSTM represents a groundbreaking advancement in artificial intelligence, purposefully designed to overcome the vanishing gradient problem inherent in traditional RNNs. Originally introduced by Hochreiter and Schmidhuber in 1997, LSTMs distinguish themselves by their exceptional ability to process and predict sequences of data. Steering clear of the limitations posed by vanishing gradients, LSTMs introduce a specialized memory cell that adeptly stores and retrieves information across extensive sequences (Habtemariam & Kekeba, 2023). This architectural innovation involves three gates input, forget, and output gates effectively regulating the flow of information through the cell. The input gate controls updates to the memory cell from the input and the forget gate decides which information to keep and which to discard, while the output gate determines the information to pass to the next state. The result is an LSTM network with the capacity to capture and recall long-term dependencies within sequential data, positioning it as a formidable solution where conventional RNNs encounter difficulties(Chacón et al., 2023b; Van Houdt et al., 2020; Wang et al., 2023). The mathematical formulation of the LSTM is shown in Equation 3.12 to Equation 3.17.

$$g_t = \sigma(\omega_f [f_{t-1}, x_t] + b_f) \qquad \text{Equation 3.12}$$

$$j_t = \sigma(\omega_j [f_{t-1}, x_t] + b_i) \qquad \text{Equation 3.13}$$

$$\tilde{Z}_t = tanh(\omega_z X [f_{t-1}, x_t] + b_z) \qquad \text{Equation 3.14}$$

$$Z_t = f_t * Z_{t-1} + j_t * \tilde{Z}_t \qquad \text{Equation 3.15}$$

$$O_t = \sigma(\omega_o X [f_{t-1}, x_t] + b_o) \qquad \text{Equation 3.16}$$

$$f_t = O_t * tanh(Z_t) \qquad \text{Equation 3.17}$$

Where g_t, j_t and $\tilde{Z}_t$ represent the forget gate, the input gate, and the candidate cell state, respectively. O_t denotes the forget gate, Z_t and f_t are the cell output at the current time t; $Z_{t-1}$and f_{t-1} are the cell outputs at the previous time $x_t - 1$ and x_t is the imput to the LSTM cell. w is the weight of neurons and b is the bias for each weight as shown in Figure 3.5.

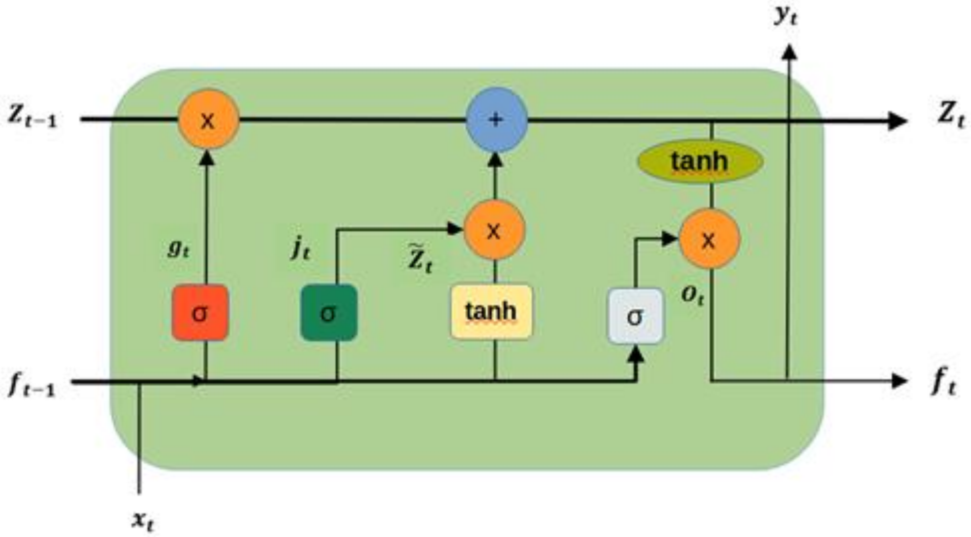

Figure 3.5. LSTM model (Habtemariam et al., 2023)

3.2.5. Convolutional Neural Network

CNNs consist of multiple stages of learning, including convolutional layers, non-linear processing units, and subsampling layers. Their feedforward architecture employs convolutional kernels for various transformations, making them well-suited for tasks such as image processing and text classification (Yasin et al., 2024) .While primarily used in these domains, CNNs excel in pattern recognition and can leverage spatial or temporal correlations in data, making them applicable for forecasting tasks. In time series forecasting, CNNs train filters to detect recurring patterns in data and use these patterns to predict future values. Their layered structure allows for effective noise reduction, progressively filtering out irrelevant information while retaining essential patterns (Borovykh & Bohte, 2018). The architecture includes mechanisms like max-pooling to extract maximum values from sliding windows, helping to reduce overfitting. A flattened layer is often placed between the max-pooling and dense layers for efficient data processing. For instance, this study presents a 1D CNN model with three one-dimensional convolutional layers, each containing 128 filters, as illustrated in Figure 3.6.

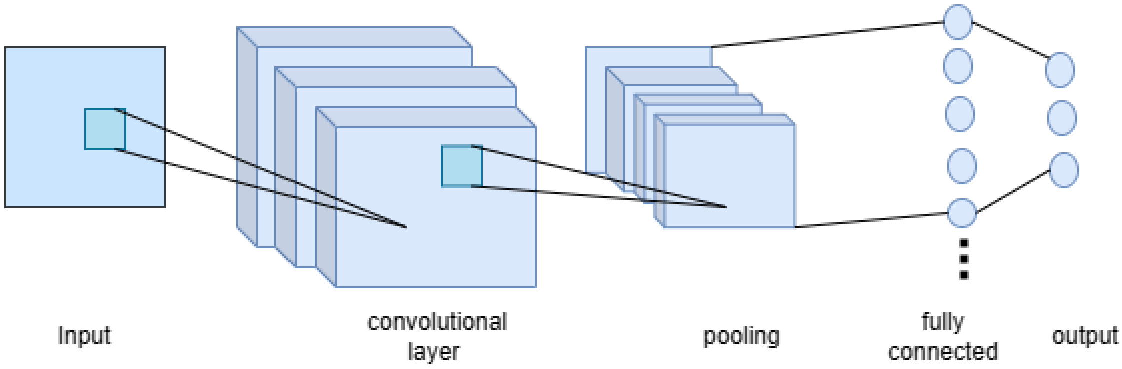

Figure 3.6. CNN architecture (Albelwi & Mahmood, 2017).

3.2.6. Recurrent Neural Network

RNNs are specialized neural networks designed to handle sequential data, such as sequences of words in machine translation, audio data in speech recognition, or time series in forecasting problems(Torres et al., 2021). The data in all these scenarios exhibit temporal dependencies. Conventional feedforward neural networks lack the capability to consider these dependencies, prompting the emergence of RNNs specifically to handle this issue (Torres et al., (Khaldi et al., 2023; Torres et al., 2021). RNNs incorporate feedback loops to capture temporal dependencies, making them well-suited for tasks involving sequential data (Cirstea et al., 2018). RNNs possess a unique "memory" mechanism that allows them to incorporate past inputs to influence current outputs, making them an excellent option for tasks where element arrangement matters. To achieve this, cells with gates that use historical observations to influence the output are integrated into the network (Zeroual et al., 2020). Despite their potential, RNNs encounter various challenges, including the vanishing gradient problem. Developing resilient RNN models for time series forecasting remains difficult due to the lack of a comprehensive understanding of time series data itself. Consequently, there is limited knowledge about which cell structures are most suitable for different data types (Khaldi et al., 2023). Figure 3.7 presents a schematic diagram depicting RNNs and their mathematical formulation is shown in Equation 3.18.

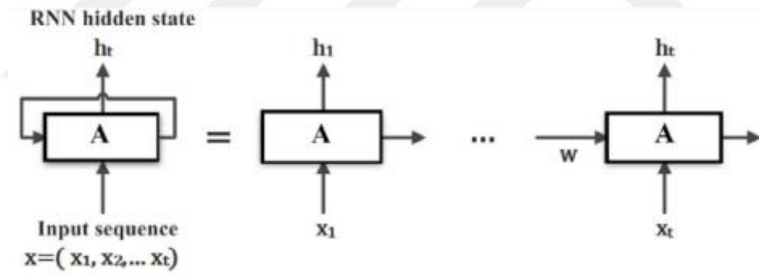

Figure 3.7. Representation of RNN(Zeroual et al., 2020).

In an RNN, the hidden state h_t can be calculated for a given input sequence $x = (x_1, x_2, x_3, \dots, x_t)$ as follows:

$$ht = \begin{cases} 0 & t = 0 \\ \varphi(W_{xt}, X_t) & \text{otherwise,} \end{cases}$$

where φ represents non-linear function. The update for the recurrent hidden state is as follows.

$$h_t = g(W_{xt} + uh_t - 1\,) \qquad \text{Equation 3.18}$$

where g denotes hyperbolic tangent function (tanh).

3.2.7. Bayesian Optimization

In this study, we applied Bayesian optimization methodology using the BayesianOptimization module from the kerastuner.tuners package. We also employed the hyperparameters module from the kerastuner.engine package to specify the hyperparameters for the optimization procedure.

According to (Snoek et al., 2012), the fundamental steps of Bayesian optimization (BO) are as outlined below and shown in Figure 3.8.

1. Construct a probabilistic surrogate model representing the objective function.
2. Identify the optimal hyperparameter values using the surrogate model.
3. Evaluate these hyperparameter values on the actual objective function.
4. Update the surrogate model with the obtained results.
5. Iterate through steps 2 to 4 until the maximum number of iterations is attained.

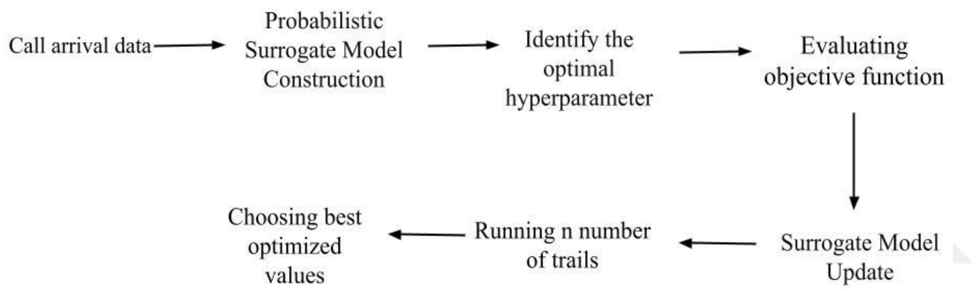

Figure 3.8. Bayesian optimization framework

Consistent with the methodology outlined by (Snoek et al., 2012), for all experiments conducted, the surrogate function utilized is the Gaussian Process (GP). GP being the conventional surrogate model used for modeling objective functions in Bayesian optimization (BO). It employs a probabilistic regression model where $f(y)$ conforms to a Gaussian Process as shown in Equation 3.19 (Cowen-Rivers et al., 2022).

$$x^* = \underset{y \varepsilon Y}{\arg\max} f(y) \qquad \text{Equation 3.19}$$

with y denoting a configuration choice, Y a mixed design space, and $f(y)$ a validation accuracy we wish to maximize.

BO employs a sequential decision-making approach to the global $f: Y \rightarrow \mathbb{Z}$ over a bounded input domain Y . BO aims to maximize a validation accuracy $f(y)$ over a bounded input domain Y. The process involves sequential decision-making rounds where the algorithm selects q inputs

$y_{1:q}^{new} \in Y^q$ and observes corresponding function values $z_{1:q}^{(new)} = f(y_{1:q}^{(new)})$. domain Y .The goal is to approach the maximum $y^* = \underset{y \varepsilon Y}{\arg\max} f(y)$ efficiently. Since both $f(.)$ and y^* are known, the algorithm balances exploitation and exploration using an acquisition function, which guides the selection of new points based on the trade-off between exploration and exploitation. Exploration involves sampling instances from unexplored areas, while exploitation entails sampling from currently promising regions where the global optimum is expected to be found, guided by the posterior distribution (Snoek et al., 2012).

3.2.8. Feature Selection

This section focuses on identifying significant features for our multivariate experiments. Given the dataset's abundance of features, it's crucial to pinpoint those with the most impact. Initially, we employed a RF feature selection method to assess each feature's contribution to model performance (Habtemariam et al., 2023; Andrade & Moazeni, 2023, 2023; Huo et al., 2021; X. Li et al., 2020).

Subsequently, to gain additional insights into feature relevance and relationships, we conducted Pearson correlation analysis among the selected features, enabling us to gauge individual features' predictive capabilities and identify any linear relationships, including positive and negative correlations(Chakrabarty, 2020; Hou et al., 2022; C. Li et al., 2020; Paouris et al., 2021). The combined approach of RF feature selection and Pearson correlation analysis offers additional insights into feature relevance. While RF feature selection identifies the most impactful features individually, Pearson correlation analysis digs into inter-feature relationships, including positive and negative correlations. By integrating both methods, we aim to enhance the feature selection process and improve the predictive power of our models(Andrade & Moazeni, 2023). In prioritizing features for inclusion in the predictive models, the study considered factors such as high dominance and strong positive correlations with the target variables. Features demonstrating significant importance on call arrival patterns and exhibiting strong relationships with other variables were given precedence in the selection process. In cases where feature importance was low, those with higher positive correlations were prioritized.

Random Forest Feature selection

Feature selection using random forests has two main objectives: to identify highly dependent feature variables (X. Li et al., 2020) and to find feature variables with relatively little data that can effectively express forecasting results (Niu et al., 2020). Random forests are widely used because they provide importance measures, offering insights at a global level regarding the relevance of input variables in predicting a specific output. Two common measures for this are Mean Decrease Impurity (MDI) and Mean Decrease Accuracy (MDA) (Gwetu et al., 2019).

In this study, we utilized the Mean Decrease Impurity (MDI) method to evaluate the importance of features within a Random Forest model. MDI, also known as Gini importance, is a typical example of an embedded feature selection method since it uses information generated in the RF learning process, making it a computationally attractive option. It doesn't demand much space, usually just order A, as it keeps track of the total Gini impurity values for each attribute (Gwetu et al., 2019). The MDI method assesses the contribution of each feature to the predictive accuracy of the model by measuring the decrease in Gini impurity when the feature is used for splitting the data (X. Li et al., 2020). The detailed process for calculating feature importance is shown in Equation 3.20.

$$MDI\ (X,j) = \frac{1}{T}\sum_{k=1\Theta_k^d \in h_k}^{T}\ \sum_{:\Theta_k^o = X_k \phi \in}\ \sum_{\Phi(\Theta_k^d{'}j)} \delta GI(\phi) I(\phi = \phi^* \Lambda j = j^*) \qquad \text{Equation 3.20}$$

Where T represents the total number of trees in the Random Forest

Θ_k^d denotes the set of decision nodes in tree k at depth d

h_k represents the set of decision paths within tree k

Θ_k^o is the root node of tree k

X_k is the feature vector for tree k

$\Phi(\Theta_k^d{'}j)$ represents the set of all splits in tree K involving feature j.

$\delta GI(\phi)$ measures the decrease in Gini impurity at split ϕ

$I(\phi = \phi^* \Lambda j = j^*)$ I an indicator returning 1 if split ϕ is the optimal split involving feature, and 0 if otherwise

The MDI value is averaged over all T trees in the forest. For each tree, the decision paths h_k where feature j is used are considered. Only splits where feature j yields the optimal split ($\phi = \phi^*$) are taken into account. The decrease in Gini impurity $\delta GI(\phi)$ at each split involving feature j is summed up. This method ensures a detailed and rigorous calculation of feature importance by considering the optimal splits and their contribution to reducing impurity across all trees in the Random Forest. By using this approach, we can accurately identify the most significant features contributing to the prediction model, thereby enhancing the model's interpretability and performance.

Pearson correlation

The Pearson correlation analysis is a common criterion to measure the correlation between variables (Cui et al., 2020). Pearson correlation formulation is in Equation 3.21.

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^{2\cdot}}\sqrt{\sum(Y_i - \bar{Y})^2}}$$ Equation 3.21

It ranges from -1 to 1, where:

$r = 1$ ndicates a perfect positive linear relationship, $r = -1$ indicates a perfect negative relationship and $r = 0$ indicates no linear relationship, X_i and Y_i are individual data points and $\bar{X}$ and , $\bar{Y}$ are the means of variables X and Y, respectively.

3.3. Theoretical Framework

To provide a theoretical foundation for our investigation into call arrival prediction, we developed a conceptual framework (Figure 3.9). This framework integrates essential theoretical concepts and research components, serving as a guide for understanding the relationships between different variables and the methodology employed to explore these connections

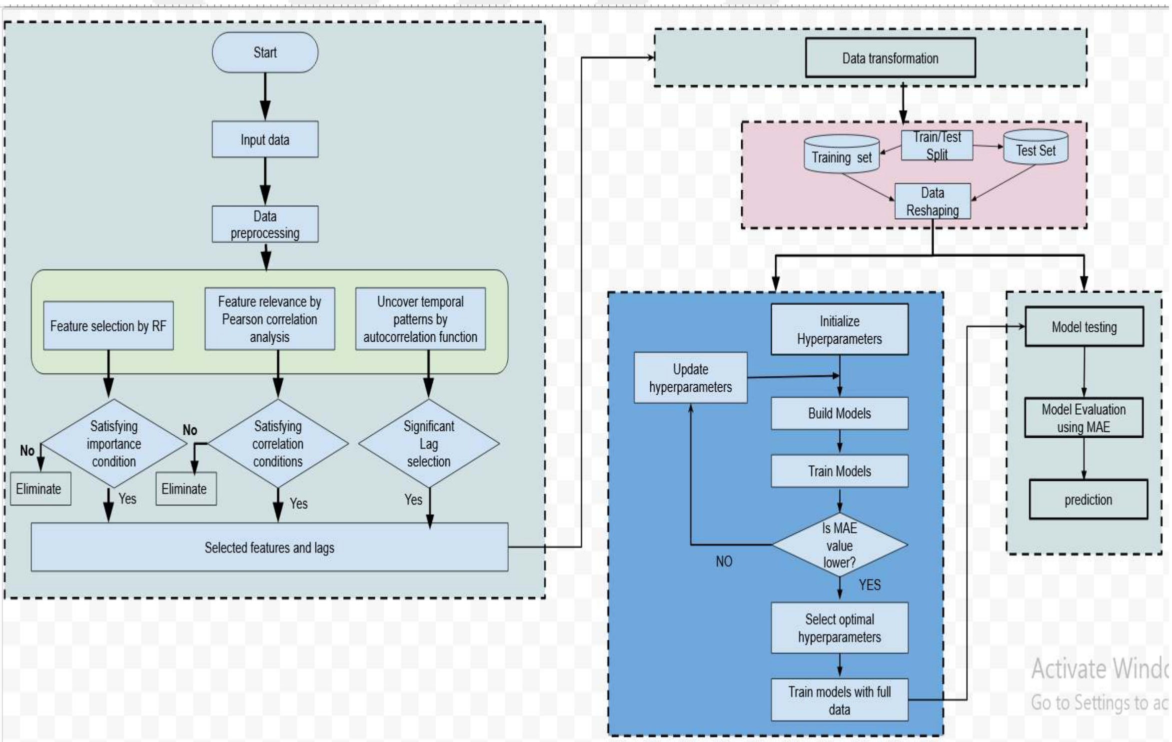

Figure 3.9. Over view of theoretical framework

4. RESULTS AND DISCUSSIONS

This section offers a concise analysis of the outcomes from conducted experiments. It presents the results of autocorrelation analysis, feature selection process, Bayesian optimization, performance of implemented models, identifies key findings, and discusses their implications. These operations are done with six ML and DL models including MLP, RF, SVM, RNN, CCN and LSTM using three datasets of daily, hourly, and half-hourly aggregations. The prediction task involved both call volume and average handling time(AHT). All graphs for all models are plotted with MAE values. Obtained test results were evaluated in terms of each model, then results for all optimized univariate models and multivariate models with all features and selected features are examined respectively and the results of the best performing univariate models and multivariate models are summarised in the conclusion section.

4.1. Experimental Results

4.1.1. Autocorrelation results

After employing autocorrelation functions, the findings unveiled repeating patterns in the data with a lag of 7 time units for daily observations, 24 time units for hourly observations, and 48 time units for half-hourly observations. This indicates recurring patterns in the data, with cycles repeating every 7 days for daily observations, every 24 hours for hourly observations, and every 48 half-hours for half-hourly observations. This is visually depicted in Figure 4.1 through Figure 4.6. Consequentially, this led to the selection of 7, 24, and 48 as optimal sequences for daily, hourly, and half-hourly datasets, respectively.

ACF graphs and Partial ACF graphs illustration

The autocorrelation plot for daily Call volume, illustrated in Figure 4.1, reveals high peaks at lags 1, 7, 14, and a weekly repeating pattern. Positive significance continues up to lag 36 except for lag 33, then decreases gradually as the lag value increases but maintains the significant positive weekly spikes above the confidence level until lag 70. This pattern signifies a strong weekly seasonality within the data, indicating that daily Call volume values from the previous 36 days' exhibit notable and statistically significant correlations.

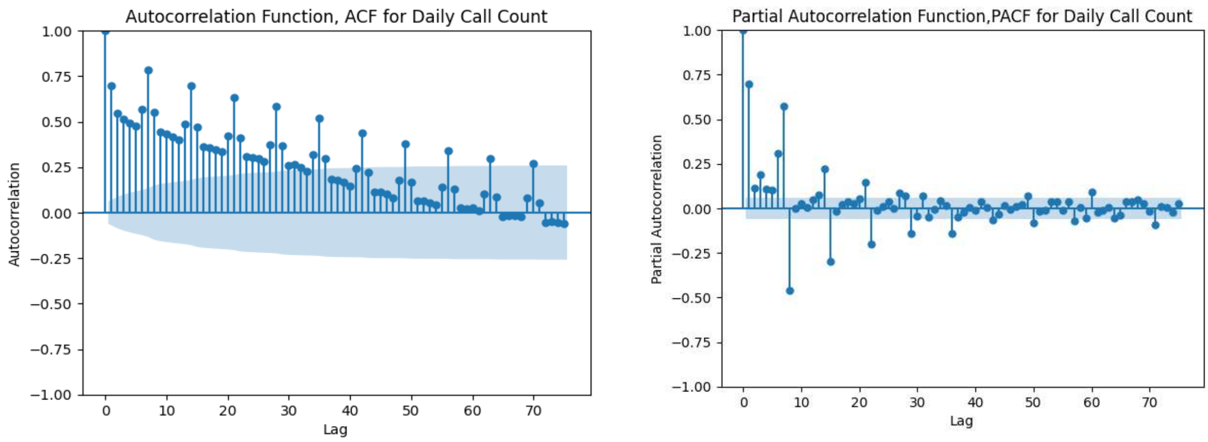

Figure 4.1. Autocorrelation (left) Partial Autocorrelation (right) for daily Call volume

The autocorrelation plot for hourly Call volume, illustrated in Figure 4.2, reveals high positive peaks at lags 1, a pattern of troughs at lag 12 and peaks at lag 24. Peaking positively at lag 1, indicates that recent observations are influential, positive peaks at lags such as 24,48 suggest a strong positive correlation every 24 hours, which could represent a daily pattern in the hourly Call volume data.

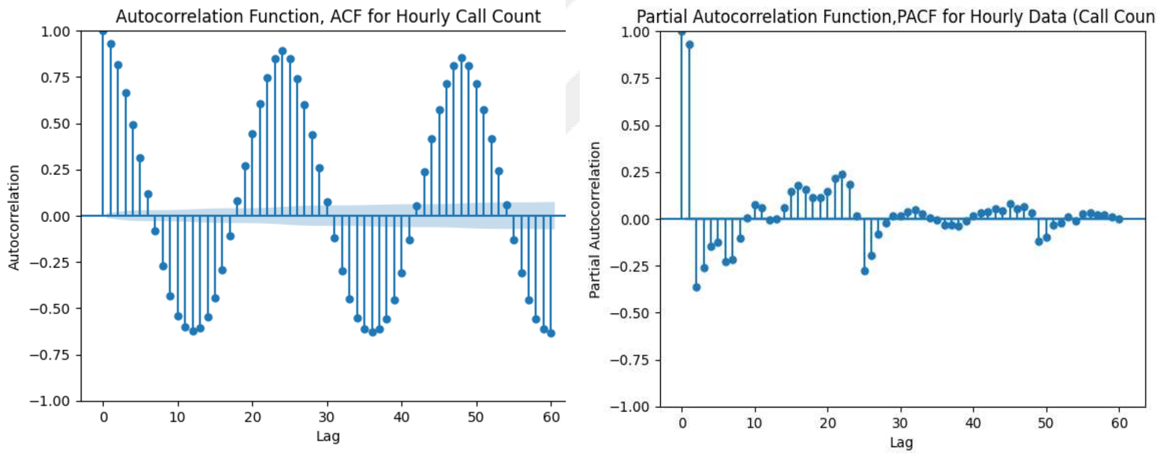

Figure 4.2. Autocorrelation (left) Partial Autocorrelation (right) for hourly Call volume

The autocorrelation plot for half-hourly Call volume, illustrated in Figure 4.3., reveals high positive peaks at lags 1, a pattern of troughs at lag 24 and peaks at lag 48. Positive autocorrelation at lag 1 suggests that recent observations are influential, while half hourly positive patterns at lag 48 indicate the importance of capturing daily cycles.

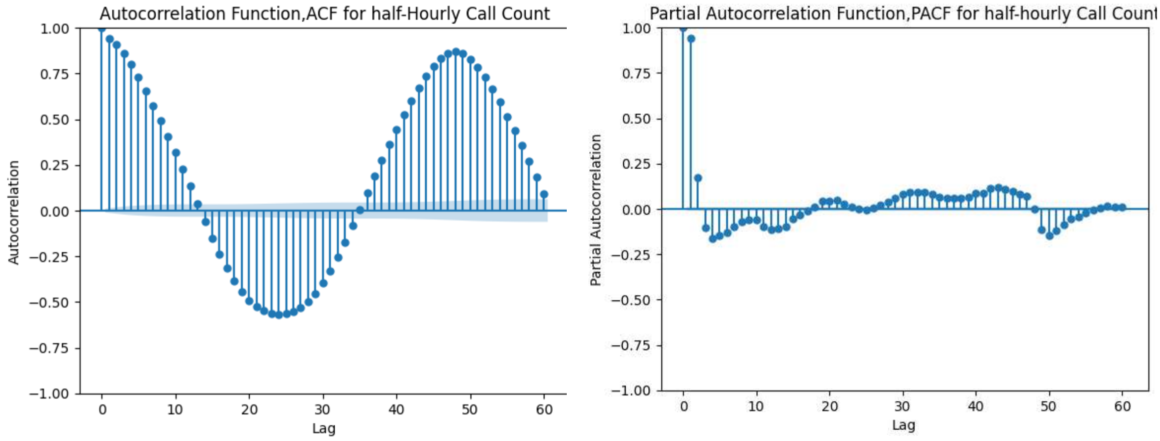

Figure 4.3. Autocorrelation (left) Partial Autocorrelation (right) for half-hourly Call volume

Average Handling Time (AHT)

The autocorrelation plot for daily AHT, illustrated in Figure 4.4., reveals high peaks at lags 1, 6, 14,21 and a repeating 7-day pattern. Positive significance continues up to lag 70. This pattern signifies a strong weekly seasonality within the data, indicating that daily AHT values upto lag 70 exhibit significant positive correlations. The analysis captures recurring weekly trends, offering valuable insights into the temporal dependencies inherent in the time series.

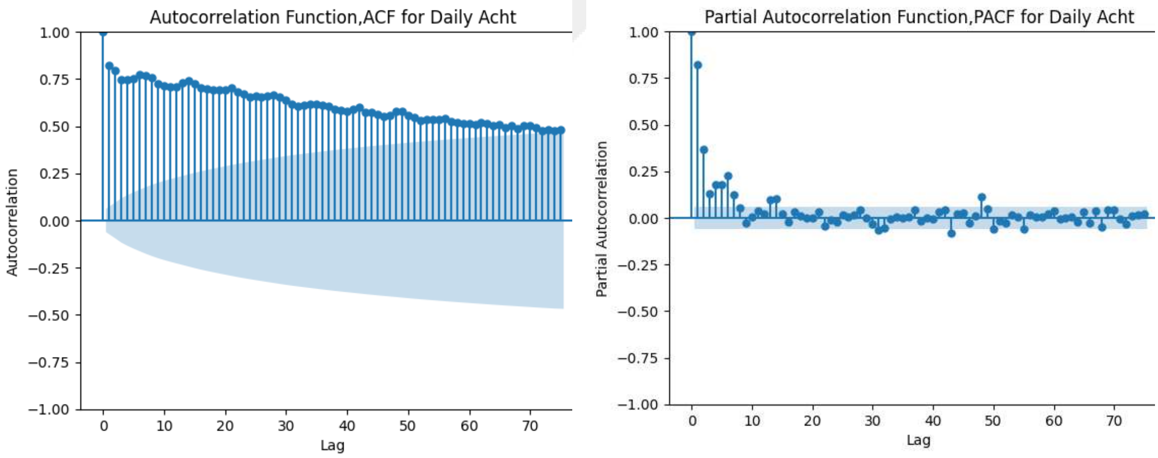

Figure 4.4. Autocorrelation (left) Partial Autocorrelation (right) for daily AHT

The autocorrelation plot for hourly AHT, illustrated in in Figure 4.5., reveals high positive peaks at lags 1,24,48. This 24 hourly pattern of peaks provides insights into the temporal dependencies; Peaking positively at lag 1, indicates that recent observations are influential, positive peaks at lags such as 24,48 suggest a strong positive correlation every 24 hours, which could represent a daily cycle in the hourly AHT data.

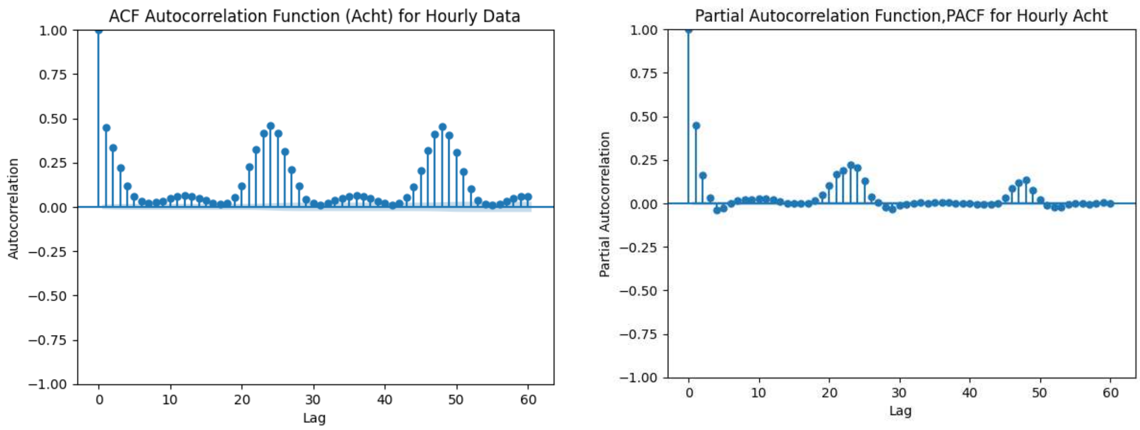

Figure 4.5. Autocorrelation (left) Partial Autocorrelation (right) for hourly AHT

The autocorrelation plot for half-hourly AHT, illustrated in Figure 4.6., reveals high positive peaks at lags 1, that troughs but positively peaks again at lag 48. Positive autocorrelation at lag 1 suggests that recent observations are influential, while half hourly positive patterns at lag 48 indicate the importance of capturing daily cycles

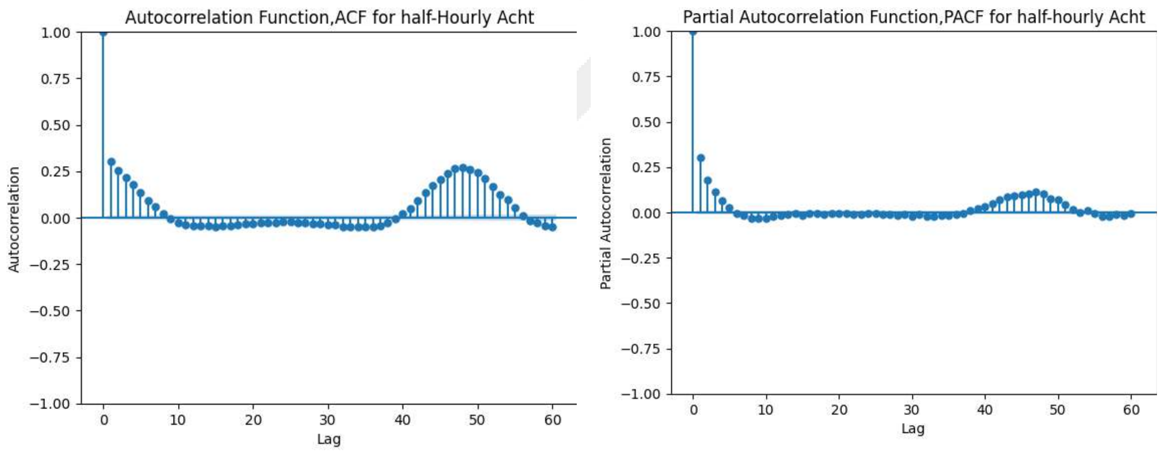

Figure 4. 6. Autocorrelation (left) Partial Autocorrelation (right) for half hourly AHT

4.1.2. Baseline models results

In preliminary stages, our examination of call arrival prediction with baseline models whose hyper parameters were chosen manually (Table 4.1) noted significant performance discrepancies among the various models. The utilization of default hyper parameters resulted in suboptimal performance across the board as discussed below.

For multivariate observations:

In daily forecasting, LSTM models exhibited the highest MAE of 0.44 for call volume, while RF achieved the lowest MAE of 0.04. Conversely, for Average Handling Time (AHT), RNN showed the highest MAE of 0.47, with RF and MLP recording the lowest MAE of 0.06.

For hourly forecasting, CNN displayed the highest MAE of 0.74 for call volume, while RF attained the lowest MAE of 0.06. In terms of AHT, CNN had the highest MAE of 0.53, whereas RF achieved the lowest MAE of 0.09.

In half-hourly forecasting, CNN demonstrated the highest MAE of 0.74 for call volume, while MLP exhibited the lowest MAE of 0.05. Similarly, for AHT, CNN displayed the highest MAE of 0.78, whereas RF recorded the lowest MAE of 0.06.

For univariate observations:

In daily forecasting, SVM exhibited the highest MAE of 0.14 for call volume, while RF achieved the lowest MAE of 0.05. Regarding Average Handling Time (AHT), CNN recorded the highest MAE of 0.18, while RF and MLP had the lowest MAE of 0.06.

For hourly forecasting, CNN displayed the highest MAE of 0.23 for call volume, whereas RF attained the lowest MAE of 0.04. In terms of AHT, RNN showed the highest MAE of 0.72, while RF recorded the lowest MAE of 0.09.

In half-hourly forecasting, CNN demonstrated the highest MAE of 0.88 for call volume, with RF and MLP achieving the lowest MAE of 0.06. Similarly, for AHT, CNN displayed the highest MAE of 0.39, whereas RF recorded the lowest MAE of 0.06.

Table 4.1. Benchmark hyper parameters for the baseline models

MODELS	PARAMETERS	VALUES
SVM	kernel	linear
RF	n_estimators	100
MLP	hidden_layer1	64
	activation	Relu
	activation	linear
	Optimizer	Adam
CNN	layer1	32
	layer2	64
	kernel size	3
	pool size	2
	activation	relu
	Optimizer	Adam
	learning rate	0.1
	epoch	4
RNN	layer1	64
	activation	tanh
	activation	linear
	Optimizer	Adam
	learning rate	0.1
	epoch	4
LSTM	layer1	64
	activation	tanh
	activation	linear
	Optimizer	Adam
	learning rate	0.1
	epoch	4

4.1.3. Bayesian Optimization

Upon recognizing the limitations of baseline models in handling multivariate data, our study sought to address this challenge through optimization techniques. We conducted hyper parameter tuning using a set of predefined values and the Bayesian optimization algorithm.

Subsequently, a set of hyper parameters with a range of values that suit specific models was used to tune the proposed algorithms (see Table 4.2). The hyper parameters were defined. The models were built using these hyper parameters and the optimal values were searched from the hyper parameter space using the BO algorithm however hyper parameters for epochs, drop out and batch size were selected as 2, 50 and 32. The dropout technique prevented overfitting and leaves unrelated information from the network to improve performance (Rehman et al., 2019).

Initially, the optimal models were trained using the complete set of features from the dataset, and their performance was assessed.

Table 4.2. Optimal hyper parameters for the different models using BO

Models	Hyper parameters	Range Values	Optimal Parameters
Long Short Term Memory (LSTM)	Learning Rate	0.0001,0.001,0.002,0.003,0.01,0.02	0.003
	Activation Function	relu, tanh, Linear	relu
	Optimizer	Adam, RMSprop, sgd	Adam
	Neurons	range of min_value=50, max_value=1000, step=100	850
Convolutional Neural Network (CNN)	Neurons layer1	min_value = 32, max_value = 512, step=32	128
	Neurons layer2	min_value =32, max_value=512, step=32	64
	Kernel size	min_value=3, max_value=5, step=1	5
	Pool size	min_value=2, max_value=8, step=1	6
	Learning Rate	0.0001,0.001,0.002,0.003,0.01,0.02	0.003
	Activation	relu, tanh, Linear	Tanh
	Optimizer	Adam, RMSprop, SGD	Adam
Random Forest (RF)	n_estimators	20,50, 100, 150	20
	max_depth	None, 10, 20	10
	min_samples_split	2, 5, 10	2
	min_samples_leaf	1, 2, 4	1
Support Vector Machines (SVM)	Kernel	linear, rbf, poly	rbf
	C	10,20,60,80,100,120,150	100
	epsilon	0.02,0.04,0.06,0.08,0.5	0.06
Multi-Layer Perceptron (MLP)	Dense layer1	min_value = 32, max_value = 512, step = 32	128
	Learning Rate	0.0001,0.001,0.002,0.003,0.01,0.02	0.003
	Activation	relu, tanh, Linear	relu
	Optimizer	Adam, RMSprop, SGD	Adam
Recurrent Neural Network (RNN)	Dense layer1	min_value=32, max_value=512, step=32	128
	Learning Rate	0.0001,0.001,0.002,0.003,0.01,0.02	0.003
	Activation	relu, tanh, Linear	relu
	Optimizer	Adam, RMSprop, SGD	Adam

4.1.4. Feature Selection Results

Following the initial training of optimal models with the complete dataset, a feature selection process was implemented to refine the models further. Firstly, the RF feature selection process shed light on key predictors crucial for call volume and average handling time prediction. Interestingly, most features were deemed to have low importance, signaling a necessity for a more precise selection approach. To gain further insights into feature relevance and relationships, we conducted Pearson correlation analysis. Remarkably, despite their low importance, some of these features exhibited positive correlations, emphasizing their significance in the prediction process. Our findings highlighted several prominent features.

Features were selected based on their high correlation and slightly higher importance. Notably, CallCount, TalkCount, and TalkTimeSec emerged as primary predictors for call volume forecasting. Conversely, for the prediction of AHT, features such as Acht, TalkTimeSec, and Answered30PSec were identified as crucial factors. These findings are summarized in Table 4.3 and visually represented in Figure 4.7 and Figure 4.8; Figure 4.9 shows the correlation matrix for all features in the dataset.

Table 4.3. Feature selection results

Target variable	Features	Feature Importance Score	Pearson Correlation analysis score
CallCount	CallCount	**0.9674**	**1.000**
	TalkCount	**0.0256**	**0.990**
	TalkTimeSec	**0.0040**	**0.932**
	Answered20Sec	0.0011	0.833
	Answered20and30Sec	0.0010	0.854
	Answered30PSec	0.0003	0.588153
	Abandon20Sec	0.0002	0.573
	Abandon Call	0.000094	0.409108
	Abandon20and30Sec	0.000073	0.499980
	Abandon30PSec	0.0003	0.389
Average Handling Time	Acht	**0.9995**	**1.000**
	TalkCount	0.00007	-0.14
	Abandon30PSec	0.00007	0.197
	Abandon Call	0.00007	0.176
	Abandon20and30Sec	0.00006	0.024
	Answered30PSec	**0.00005**	**0.202**
	TalkTimeSec	**0.00004**	**0.220**
	Answered20Sec	0.00003	-0.267
	Abandon20Sec	0.00003	-0.267
	Call Count	0.00002	-0.107

Figure 4.7. Feature importance and pearson correction score for Call volume data

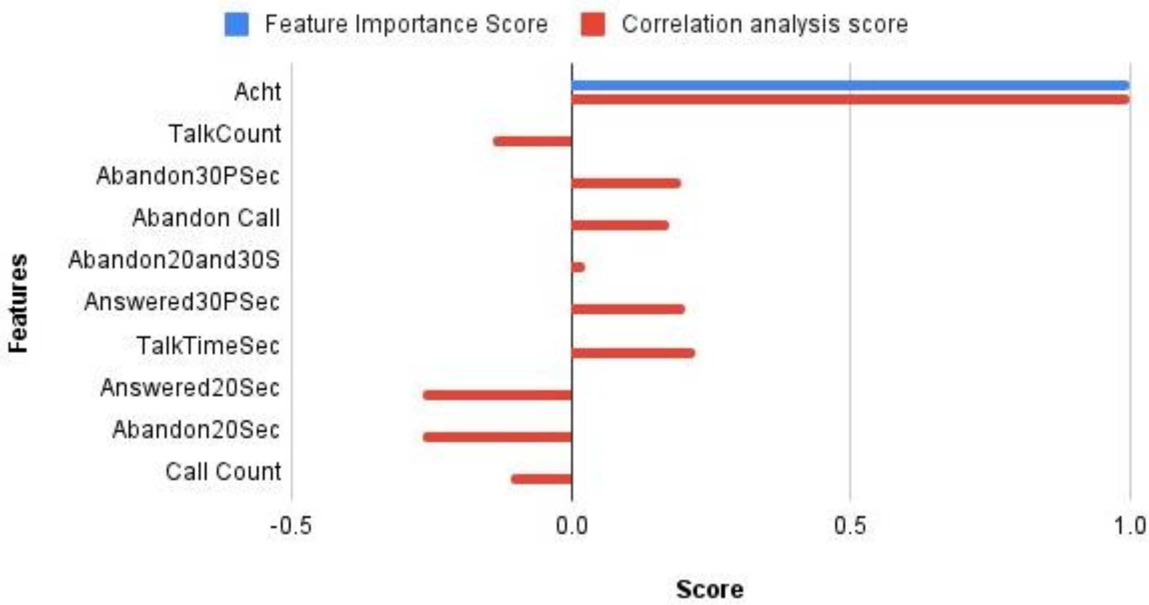

Figure 4.8. Feature importance and pearson correction score for Average handling time data

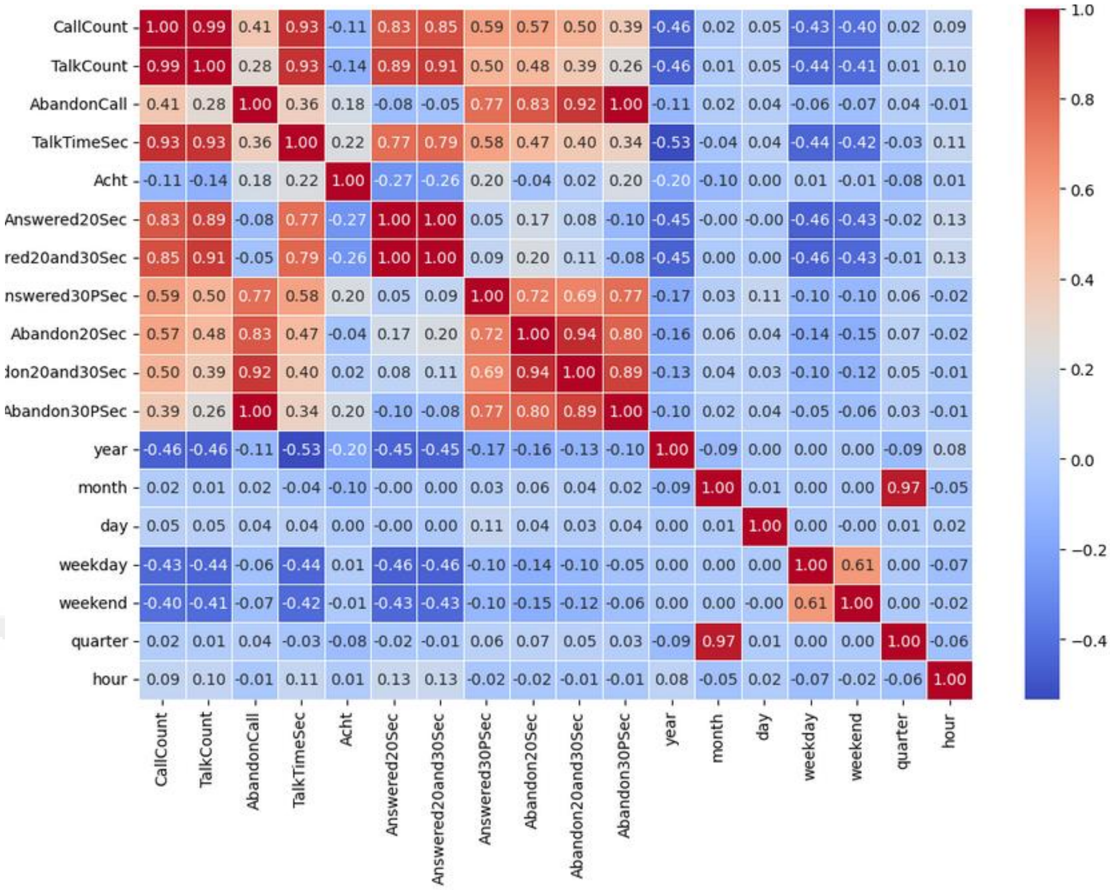

Figure 4.9. Correlation matrix for all features for call Arrival observation

4.1.5. Model Performance by Dataset

This section provides a comprehensive comparison of various model's performance across different multivariate datasets (daily, hourly, and half-hourly) under three distinct scenarios: baseline performance, optimised model performance utilizing all available features, and optimised model performance employing features selected through feature selection techniques. Obtained results were evaluated in terms of each model. Each dataset is evaluated based on the MAE metric and RMSE, representing various performance aspects such as call volume and AHT. Results were discussed based on MAE.

MLP

Table 4.4. Multivariate observations for MLP model

Prediction Task	Multivariate Datasets	Baseline MAE	Optimized (All Features) MAE	Optimized (All Features) RMSE	Optimized (Selected Features) MAE	Optimized (Selected Features) RMSE
Call volume	Daily	0.05	0.03	0.04	0.02	0.03
	Hourly	0.18	0.03	0.04	0.02	0.03
	Half hourly	0.05	0.04	0.05	0.03	0.04
Average handling time	Daily	0.06	0.03	0.04	0.02	0.03
	Hourly	0.16	0.03	0.05	0.02	0.03
	Half- hourly	0.08	0.04	0.05	0.02	0.03

Table 4.5. Univariate observations for MLP model

Univariate	Datasets	Baseline MAE	Optimized MAE	Optimized RMSE
Call volume	Daily	0.06	0.03	0.04
	Hourly	0.07	0.03	0.04
	Half hourly	0.05	0.04	0.05
Average handling time	Daily	0.05	0.04	0.05
	Hourly	0.06	0.04	0.05
	Half hourly	0.07	0.03	0.05

Table 4.4 is interpreted as a result of the tests made with multivariate MLP, the models performance improves when using an optimized model in comparison with baseline model, as evidenced by decreased values in call volume and average handling time metrics. More specifically, model performance was much better with optimized models with selected features. For call volume predication, daily and hourly dataset had the best estimates with MAE of 0.02. For Average Handing Time predications, all datasets performed satisfactorily with MAE of 0.02. These results are visualized in Figure 4.10Figure 4.10 and 4.12.

Table 4.5, focuses on univariate MLP and compares baseline with optimized model performances. Results show that optimized models improved model performance across all datasets. Specifically, for univariate call volume predication, Hourly and Daily dataset with MAE of 0.03 had the best estimates. For Average Handing Time predications, Half hourly dataset with MAE of 0.03 performed satisfactorily (see Figure 4.13 and 4.13). It's generally observed that BO MLP models greatly improved prediction accuaracy. Likewise, multivariate models with feature selection outperformed univariate models.

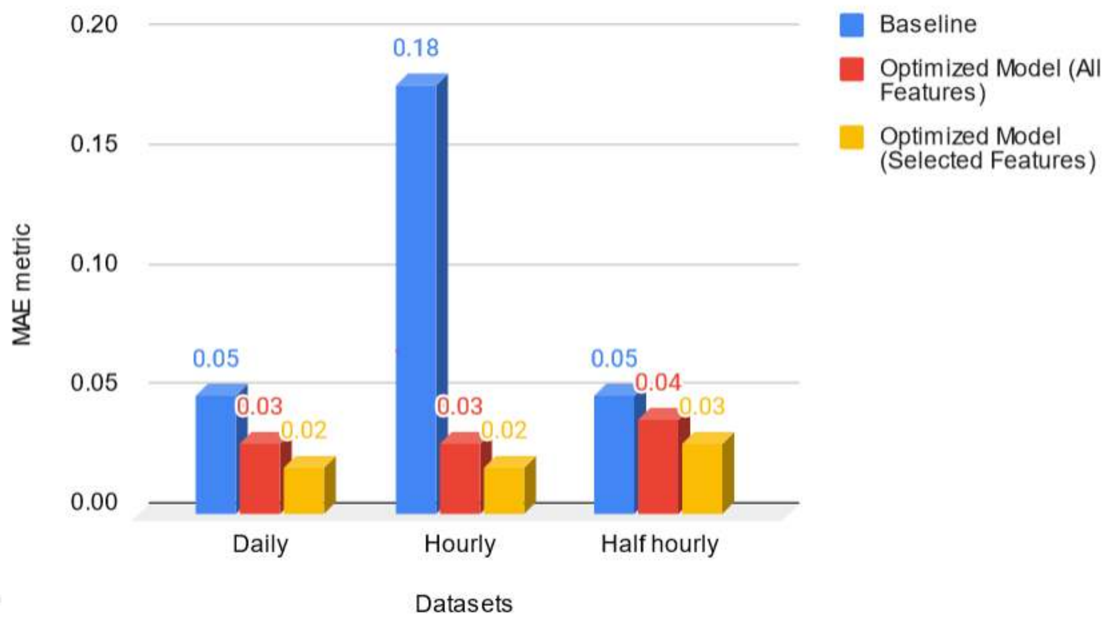

Figure 4.10. Multivariate Call Volume Prediction, MLP Model

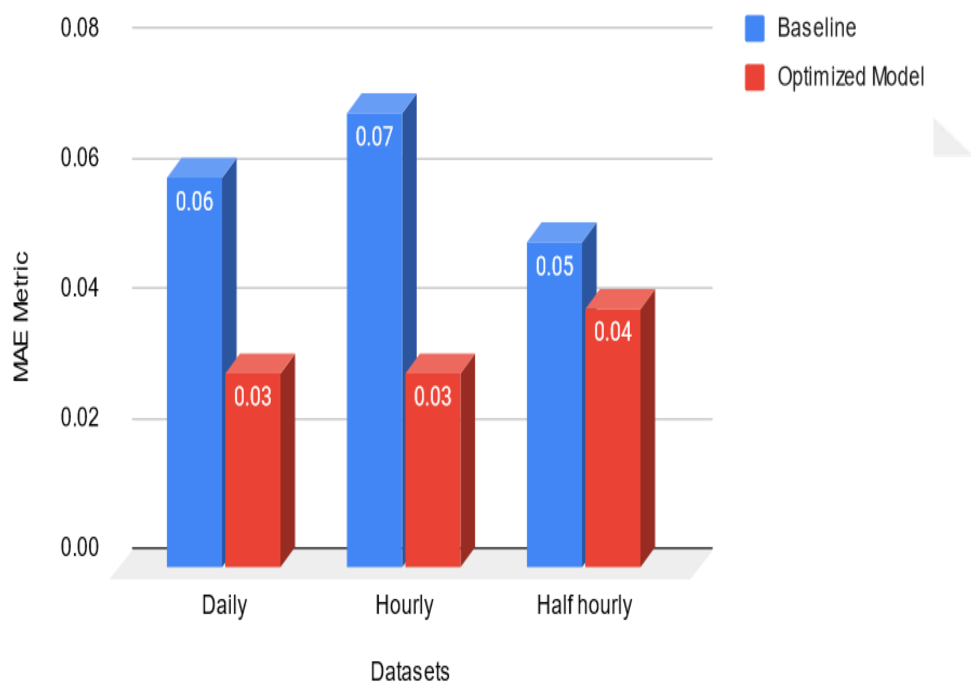

Figure 4.11. Univariate Call Volume Prediction, MLP Model

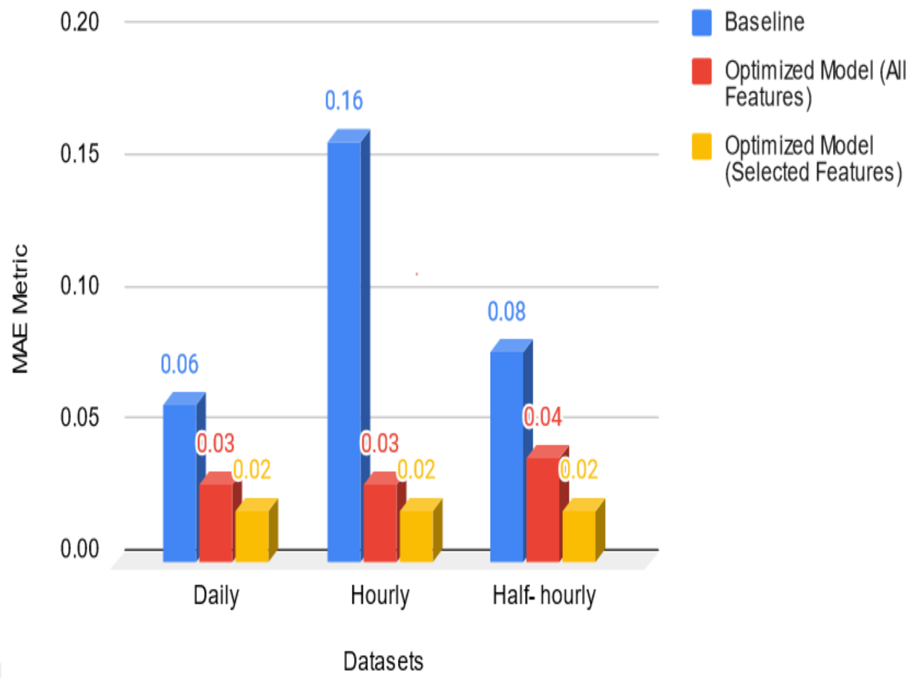

Figure 4.12. Multivariate Average Handling Time MLP Model

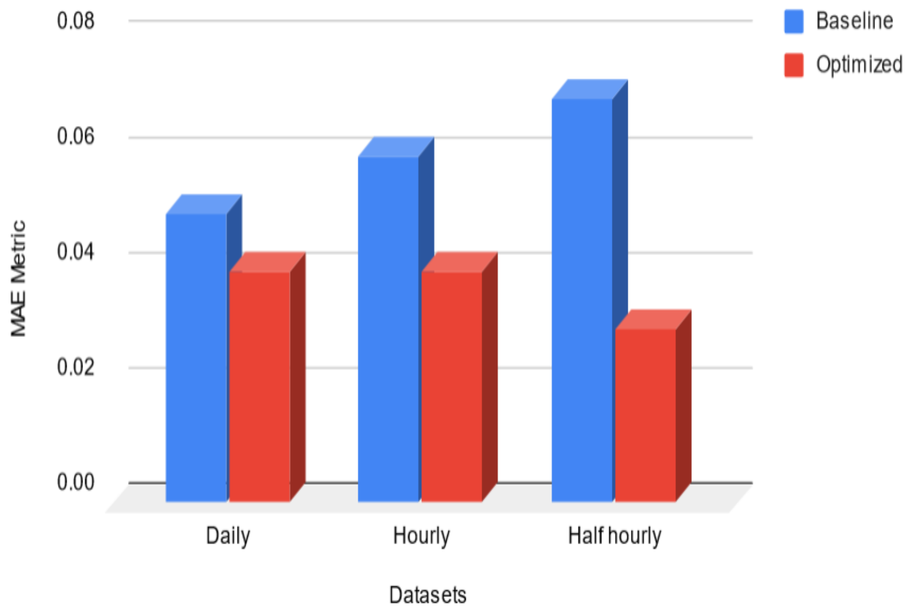

Figure 4.13. Univariate Average Handling Time Prediction, MLP Model

RF

Table 4.6. Multivariate observations for RF model

Multivariate	Datasets	Baseline MAE	Optimized (All Features) MAE	Optimized (All Features) RMSE	Optimized (Selected Features) MAE	Optimized (Selected Features) RMSE
Call volume	Daily	0.04	0.03	0.05	0.02	0.03
	Hourly	0.06	0.04	0.06	0.02	0.03
	Half hourly	0.06	0.03	0.05	0.03	0.04
Average handling time	Daily	0.06	0.04	0.05	0.02	0.03
	Hourly	0.09	0.05	0.07	0.04	0.06
	Half hourly	0.06	0.04	0.05	0.02	0.03

Table 4.7. Univariate observations for RF model

Univariate	Datasets	Baseline MAE	Optimized MAE	Optimized RMSE
Call volume	Daily	0.05	0.03	0.05
	Hourly	0.04	0.03	0.04
	Half hourly	0.05	0.03	0.05
Average handling time	Daily	0.06	0.04	0.05
	Hourly	0.06	0.04	0.05
	Half hourly	0.06	0.03	0.04

Table 4.6 results demonstrate that the multivariate RF model's performance improves when using Optimized Model and a greater improvement more especially optimized models with selected features. For call volume predictions, daily and hourly datasets performed adequately with MAE of 0.02 whereas for Average handling time, daily and half hourly datasets with MAE of 0.02 performed well. These results are visualized in Figure 4.14 and Figure 4.16.

Table 4.7 focuses on univariate RF model performance comparing baseline and optimal model performance. For call volume forecasting all datasets performed satisfactorily with MAE of 0.03 whereas for AHT Half hourly performed well with MAE of 0.03. These results are visualized in Figure 4.15 and Figure 4.17. It's generally observed that Optimised RF model greatly improved prediction accuracy especially with multivariate models with selected features.

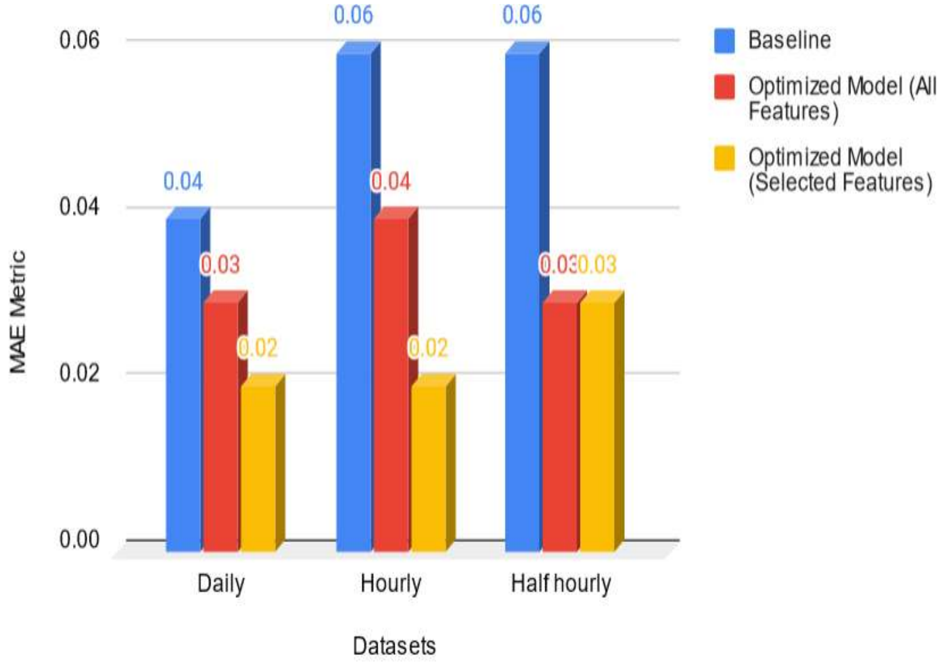

Figure 4.14. Multivariate Call Volume Prediction, RF Model

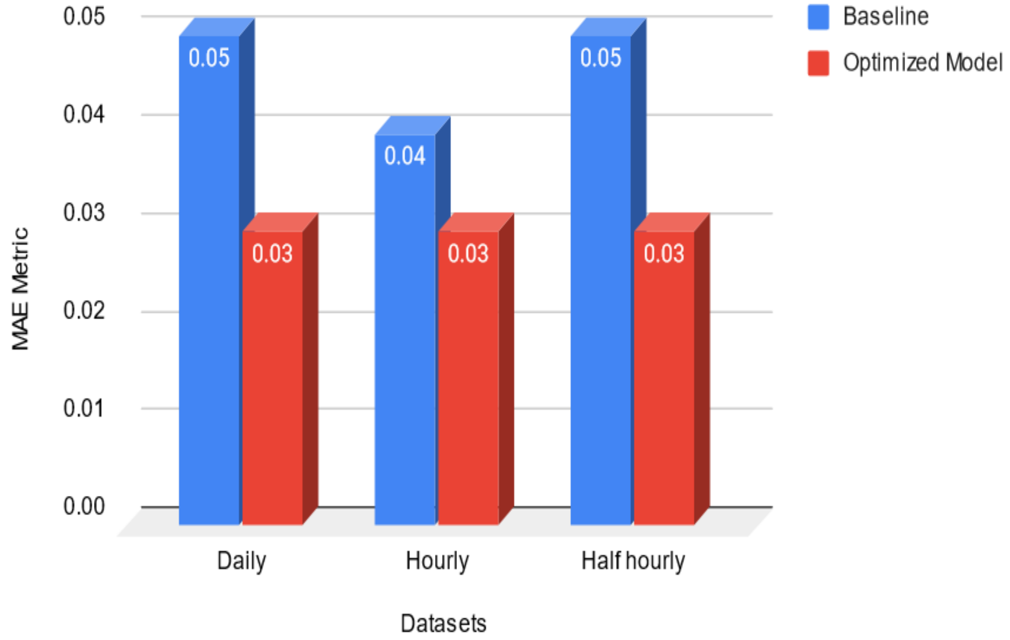

Figure 4.15. Univariate Call Volume Prediction, RF ModeL

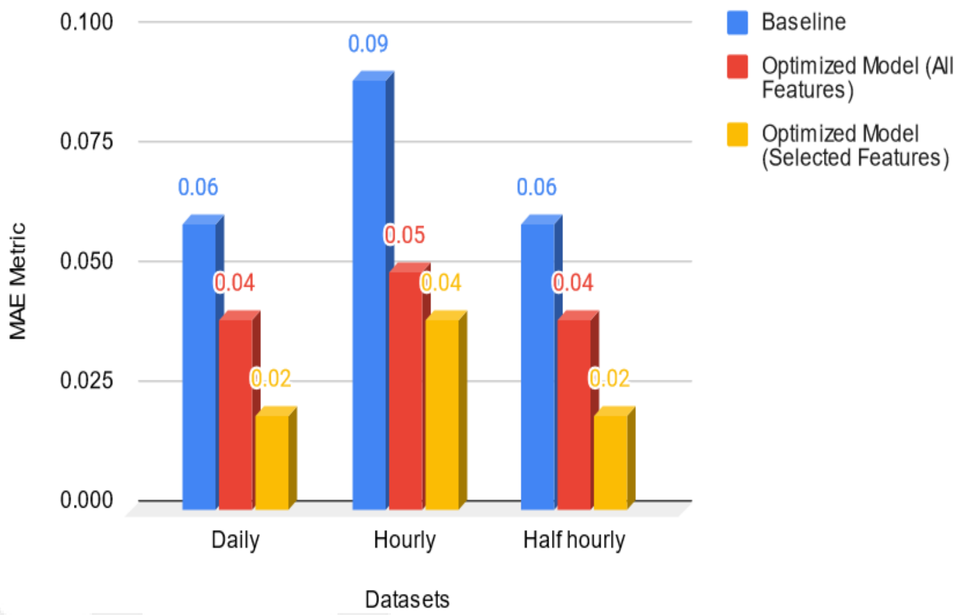

Figure 4.16. Multivariate Avergae Handling Time, RF Model

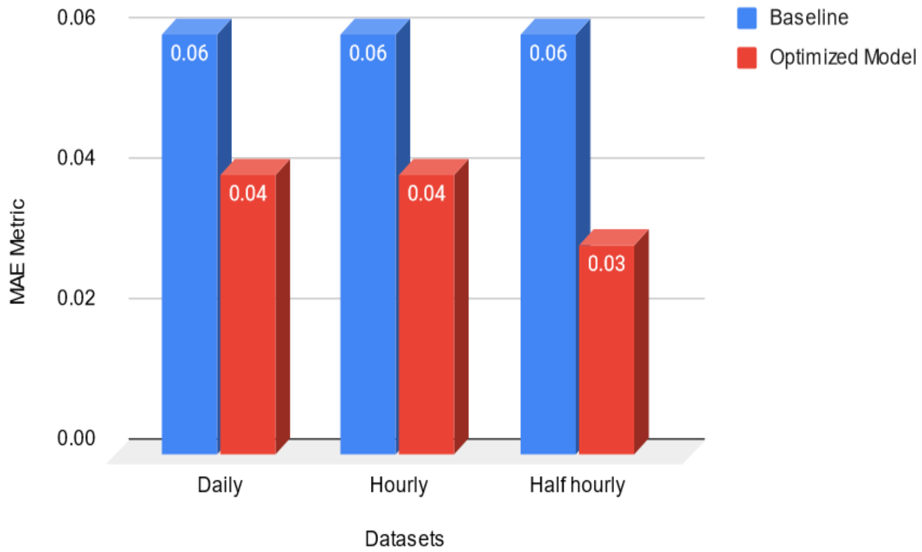

Figure 4.17: Univariate Avergae Handling Time, RF Model

SVM

Table 4.8. Multivariate observations for SVM model

Multivariate	Datasets	Baseline MAE	Optimized (All Features) MAE	Optimized (All Features) RMSE	Optimized (Selected Features)	Optimized (Selected Features) RMSE
Call volume	Daily	0.12	0.08	0.10	0.03	0.05
	Hourly	0.08	0.07	0.09	0.03	0.05
	Half hourly	0.07	0.06	0.08	0.04	0.06
Average handling time	Daily	0.14	0.07	0.09	0.03	0.04
	Hourly	0.11	0.09	0.11	0.03	0.05
	Half hourly	0.07	0.06	0.08	0.04	0.06

Table 4.9. Univariate observations for SVM model

Univariate	Datasets	Baseline MAE	Optimized MAE	Optimized RMSE
Call volume	Daily	0.14	0.04	0.05
	Hourly	0.09	0.03	0.04
	Half hourly	0.07	0.04	0.05
Average handling time	Daily	0.12	0.04	0.05
	Hourly	0.05	0.03	0.04
	Half hourly	0.08	0.04	0.05

Multivariate Observations (Table 4.7):

Table 4.8. Multivariate observations for SVM model results demonstrate that model's performance improved when using optimized SVM models, as evidenced by decreased call volume and average handling time metrics across different datasets. In addition to that, models with selected features outperformed optimised models with all features. Additionally, for Call volume predictions and AHT, daily and hourly dataset had the lowest MAE of 0.03.Visual representations are in Figure 4.18 and Figure 4.20.

Table 4.9 focuses on univariate RF model performance comparing baseline and optimal model performance. Results indicate optimized models outperformed baseline models. For univariate predictions, hourly datasets had the lowest MAE of 0.03 for both Call volume and AHT predictions as seen as Figure 4.19 and Figure 4.21.

The superior performance of hourly datasets for univariate and both hourly and daily datasets emphasizes the significance of increasing granularity in data as it provides more datapoints for the model to learn from.

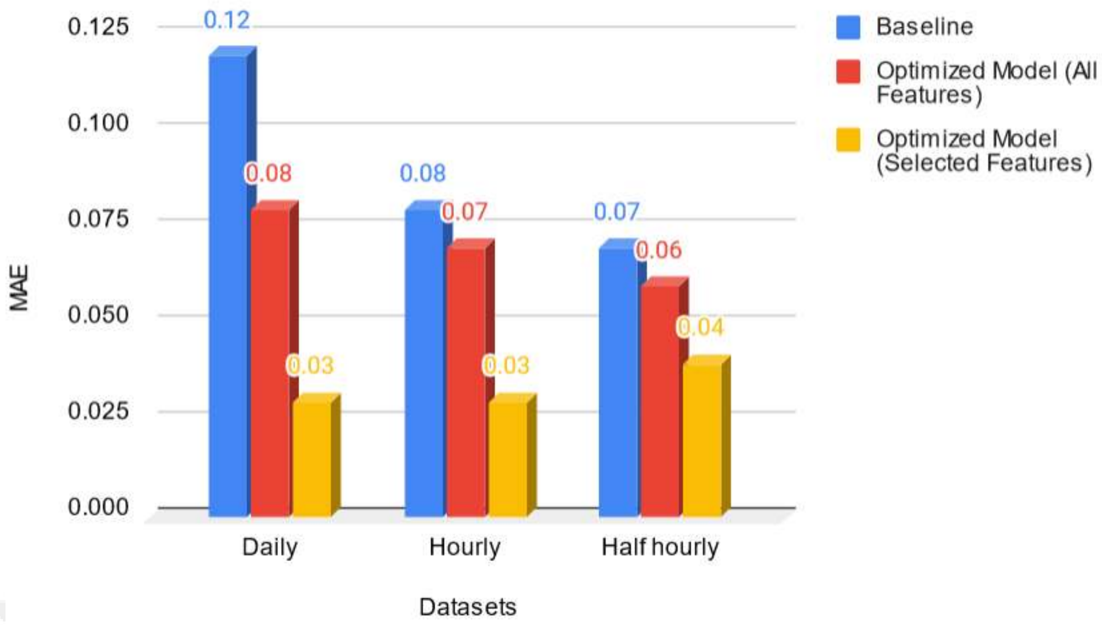

Figure 4.18. Multivariate Call Volume Prediction, SVM Model

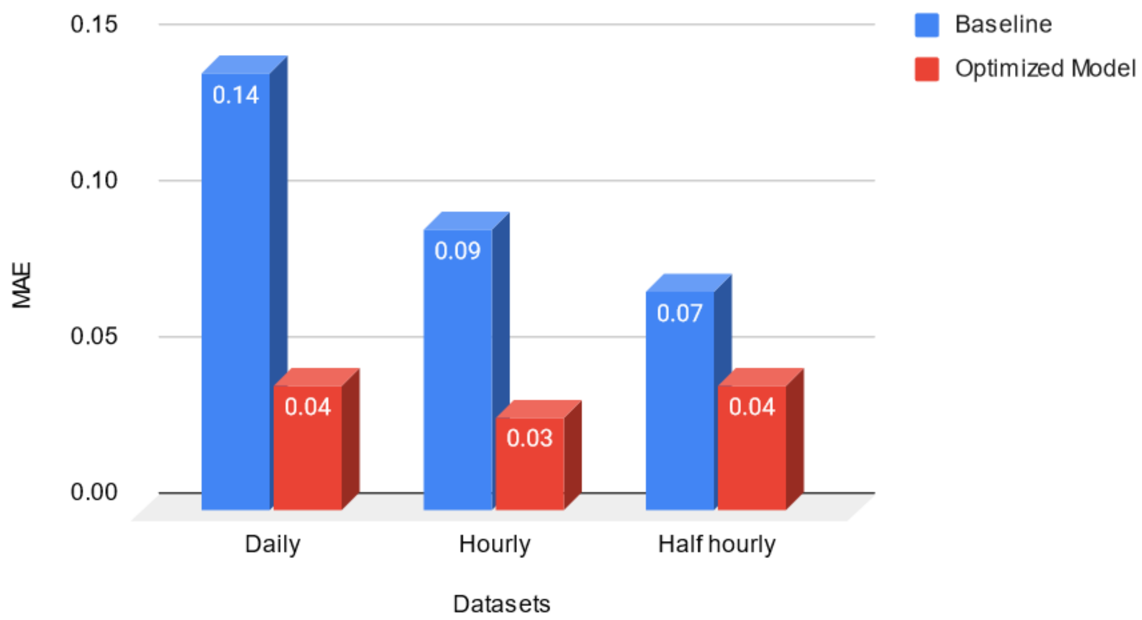

Figure 4.19. Univariate Call Volume Prediction, SVM Model

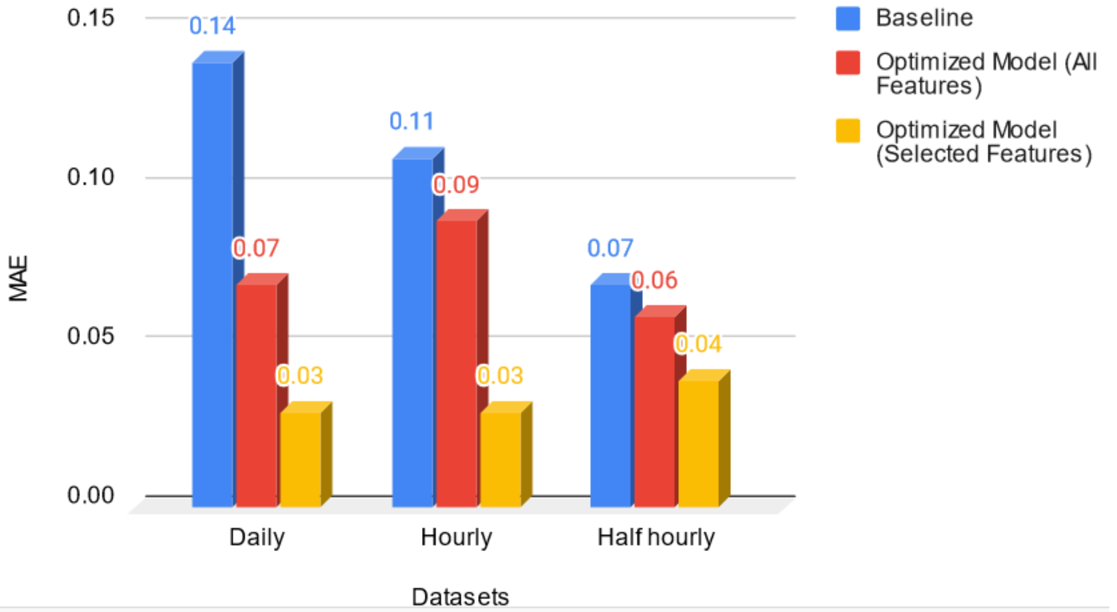

Figure 4.20. Multivariate Avergae Handling Time, SVM Model

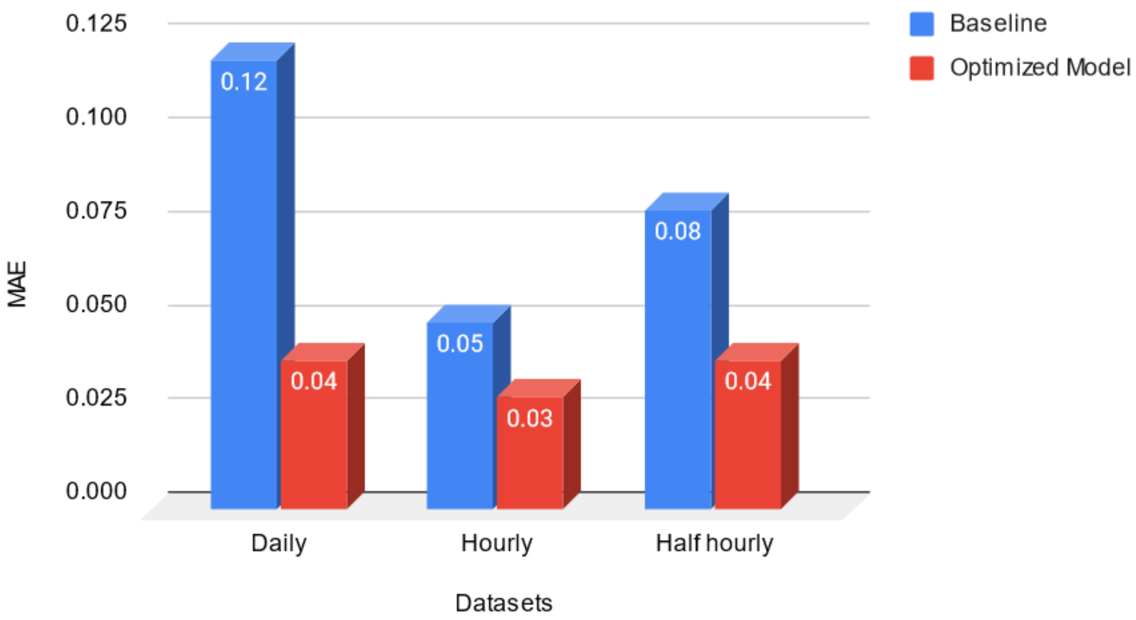

Figure 4.21. Univariate Avergae Handling Time, SVM Model

RNN

Table 4.10. Multivariate observations for RNN model

Multivariate	Datasets	Baseline MAE	Optimized (All Features) MAE	Optimized (All Features) RMSE	Optimized (Selected Features) MAE	Optimized (Selected Features) RMSE
Call volume	Daily	0.25	0.04	0.05	0.03	0.05
	Hourly	0.23	0.04	0.05	0.02	0.03
	Half hourly	0.46	0.04	0.05	0.03	0.04
Average handling time	Daily	0.47	0.07	0.08	0.02	0.04
	Hourly	0.29	0.04	0.05	0.02	0.03
	Half hourly	0.55	0.05	0.06	0.02	0.03

Table 4.11. Univariate observations for RNN model

Univariate	Datasets	Baseline MAE	Optimized MAE	Optimized RMSE
Call volume	Daily	0.13	0.03	0.05
	Hourly	0.17	0.03	0.05
	Half hourly	0.37	0.04	0.06
Average handling time	Daily	0.06	0.04	0.05
	Hourly	0.72	0.04	0.06
	Half hourly	0.25	0.03	0.04

Table 4.10 results demonstrates that multivariate RNN model's performance improved when using optimized models, as evidenced by decreased call volume and average handling time metrics across different datasets. In addition to that, optimised models with selected features outperformedthose with all features. Additionally, for Call volume predictions, hourly dataset had the lowest MAE of 0.02 whereas for AHT, all datasets had the lowest MAE of 0.02. as shown in Figure 4.22 and Figure 4.24, indicating the efficacy of feature selection in enhancing predictive accuracy.

Table 4.11 demonstrates the performance of univariate RNN model and it shows similar trends, with improved model performance observed across various datasets when optimal hyper parameters are utilized. Furthermore, for Call volume predictions, daily and hourly dataset had the lowest MAE of 0.03 whereas for AHT, it was half hourly datasets with MAE of 0.03.These are visualized in Figure 4.23 and Figure 4.25.

These results demonstrate optimized RNN's ability to improve forecasting accuracy across different time intervals.

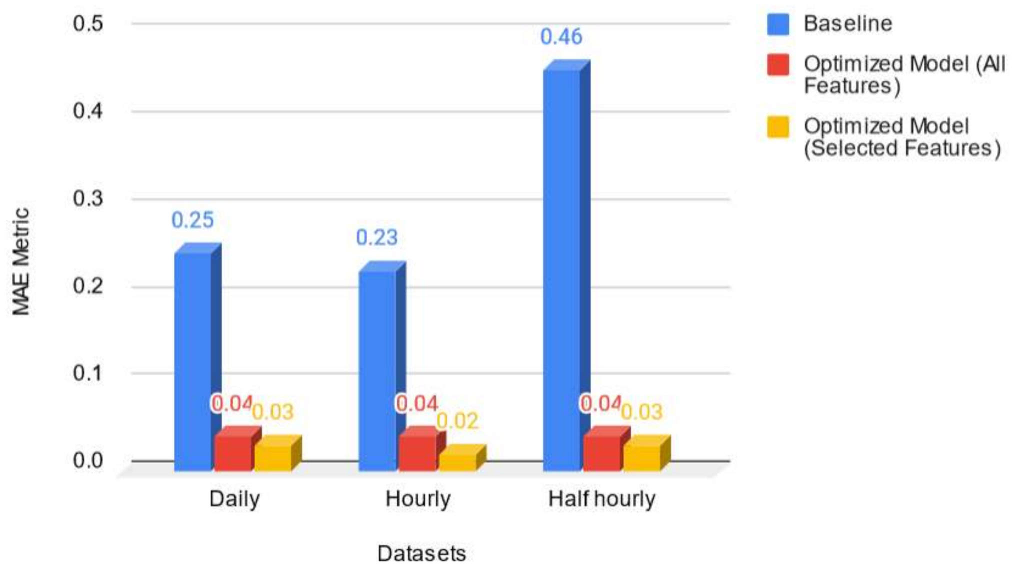

Figure 4. 22. Multivariate Call Volume Prediction, RNN Model

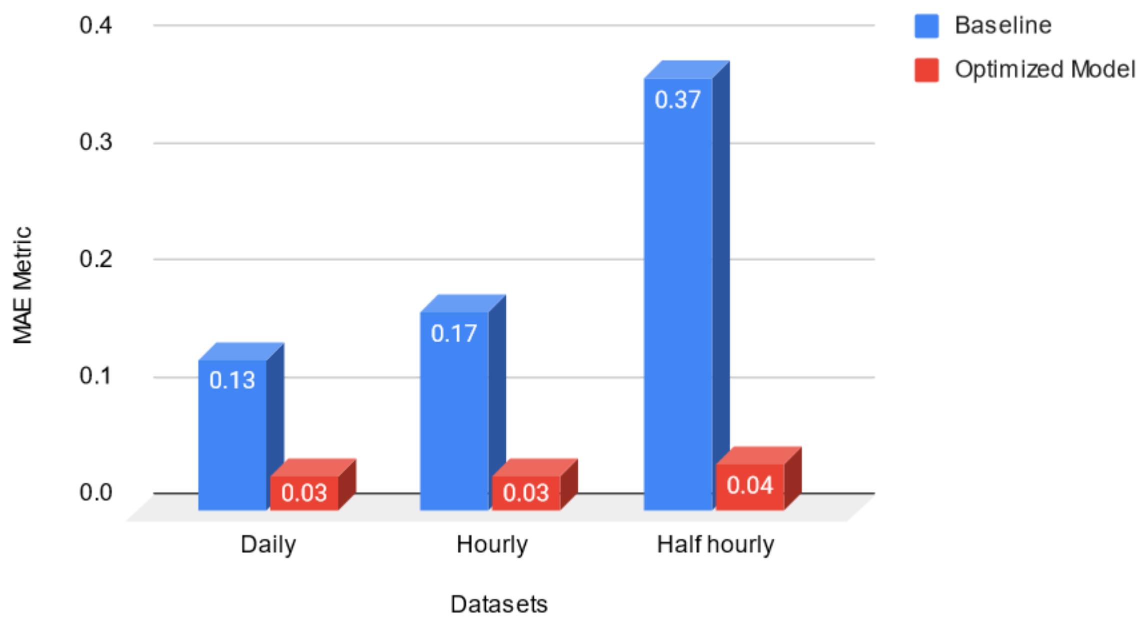

Figure 4. 23. Univariate Call Volume Prediction, RNN Model

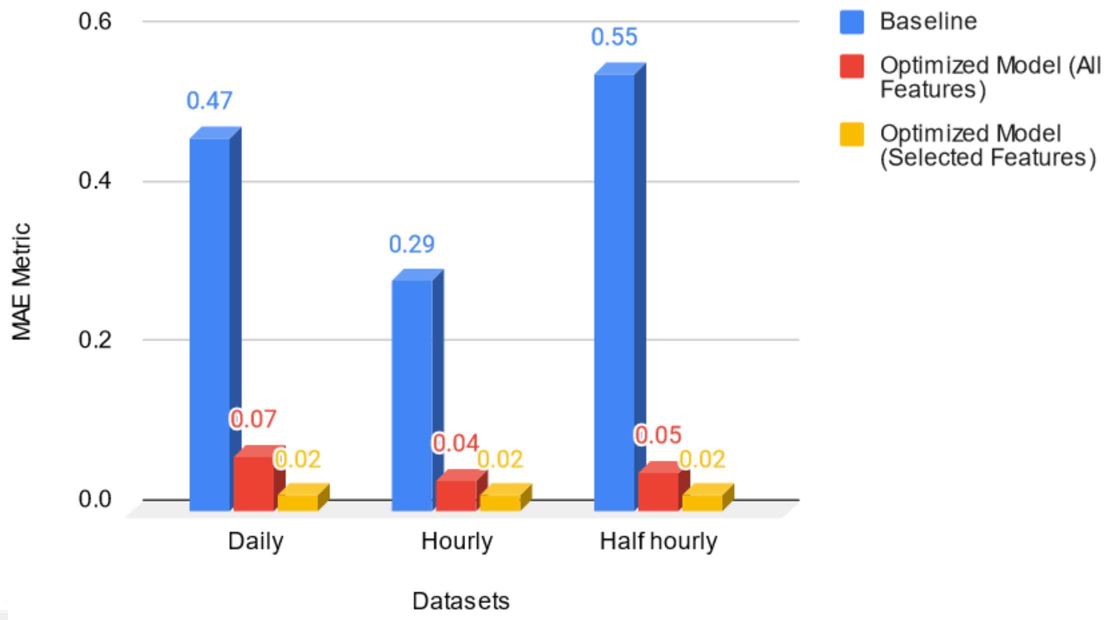

Figure 4. 24. Multivariate Avergae Handling time, RNN Model

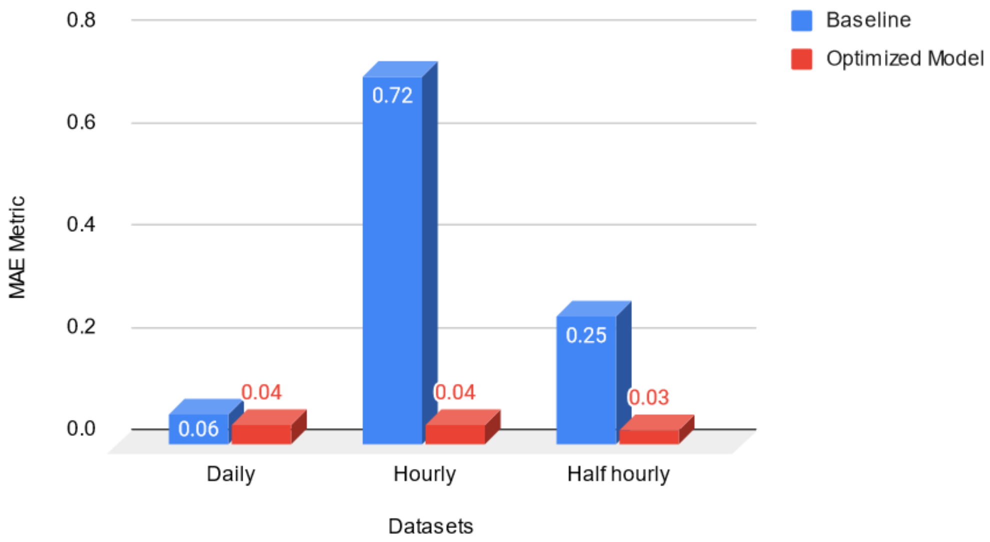

Figure 4. 25. Univariate Avergae Handling time, RNN Model

CNN

Table 4.12. Multivariate observations for CNN model

Multivariate	Datasets	Baseline MAE	Optimized (All Features) MAE	Optimized (All Features) RMSE	Optimized (Selected Features) MAE	Optimized (Selected Features) RMSE
Call volume	Daily	0.31	0.05	0.07	0.02	0.03
	Hourly	0.74	0.03	0.05	0.02	0.03
	Half-hourly	0.74	0.04	0.06	0.03	0.04
Average handling time	Daily	0.37	0.05	0.06	0.03	0.04
	Hourly	0.53	0.04	0.05	0.02	0.03
	Half-hourly	0.78	0.04	0.05	0.02	0.03

Table 4.13. Univariate observations for CNN model

Univariate	Datasets	Baseline MAE	Optimized MAE	Optimized RMSE
Call volume	Daily	0.26	0.02	0.03
	Hourly	0.23	0.02	0.03
	Half hourly	0.88	0.03	0.04
Average handling time	Daily	0.18	0.03	0.04
	Hourly	0.35	0.03	0.04
	Half hourly	0.39	0.03	0.04

When Table 4.12 is interpreted as a result of the tests made with multivariate CNN, the model's performance improves when using an optimized model in comparison with baseline model, as evidenced by decreased values in call volume and average handling time metrics. More specifically, model performance was much better with optimized models with selected features. For call volume predication, daily and hourly dataset had the best estimates MAE of 0.02. For Average Handing Time predications, hourly and half hourly with MAE of 0.02 performed satisfactorily. These results are visualized in Figure 4.26 and Figure 4.28.

Table 4.13 focuses on univariate CNN and compares baseline with optimized model performances. Results show that optimized models improved model performance across all datasets outperforming baseline models. Specifically, for univariate call volume predication, daily and hourly dataset had the best MAE of 0.02. For Average Handing Time predications, all datasets performed satisfactorily with MAE 0.03 (see Figure 4.27 and Figure 4.29). These findings highlight optimised CNN's potential to enhance call arrival predictions across different time intervals.

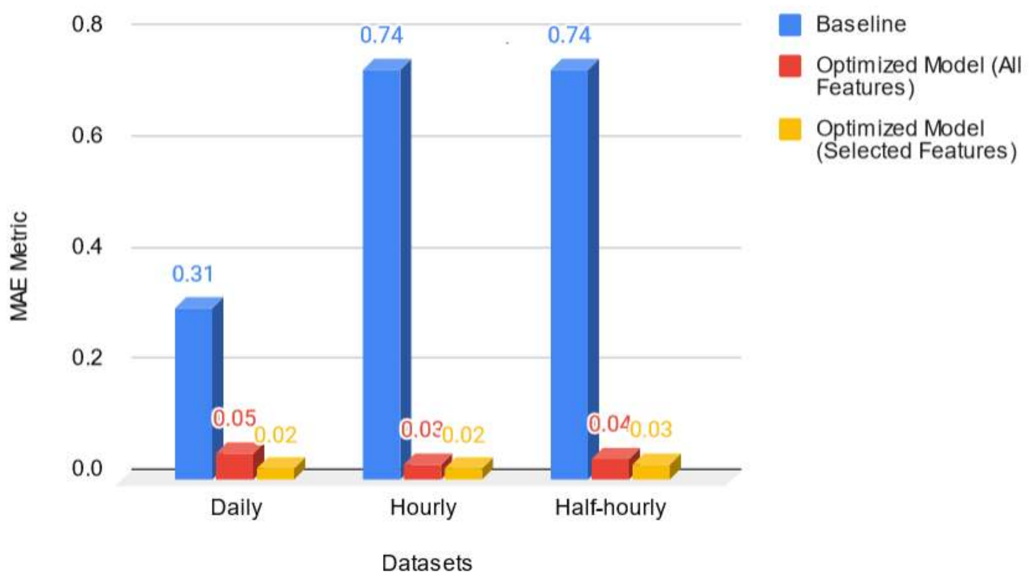

Figure 4. 26. Multivariate Call Volume Prediction, CNN Model

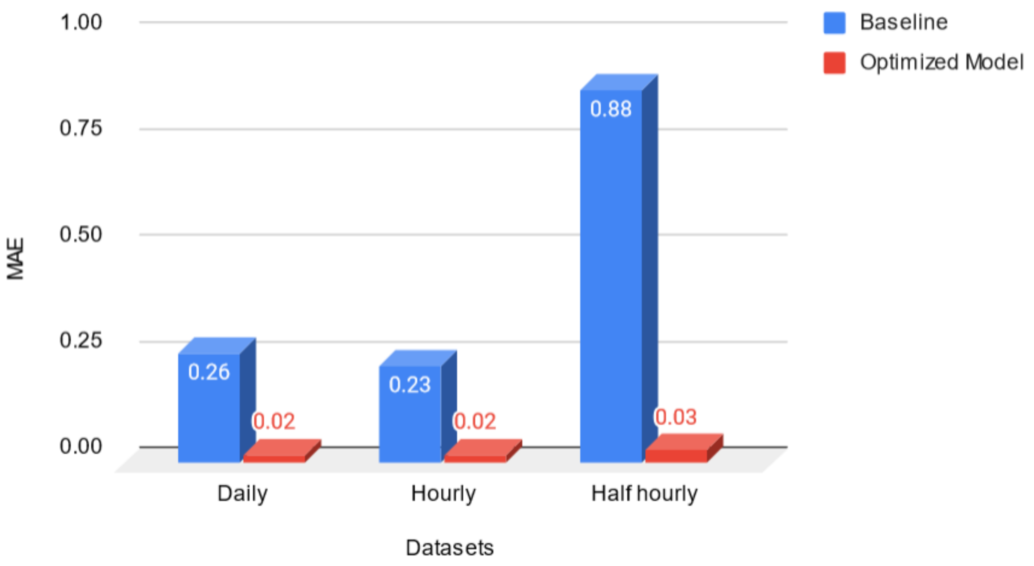

Figure 4.27. Univariate Call Volume Prediction, CNN Model

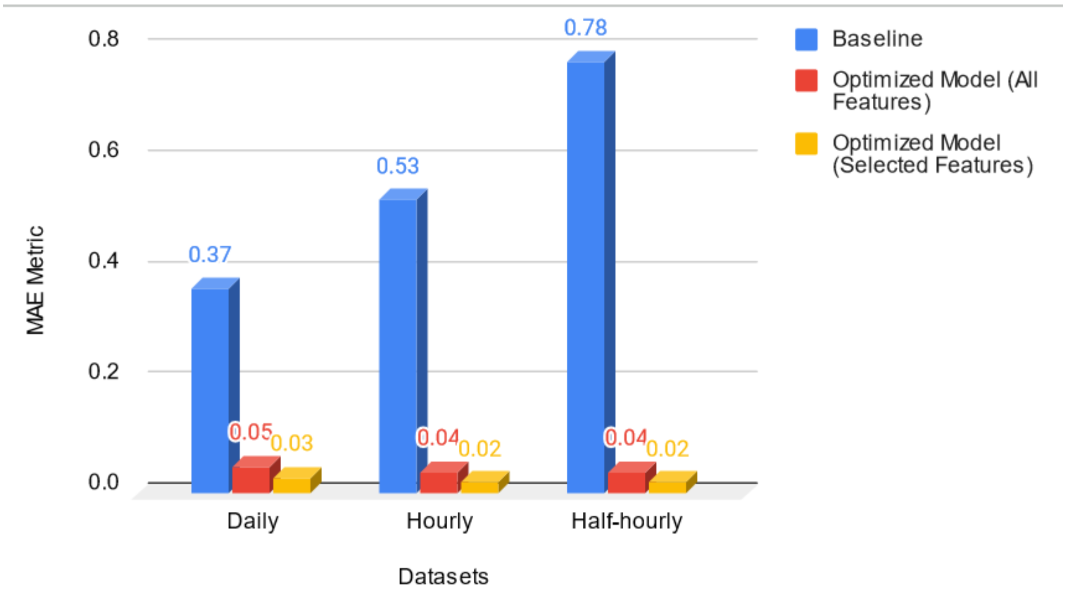

Figure 4.28. Multivariate Avergae handling time, CNN Model

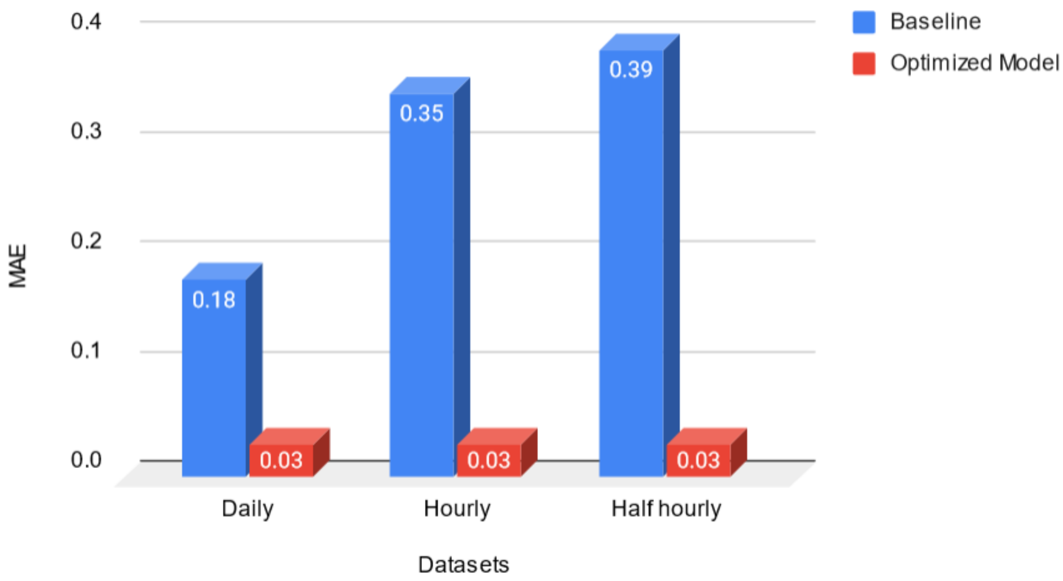

Figure 4.29. Univariate Average Handling Time Prediction, CNN Model

LSTM

Table 4.14. Multivariate observations for LSTM model

Multivariate	Datasets	Baseline MAE	Optimized (All Features) MAE	Optimized (All Features) RMSE	Optimized (Selected Features) MAE	Optimized (Selected Features) RMSE
Call volume	Daily	0.44	0.03	0.04	0.02	0.03
	Hourly	0.16	0.05	0.06	0.02	0.03
	Half hourly	0.17	0.04	0.05	0.03	0.04
Average handling time	Daily	0.26	0.07	0.08	0.02	0.03
	Hourly	0.37	0.22	0.25	0.02	0.03
	Half hourly	0.32	0.05	0.06	0.02	0.03

Table 4.15. Univariate observations for LSTM model

Univariate	Datasets	Baseline	Optimized Model MAE	Optimized Model RMSE
Call volume	Daily	0.08	0.03	0.04
	Hourly	0.19	0.02	0.03
	Half hourly	0.24	0.03	0.04
Average handling time	Daily	0.06	0.03	0.04
	Hourly	0.17	0.03	0.04
	Half hourly	0.18	0.03	0.04

In Table 4.14. Multivariate observations for LSTM model, when results for multivariate LSTM were evaluated, reductions in call volume and average handling time metrics were observed with optimized models in comparison with baseline models. Furthermore, optimized models with selected features performed better than optimized models with all features. Specifically, daily and hourly datasets MAE of 0.02 performed optimally for call volume forecasting whereas all datasets performed well for AHT with MAE of 0.02 as visualized in Figure 4.30 and Figure 4.32.

Table 4.15. Univariate observations for LSTM model presents result for univariate LSTM model. These results reveal a reduction in MAE values with the use optimized models versus baseline models. In summary, hourly datasets performed satisfactorily for call volume forecasting with MAE of 0.02 whereas all datasets performed well MAE of 0.03 for AHT. This is visually seen in Figure 4.31 and Figure 4.33.

Overall, the results section highlights the positive impact of optimization on all datasets but hourly intervals consistently scored.

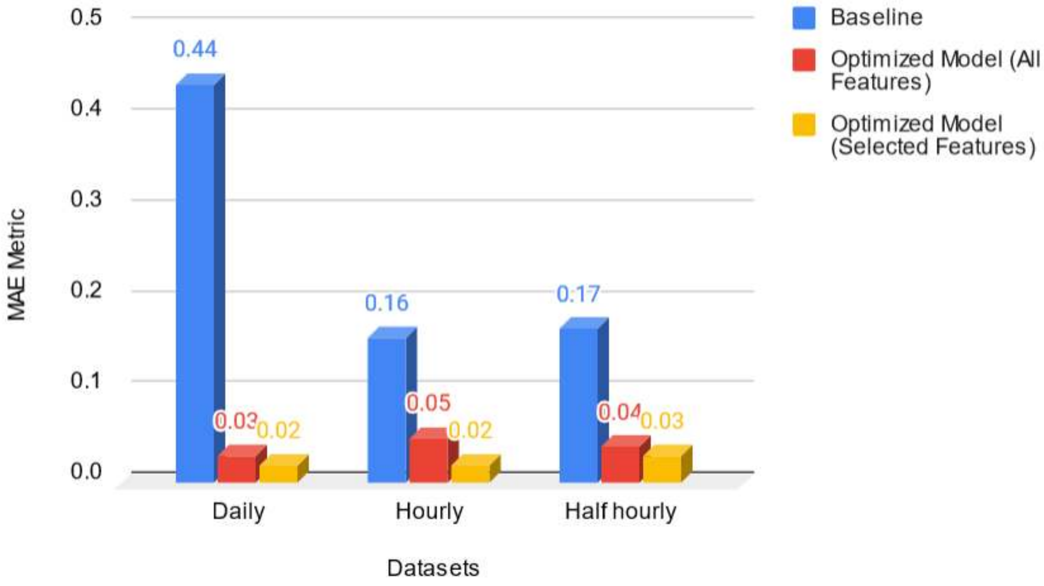

Figure 4.30. Multivariate Call Volume Prediction, LSTM Model

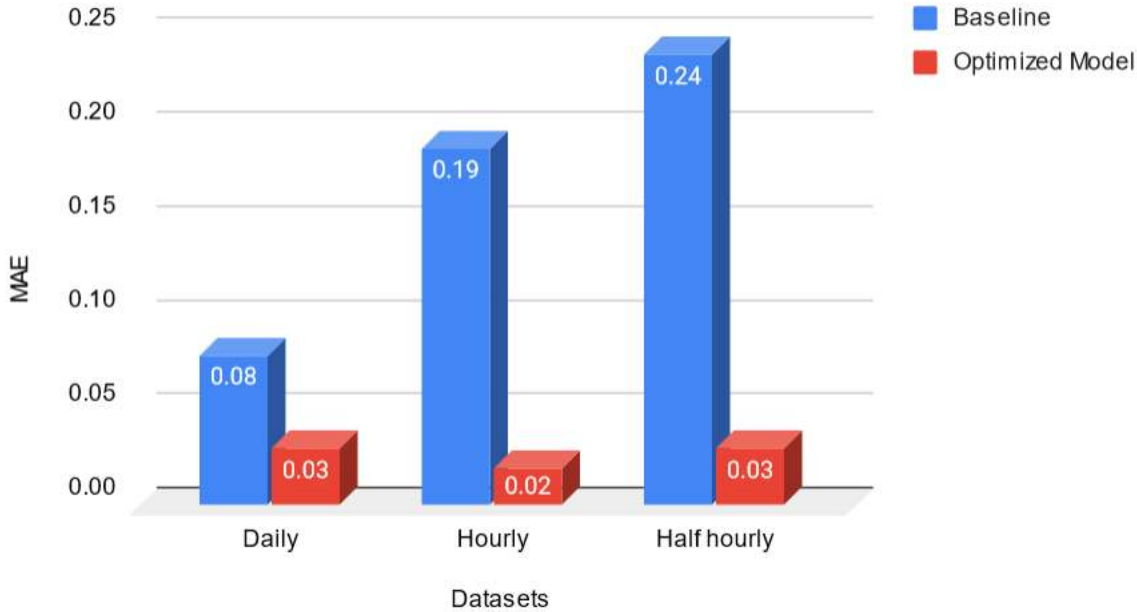

Figure 4.31. Univariate Call Volume Prediction, LSTM Model

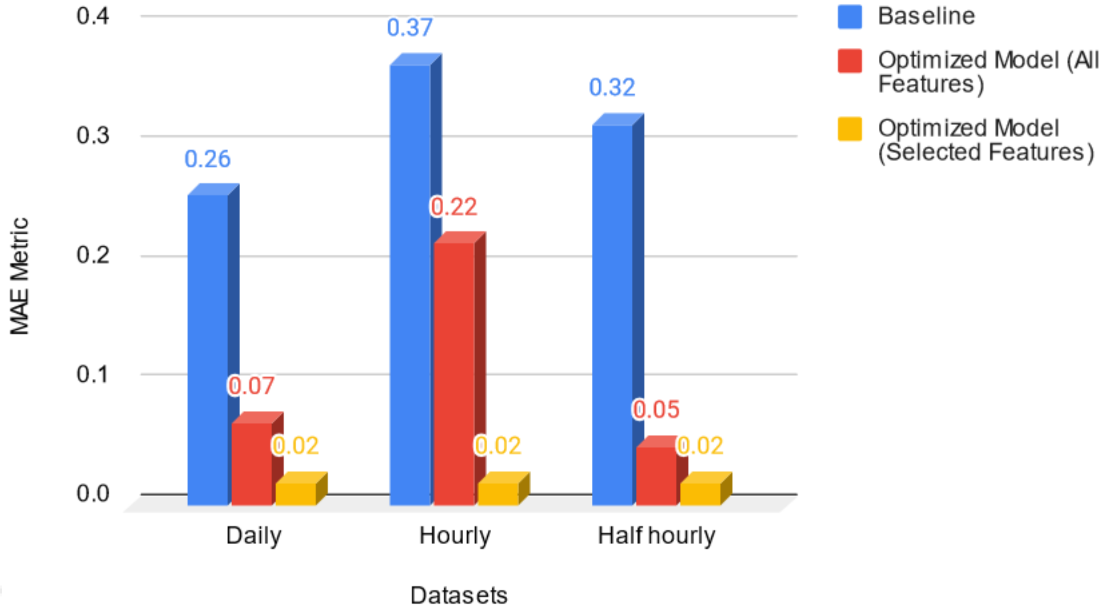

Figure 4.32. Multivariate Avergae Handling Time, LSTM Model

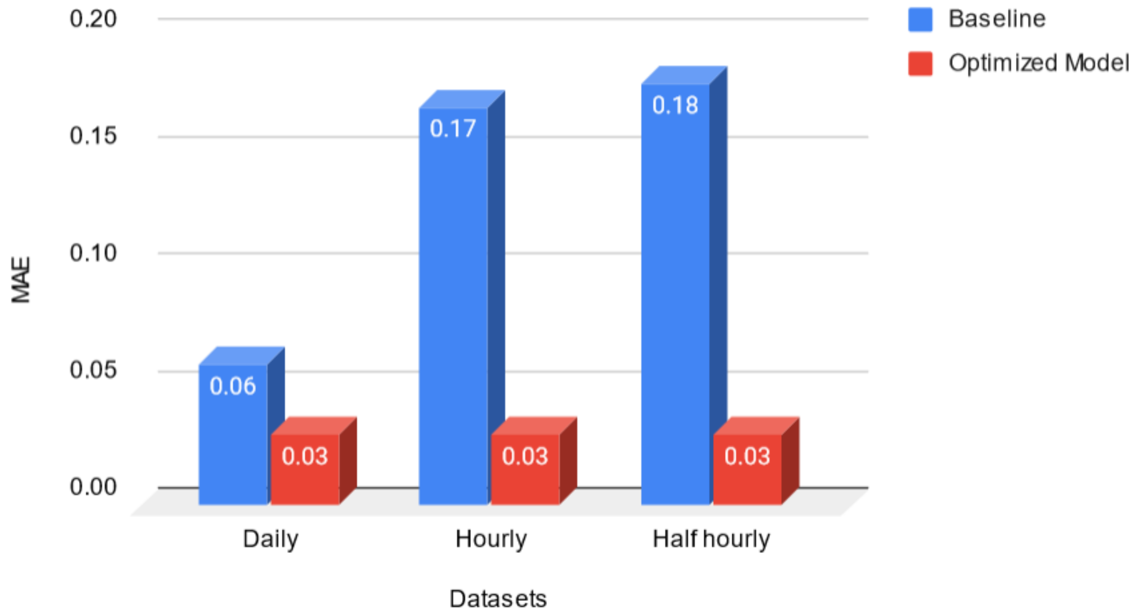

Figure 4. 33. Univariate Average Handling Time, LSTM Model

4.1.6. Multivariate Observations for Call Volume and AHT Across All Models

This section presents a detailed analysis of call volume predictions using various multivariate optimised models, including MLP, RF, SVM, RNN, LSTM, and CNN. The aim is to understand how these models perform when taking multiple features into account and to identify the best-performing approaches. Call Volume Multivariate observations for all models are depicted in Table 4.16 whereas model predictions for AHT are shown in Table 4.17.

Table 4.16. Call Volume Multivariate observations for all models

Dataset	Models	Baseline MAE	Optimized Model (All Features) MAE	Optimized Model (Selected Features) MAE
Daily	MLP	0.05	0.03	0.02
	RF	0.04	0.03	0.02
	SVM	0.12	0.08	0.03
	RNN	0.25	0.04	0.03
	CNN	0.31	0.05	0.02
	LSTM	0.44	0.03	0.02
hourly	MLP	0.18	0.03	0.02
	RF	0.06	0.04	0.02
	SVM	0.08	0.07	0.03
	RNN	0.23	0.04	0.02
	CNN	0.74	0.03	0.02
	LSTM	0.16	0.05	0.02
Half hourly	MLP	0.05	0.04	0.03
	RF	0.06	0.03	0.03
	SVM	0.07	0.06	0.04
	RNN	0.46	0.04	0.03
	CNN	0.74	0.04	0.03
	LSTM	0.17	0.04	0.03

For call volume prediction, with daily intervals, when using optimized models with all features, MLP, RF, and LSTM exhibited the lowest MAE of 0.03, whereas with feature selection, MLP, RF, CNN and LSTM showcased improved performance with an MAE of 0.02. Whereas for AHT, optimized models with all features had MLP achieved the lowest MAE of 0.03. However, with feature selection, MLP, RNN, and LSTM models exhibited improved performance with an MAE of 0.02. this is visualised in Figure 4.34 and Figure 4.35.

In the hourly dataset, for call volume prediction, optimized models with all features had MLP and CNN performing notably well with an MAE of 0.03, whereas all models performed satisfactorily after feature selection except SVM. Similarly, for AHT, MLP performed notably well with an MAE of 0.03 when using optimized models with all features, and after feature selection all models showed satisfactory performance except for RF and SVM as seen in Figure 4.36 and Figure 4.37.

Lastly in half-hourly dataset for call volume prediction, while using optimized models with all features, RF demonstrated strong performance with an MAE of 0.03, and after feature selection, all models displayed robust performance with an MAE of 0.03 except SVM. However, for AHT forecasts, MLP, RF and CNN demonstrated strong performance with an MAE of 0.03 when using all features, and after feature selection, all models performed robustly with an MAE of 0.03 except for SVM visualised in Figure 4.38 and Figure 4.39.

For call volume prediction, hourly dataset has more optimal performance whereas for AHT, hourly and half hourly datasets had more acceptable performances.

Table 4.17. AHT Multivariate observations for all models

Dataset	models	Baseline MAE	Optimized (All Features) MAE	Optimized (Selected Features) MAE
Daily	MLP	0.06	0.03	0.02
	RF	0.06	0.05	0.04
	SVM	0.14	0.07	0.07
	RNN	0.47	0.07	0.02
	CNN	0.37	0.05	0.03
	LSTM	0.26	0.07	0.02
hourly	MLP	0.16	0.03	0.02
	RF	0.09	0.05	0.04
	SVM	0.11	0.09	0.06
	RNN	0.23	0.04	0.02
	CNN	0.53	0.04	0.02
	LSTM	0.37	0.22	0.02
Half hourly	MLP	0.08	0.04	0.02
	RF	0.06	0.04	0.02
	SVM	0.07	0.06	0.06
	RNN	0.46	0.04	0.03
	CNN	0.78	0.04	0.02
	LSTM	0.32	0.05	0.02

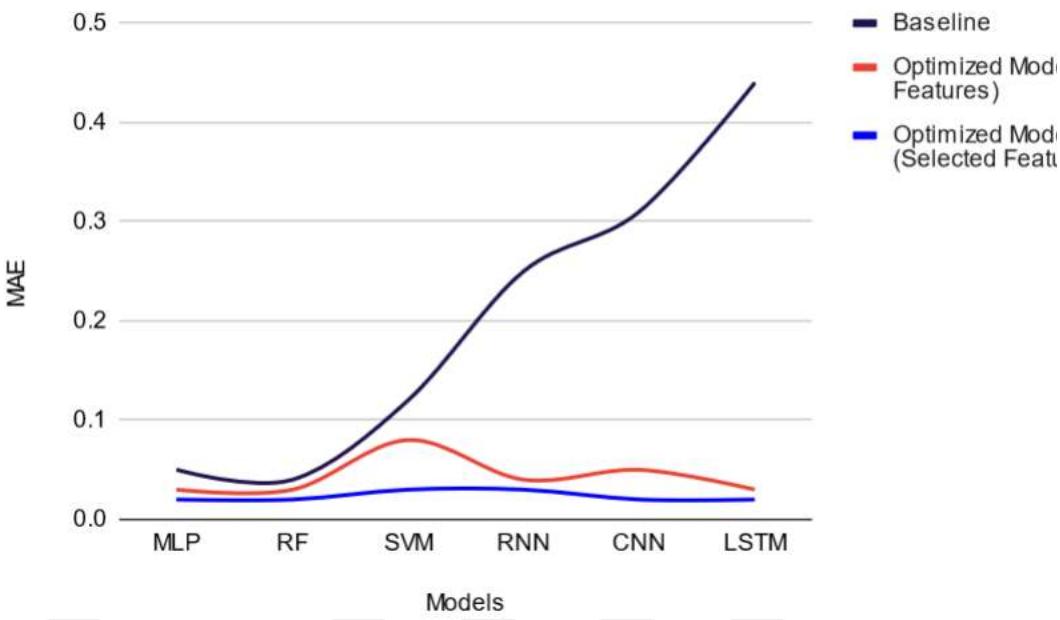

Figure 4.34. Models' results for daily multivariate models for Call volume

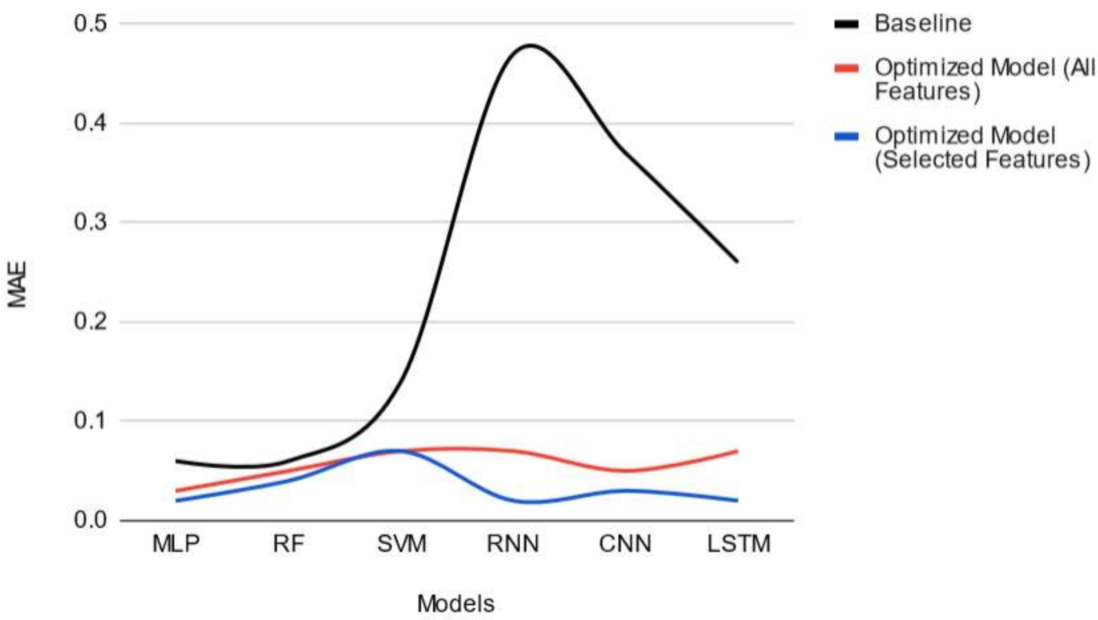

Figure 4.35. Models' results for daily multivariate models AHT

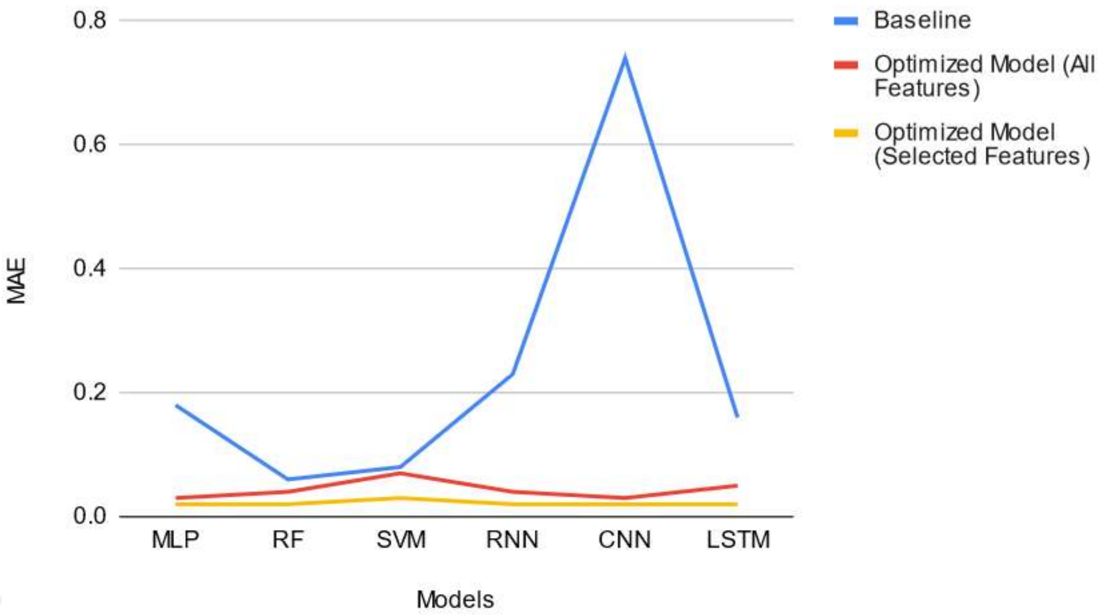

Figure 4.36. Models' results for hourly multivariate models for Call volume

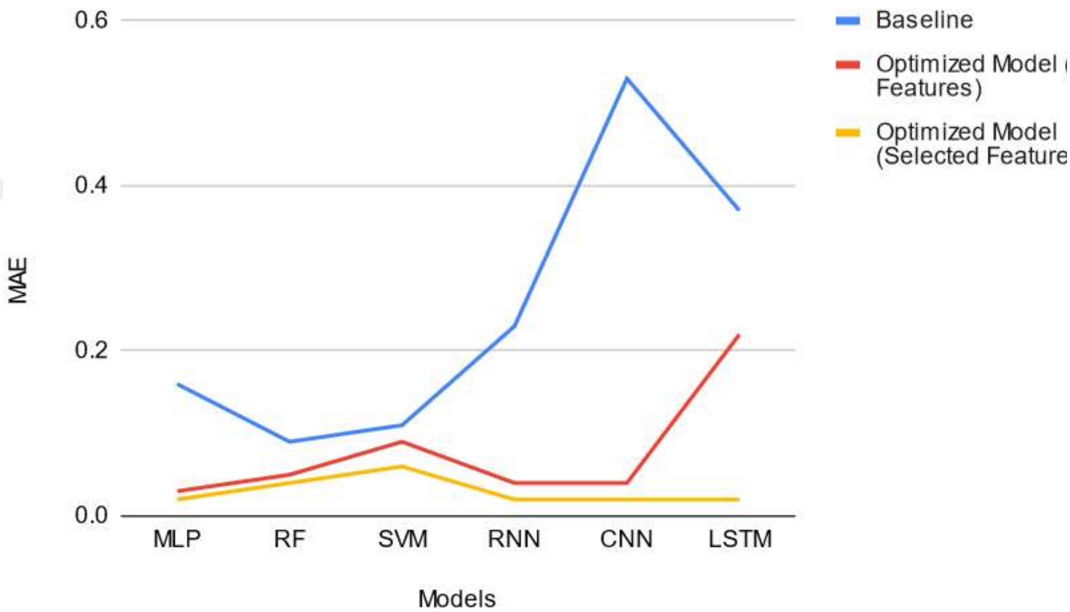

Figure 4.37. Models' results for hourly multivariate models AHT

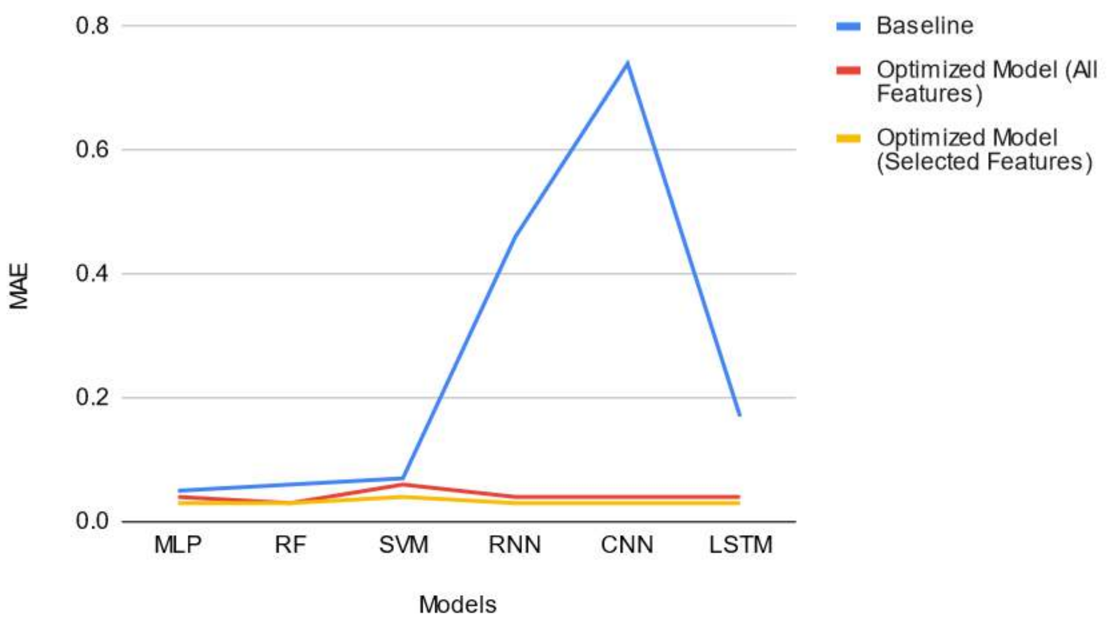

Figure 4.38. Models' results for half hourly multivariate for Call volume

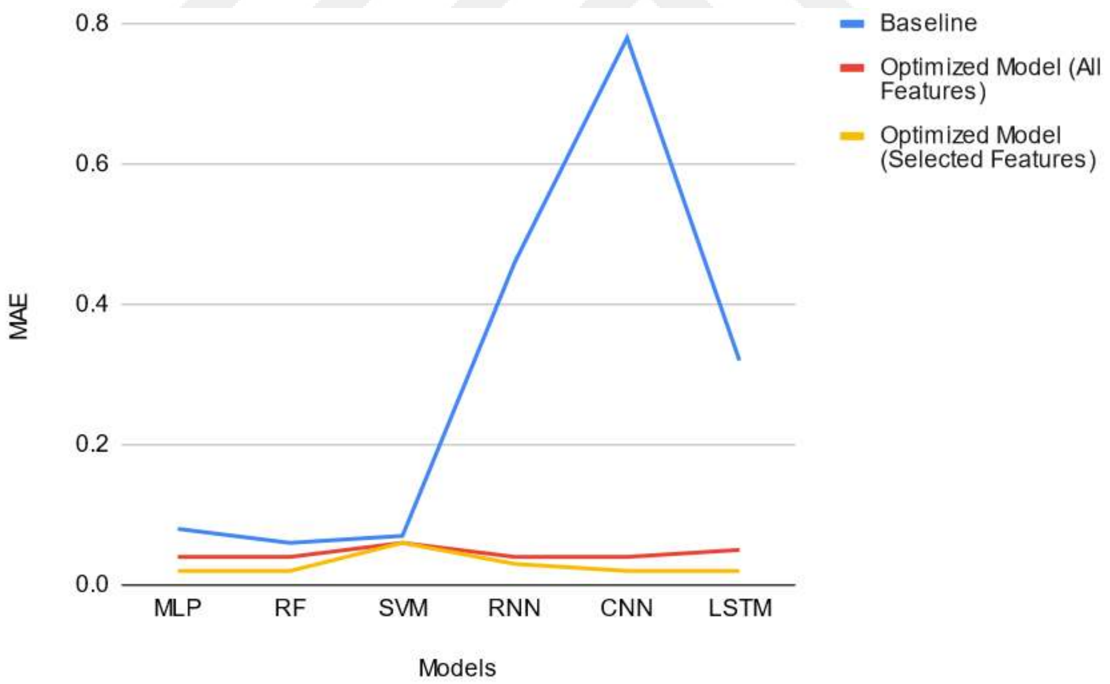

Figure 4.39. Models' results for half hourly multivariate models AHT

4.1.7. Univariate Observations for Call Volume and AHT Across All Models

Table 4.18 and Table 4.19presents the error forecasting results for univariate model predictions.

Table 4.18. Univariate Call volume model results

Dataset	Models	Baseline MAE	Optimized MAE
Daily	MLP	0.06	0.03
	RF	0.05	0.03
	SVM	0.14	0.04
	RNN	0.13	0.03
	CNN	0.26	0.02
	LSTM	0.08	0.03
hourly	MLP	0.07	0.03
	RF	0.04	0.03
	SVM	0.09	0.03
	RNN	0.17	0.03
	CNN	0.23	0.03
	LSTM	0.19	0.03
Half hourly	MLP	0.05	0.04
	RF	0.05	0.03
	SVM	0.07	0.04
	RNN	0.37	0.04
	CNN	0.88	0.03
	LSTM	0.24	0.03

Considering daily observations,for call volume forecasting, the CNN architecture achieved the best performance, with the MAE decreasing from 0.26 to 0.02. The MAEs for MLP, RF, SVM, RNN, CNN, and LSTM models have similar errors magnitude (0.03). The increase in error reductions highlights the improvement in MAE after optimization. Additionally, DL models showed the highest responsiveness to the optimized methods, greatly enhancing performance beyond strong baseline ML models like MLP and RF, especially true for CNN. For AHT, CNN and LSTM exhited the lowest error rate (0.03) and the rest of the models followed suit with MAE of 0.04.

When evaluating results for univariate hourly intervals, for call volume predictions, all models achieved an error value of 0.03. MLP and RF had strong baseline performances. For AHT, SVM,CNN and LSTM had the lowest MAE (0.03). the rest o the models scored MAE of 0.04.

Results for half-hourly intervals, for call volume predictions, RF, CNN and LSTM had the lowest MAE of 0.03. Overall, all models responded positively to optimization techniques used in the study. For AHT, all models except SVM achieve the least MAE (0.03).

Table 4.19. Univariate AHT model results

Dataset	Models	Baseline MAE	Optimized MAE
Daily	MLP	0.05	0.04
	RF	0.06	0.04
	SVM	0.12	0.04
	RNN	0.06	0.04
	CNN	0.18	0.03
	LSTM	0.06	0.03
hourly	MLP	0.06	0.04
	RF	0.06	0.04
	SVM	0.05	0.03
	RNN	0.72	0.04
	CNN	0.35	0.03
	LSTM	0.17	0.03
Half hourly	MLP	0.07	0.03
	RF	0.06	0.03
	SVM	0.08	0.04
	RNN	0.25	0.03
	CNN	0.39	0.03
	LSTM	0.18	0.03

4.1.8. Overall model performance

For a clearer overview of each models' superior performances across various datasets, Table 4.21 and Table 4.22 provided a summary of how often each optimized model outperformed before feature selection and after feature selection respectively, and for univariate model performance respectively. Table 4.20 provided a summary of how often each optimized univariate model outperformed across different time intervals.

Table 4.21. Summary of optimized multivariate models before feature selection that exhibited superior performance with specific datasets, marked by an "x."

DATA	LSTM	CNN	SVR	RF	MLP	RNN
Multivariate Call volume						
Daily	x			x	x	
Hourly		x			x	
Half-hourly				x		
Multivariate AHT						
Daily					x	
Hourly					x	
Half-hourly		x		x	x	x
Total	**1**	**2**	**0**	**3**	**5**	**1**

Table 4.22. Summary of optimized multivariate models after feature selection that exhibited superior performance with specific datasets, marked by an "x."

DATA	LSTM	CNN	SVR	RF	MLP	RNN
Multivariate Call volume						
Daily	x	x		x	x	
Hourly	x	x		x	x	x
Half-hourly	x	x		x	x	x
Multivariate AHT						
Daily	x				x	x
Hourly	x	x		x	x	x
Half-hourly		x		x		
Multivariate Total	**5**	**5**	**0**	**5**	**5**	**4**

Table 4.23. Univariate optimised model performance summary marked with “x”.

Univariate Call volume						
DATA	LSTM	CNN	SVR	RF	MLP	RNN
Daily		x				
Hourly	x	x		x	x	x
Half-hourly		x		x		
Univariate AHT						
Daily	x	x				
Hourly	x	x	x			
Half-hourly	x	x		x	x	x
Univariate total	**4**	**6**	**1**	**3**	**2**	**2**

Table 4.22Table 4.21 and Table 4.22 summarize the performance of multivariate models before and after feature selection, highlighting how often each model achieved superior performance. An "x" denotes superior performance for specific datasets. Before feature selection, other models showed varying degrees of success, with MLP consistently performing well across different time frames. However, after feature selection, LSTM, CNN, RF, and MLP each excelled consistently (a total of five times) highlighting their gained robustness and effectiveness. RNN followed closely, excelling four times. This overall assessment underlines the significant performance improvements achieved through feature selection, particularly for LSTM, CNN, RF, and MLP models.

Table 4.23 summarizes the performance of univariate predictions. CNN demonstrated the highest frequency of superior performance with six instances of excellence, followed by LSTM with four instances. The other models showed varying levels of effectiveness: RF with three instances, MLP and RNN with two instances each, and SVM with one instance. This highlights CNN's dominance in univariate predictions, with LSTM also showing strong performance.

4.2. Discussions

When evaluating the performance of various DL and ML models - RF, MLP, SVM, RNN, CNN, LSTM - across different datasets, the results showed that baseline models provided a starting point. Bayesian-optimized models demonstrated significant improvements in performance across all datasets for all models over the baselines and those reported in the literature. Additionally, multivariate models transitioning from optimized models with all features to those with selected features continuously yielded notable error reductions. Notably, when comparing baseline and Bayesian-optimized models for call volume prediction using hourly datasets, the CNN model showed the most substantial improvement. With all features considered, the error reduced by 95.95%. Utilizing selected features further improved performance, achieving an error reduction of up to 97.30%. Similarly, for Average Handling Time (AHT) prediction with half-hourly datasets, the CNN model showed significant improvements: 94.87% error reduction for all features and 97.44% for selected features. In univariate models focusing on call volume prediction, the CNN model with half-hourly datasets achieved a remarkable 96.59% reduction in Mean Absolute Error (MAE). For AHT forecasting, the RNN model using hourly datasets reached a substantial 94.44% error reduction. Making DL models the more positively impacted by feature selection generally.

Also the findings highlighted how feature selection notably boosted all model performances enabling them to attain competitive outcomes. For example, before feature selection, other models showed varying degrees of success, with MLP consistently performing well across different time frames. However, after feature selection, LSTM, CNN, RF, and MLP each excelled consistently (a total of five times each) highlighting their gained robustness and effectiveness. These findings suggest that Bayesian optimization effectively refines models, leading to enhanced predictive performance, as noted by Garnett (2023). Furthermore, feature selection enhances the performance of multivariate optimized models, aligning with the findings of (Jiang & Wang, 2016), who emphasized the efficiency of feature selection techniques in reducing data by identifying the most relevant subset of input features related to the target variable.

Furthermore, the results highlighted the superiority of utilizing architectures like LSTM, CNN, RF and MLP when dealing with multivariate data. And for univariate predictions, CNN stood out followed by LSTM, demonstrating their effectiveness in capturing single-variable relationships. This is in line with studies done by Hochreiter & Schmidhuber (1997); BALLOUCH et al. (2021) & Bharadiya(2023) as they emphasized that DL models leverage their sequential processing capabilities to capture temporal dependencies in the data, extracting meaningful patterns and relationships from multiple variables simultaneously. Their efficiency can be explained by the recurrent connections that allow them to access the entire history of previous time series values (Borovykh & Bohte, 2018).

Findings also highlighted the superiority of multivariate forecasts over univariate, particularly when coupled with feature selection techniques. Because multivariate forecasting considers multiple variables simultaneously, capturing complex relationships and dependencies in the data is possible. (Rausch et al., 2022), noted that this boosts prediction accuracy by leveraging a broader range of information compared to univariate methods, which only consider one variable at a time.

The findings suggested that certain models, like RF and MLP, consistently perform well across various granularity levels, even in baseline predictions. This demonstrates that even models with strong baseline performance can benefit from BO optimization and feature selection. This insight informs practitioners and researchers about the critical role of model tuning in maximizing the predictive accuracy.

However, SVR exhibited excellence particularly with univariate models but performed poorly with multivariate models, indicating potential in specific scenarios despite not consistently outperforming other models. This observation is supported by (Zhang & O'Donnell, 2019), who identified notable weaknesses in SVMs, such as sensitivity to parameters, challenges in parameter selection, and algorithmic complexity. These factors likely contributed to the varied performance of SVR in our study.

The evaluation of training times for the high-performing models - RF, MLP, CNN, and LSTM - revealed a critical trade-off between predictive accuracy and computational efficiency. In the daily dataset, RF was the most time-efficient, completing training in about 5 minutes, followed by MLP at 12 minutes, CNN at 15 minutes, and LSTM at 20 minutes due to its sequential processing. As dataset size increased, the differences became more pronounced; RF took around 30 minutes for the hourly dataset, while LSTM required about 2 hours. In the half-hourly dataset, RF completed training in 1 hour, compared to 5 hours for LSTM. These variations highlight the importance of model selection in time-sensitive environments like call centers, where quicker model updates can enhance operational efficiency. While LSTM models may yield higher accuracy for time-series data, their longer training times can limit practical application, emphasizing the need to balance accuracy with computational speed for optimal resource use.

While the findings of this study align with existing literature on call arrival prediction, it also advances the field by systematically integrating Bayesian Optimization with feature selection, highlighting its substantial impact on improving model accuracy across various ML and DL models. This approach, particularly applied to real-world call center data, presents a refined methodology that has not been extensively explored in previous research.

In summary, the integration of Bayesian optimization and feature selection enhanced machine learning model performance, particularly for complex tasks where manual tuning is impractical. Specifically, BO efficiently identifies optimal hyperparameters, while feature selection pinpointed to the most important features, leading to reduced MAE across models. This approach improved predictive accuracy and supports better resource allocation and operational efficiency in call centers, aligning with findings by Ballouch et al. (2021) on improved staffing decisions and service planning, ultimately boosting operational efficiency and customer satisfaction.

Despite the valuable insights gained, this study has certain limitations that should be acknowledged. These include constraints related to sample size, the inherent complexity of call center data, and the specific parameters chosen for optimization. Future research could address these limitations by expanding sample sizes, incorporating a broader range of variables, and further refining optimization techniques. Overall, this study lays a solid foundation for advancing state-of-the-art predictive analytics and operational optimization in call center environments, opening new avenues for research and practical applications

5. CONCLUSIONS

This study proposed integrating BO and feature selection with ML and DL models for call arrival prediction, utilizing models such as LSTM, CNN, RNN, RF, SVM, and MLP across daily, hourly, and half-hourly datasets. A number of conclusions could be sought from the results.

Initially, BO was employed to identify the optimal hyperparameters for each model. The models optimized using BO techniques consistently outperformed their baseline counterparts across all datasets, showing significant improvements in predictive accuracy. This was particularly evident in deep learning models, with CNN and LSTM demonstrating strong responsiveness to BO. Moreover, ML models especially MLP and RF had the strongest baseline performances but also benefitted from BO integration. Depending on the dataset, the performance gain obtained from BO optimization ranged between 12% -83% for ML models and 40% - 96% for DL models for both Call volume and AHT predictions for multivariate models and for univariate models a percentage improvement that ranges from 20% to71% for ML and 50% to 96.6%.

The predictor variables CallCount, TalkCount, and TalkTimeSec emerged as primary predictors for call volume forecasting. For AHT, features such as Acht, TalkTimeSec, and Answered30PSec were identified as crucial factors. These variables significantly improve the performance of call arrival predictions. Conversely, the impact of the other predictor variables on call volume and AHT predictions can vary. Depending on their combination, these predictors may enhance, have a negligible effect, or even deteriorate the prediction model's performance.

Feature selection further boosted all model optimized performances, enabling them to attain competitive outcomes. Transitioning from baseline models to those with selected features further reduced errors significantly, notably with a range of 14% to 88% error reduction for ML models and 92% to 97% error reduction for DL models. Due to feature selection, more multivariate models had better performance compared to univariate ones.

The study also highlighted the importance of employing shorter time intervals for improved forecasting precision in call centers. More consistent predictions were realized with hourly and half hourly intervals. These insights are pivotal for optimizing resource allocation and operational efficiency.

Among the set of machine learning methods, MLP-based prediction models followed by RF exhibit better performance (i.e. yield lower MAE) regardless of the utilized datasets. Among the DL models, LSTM and CNN exhibited the lowest MAE. These findings emphasized the importance of selecting appropriate models to enhance prediction accuracy.

To enhance the practicality of these prediction models, developing an application that integrates the most accurate models could enable real-time predictions for new data. Moreover, while the proposed BO with feature selection has been thoroughly evaluated, further experiments on diverse datasets with varying characteristics could extend its promising potential for improving call arrival predictions. Additionally, exploring advancements in optimization algorithms could continue to improve model performance

The consistent superiority of the optimized models underlines the robustness of employing hyperparameter optimization techniques with feature selection and highlights their practical significance in addressing challenges posed by call arrival data. The research provides valuable insights into the need for optimizing models handling temporal data, showing the effectiveness of these methodologies in enhancing performance within both DL and ML frameworks.

In summary, integrating BO with feature selection proved instrumental in optimizing model performance, offering a systematic approach with broad implications for improving DL and ML applications in call center operations. This study emphasizes the role of advanced modeling techniques and optimization methods in improving the accuracy and efficiency of predictive models, ultimately contributing to more effective call center management.

REFERENCES

Abbasimehr, H., Shabani, M., & Yousefi, M. (2020). An optimized model using LSTM network for demand forecasting. *Computers and Industrial Engineering*, *143*(July 2019), 106435. https://doi.org/10.1016/j.cie.2020.106435

Abdoli, G., MehrAra, M., & Ebrahim Ardalani, M. (2020). Comparing the Prediction Accuracy of Lstm and Arima Models for Time-Series With Permanent Fluctuation. *Gênero & Direito*, *9*(2), 314–339. https://doi.org/10.22478/ufpb.2179-7137.2020v9n2.50782

Abut, F., Akay, M. F., & George, J. (2016). Developing new VO2max prediction models from maximal, submaximal and questionnaire variables using support vector machines combined with feature selection. *Computers in Biology and Medicine*, *79*(September), 182–192. https://doi.org/10.1016/j.compbiomed.2016.10.018

Ahmed, A., Sivarajah, U., Irani, Z., Mahroof, K., & Charles, V. (2024). Data-driven subjective performance evaluation: An attentive deep neural networks model based on a call centre case. *Annals of Operations Research*, *333*(2–3), 939–970. https://doi.org/10.1007/s10479-022-04874-2

Albelwi, S., & Mahmood, A. (2017). A framework for designing the architectures of deep Convolutional Neural Networks. *Entropy*, *19*(6). https://doi.org/10.3390/e19060242

Albrecht, T., Rausch, T. M., & Derra, N. D. (2021). Call me maybe: Methods and practical implementation of artificial intelligence in call center arrivals' forecasting. *Journal of Business Research*, *123*(January 2020), 267–278. https://doi.org/10.1016/j.jbusres.2020.09.033

Andrade, R., & Moazeni, S. (2023). Transfer rate prediction at self-service customer support platforms in insurance contact centers. *Expert Systems with Applications*, *212*, 118701. https://doi.org/https://doi.org/10.1016/j.eswa.2022.118701

Awad, M., Khanna, R., Awad, M., & Khanna, R. (2015). Support vector regression. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, 67–80.

BALLOUCH, M., AKAY, F., ERDEM, S., TARTUK, M., NURDAĞ, T. F., & YURDAGÜL, H. H. (2021). Forecasting Call Center Arrivals Using Machine Learning. *Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *4*(1), 96–101. https://doi.org/10.47495/okufbed.824870

Barrow, D. K. (2016). Forecasting intraday call arrivals using the seasonal moving average method. *Journal of Business Research*, *69*(12), 6088–6096. https://doi.org/https://doi.org/10.1016/j.jbusres.2016.06.016

Barrow, D., & Kourentzes, N. (2018). The impact of special days in call arrivals forecasting: A neural network approach to modelling special days. *European Journal of Operational Research*, *264*(3), 967–977. https://doi.org/10.1016/j.ejor.2016.07.015

Baskaran, T., John, N., & Dhandra, B. V. (2023). *Hybrid Model Using Interacted-ARIMA and ANN Models for Efficient Forecasting BT - Multi-disciplinary Trends in Artificial Intelligence* (R. Morusupalli, T. S. Dandibhotla, V. V. Atluri, D. Windridge, P. Lingras, & V. R. Komati (eds.); pp. 747–756). Springer Nature Switzerland.

Bastianin, A., Galeotti, M., & Manera, M. (2012). Forecast Evaluation in Call Centers: Combined Forecasts, Flexible Loss Functions and Economic Criteria. *SSRN Electronic Journal*, 1–33. https://doi.org/10.2139/ssrn.1829891

Bhulai, S. (2015). Nearest neighbour algorithms for forecasting call arrivals in call centers. *Intelligent Decision Technologies: Proceedings of the 7th KES International Conference on Intelligent Decision Technologies (KES-IDT 2015) 7*, 77–87.

Bhulai, S., Koole, G., & Pot, A. (2008). Simple methods for shift scheduling in multiskill call centers. *Manufacturing & Service Operations Management*, *10*(3), 411–420.

Borovykh, A., & Bohte, S. (2018). *Conditional time series forecasting with convolutional neural networks arXiv : 1703 . 04691v5 [stat . ML] 17 Sep 2018*. 1–22.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

Chacón, H., Koppisetti, V., Hardage, D., Choo, K.-K. R., & Rad, P. (2023a). Forecasting call center arrivals using temporal memory networks and gradient boosting algorithm. *Expert Systems with Applications*, *224*, 119983. https://doi.org/https://doi.org/10.1016/j.eswa.2023.119983

Chacón, H., Koppisetti, V., Hardage, D., Choo, K. K. R., & Rad, P. (2023b). Forecasting call center arrivals using temporal memory networks and gradient boosting algorithm. *Expert Systems with Applications*, *224*(April), 119983. https://doi.org/10.1016/j.eswa.2023.119983

Chakrabarty, N. (2020). A Regression Approach to Distribution and Trend Analysis of Quarterly Foreign Tourist Arrivals in India. *Journal of Soft Computing Paradigm*, *2*(1), 57–82. https://doi.org/10.36548/jscp.2020.1.006

Cirstea, R. G., Micu, D. V., Muresan, G. M., Guo, C., & Yang, B. (2018). Correlated time series forecasting using deep neural networks: A summary of results. *ArXiv*.

Cowen-Rivers, A. I., Lyu, W., Tutunov, R., Wang, Z., Grosnit, A., Rhys, R., Maravel, A. M., Jianye, H., Wang, J., Peters, J., & Bou-Ammar, H. (2022). HEBO: Pushing The Limits of Sample-Efficient Hyperparameter Optimisation. *Journal of Artificial Intelligence Research*, *74*(February), 1269–1349. https://doi.org/10.1613/JAIR.1.13643

Cui, W., Sun, Z., Ma, H., & Wu, S. (2020). The Correlation Analysis of Atmospheric Model Accuracy Based on the Pearson Correlation Criterion. *IOP Conference Series: Materials Science and Engineering*, *780*(3). https://doi.org/10.1088/1757-899X/780/3/032045

Ding, S. (2016). *Workforce Management in Call Centers: Forecasting, Staffing and Empirical studies*. [PhD-Thesis - Research and graduation internal, Vrije Universiteit Amsterdam].

Ebrahimpour, R., Nikoo, H., Masoudnia, S., Yousefi, M. R., & Ghaemi, M. S. (2011). Mixture of mlp-experts for trend forecasting of time series: A case study of the tehran stock exchange. *International Journal of Forecasting*, *27*(3), 804–816. https://doi.org/10.1016/j.ijforecast.2010.02.015

Elshewey, A. M., Shams, M. Y., El-Rashidy, N., Elhady, A. M., Shohieb, S. M., & Tarek, Z. (2023). Bayesian Optimization with Support Vector Machine Model for Parkinson Disease Classification. In *Sensors* (Vol. 23, Issue 4). https://doi.org/10.3390/s23042085

Freeman, B. S., Taylor, G., Gharabaghi, B., & Thé, J. (2018). Forecasting air quality time series using deep learning. *Journal of the Air and Waste Management Association*, *68*(8), 866–886. https://doi.org/10.1080/10962247.2018.1459956

Fu, H., Le, J., Wang, J., Parika, |, Faculty, G., & Benson, H. (n.d.). *Forecasting and Optimizing Call Center Staffing*.

Gans, N., Koole, G., & Mandelbaum, A. (2003). Telephone Call Centers: Tutorial, Review, and Research Prospects. *Manufacturing and Service Operations Management*, *5*(2), 79–141. https://doi.org/10.1287/msom.5.2.79.16071

Garnett, R. (2023). Bayesian Optimization. In *Bayesian Optimization*. Cambridge University Press. https://doi.org/10.1017/9781108348973

Gwetu, M. V., Tapamo, J.-R., & Viriri, S. (2019). *Exploring the Impact of Purity Gap Gain on the Efficiency and Effectiveness of Random Forest Feature Selection BT - Computational Collective Intelligence* (N. T. Nguyen, R. Chbeir, E. Exposito, P. Aniorté, & B. Trawiński (eds.); pp. 340–352). Springer International Publishing.

Habtemariam, E. T., Kekeba, K., Martínez-Ballesteros, M., & Martínez-Álvarez, F. (2023). A Bayesian Optimization-Based LSTM Model for Wind Power Forecasting in the Adama District, Ethiopia. *Energies*, *16*(5), 1–22. https://doi.org/10.3390/en16052317

Hochreiter, S., & Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. *Advances in Neural Information Processing Systems*, 473–479.

Hou, C., Cao, B., & Fan, J. (2022). A data-driven method to predict service level for call centers. *IET Communications*, *16*(10), 1241–1252.

Huo, W., Li, W., Zhang, Z., Sun, C., Zhou, F., & Gong, G. (2021). Performance prediction of proton-exchange membrane fuel cell based on convolutional neural network and random forest feature selection. *Energy Conversion and Management*, *243*, 114367. https://doi.org/https://doi.org/10.1016/j.enconman.2021.114367

Ibrahim, R., Ye, H., L'Ecuyer, P., & Shen, H. (2016a). Modeling and forecasting call center arrivals: A literature survey and a case study. *International Journal of Forecasting*, *32*(3), 865–874. https://doi.org/10.1016/j.ijforecast.2015.11.012

Ibrahim, R., Ye, H., L'Ecuyer, P., & Shen, H. (2016b). Modeling and forecasting call center arrivals: A literature survey and a case study. *International Journal of Forecasting*, *32*(3), 865–874.

Jiang, S. Y., & Wang, L. X. (2016). Efficient feature selection based on correlation measure between continuous and discrete features. *Information Processing Letters*, *116*(2), 203–215. https://doi.org/10.1016/j.ipl.2015.07.005

Kadioglu, M. A., & Alatas, B. (2023). Enhancing Call Center Efficiency: Data Driven Workload Prediction and Workforce Optimization. *Eurasia Proceedings of Science, Technology, Engineering and Mathematics*, *24*, 96–100. https://doi.org/10.55549/epstem.1406245

Kanthanathan, C., Carty, G., Raja, M. A., & Ryan, C. (2020). Recurrent Neural Network based Automated Workload Forecasting in a Contact Center. *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 1423–1428. https://doi.org/10.1109/ICISS49785.2020.9316057

Kervanci, I. S., Akay, M. F., & Özceylan, E. (2024). Bitcoin price prediction using LSTM, GRU and hybrid LSTM-GRU with bayesian optimization, random search, and grid search for the next days. *Journal of Industrial and Management Optimization*, *20*(2), 570–588. https://doi.org/10.3934/jimo.2023091

KERVANCI, I. sibe., & AKAY, F. (2023). LSTM Hyperparameters optimization with Hparam parameters for Bitcoin Price Prediction. *Sakarya University Journal of Computer and Information Sciences*, *6*(1), 1–9. https://doi.org/10.35377/saucis...1172027

Khaldi, R., El Afia, A., Chiheb, R., & Tabik, S. (2023). What is the best RNN-cell structure to forecast each time series behavior? *Expert Systems with Applications*, *215*, 119140.

Khashei, M., & Hajirahimi, Z. (2019). A comparative study of series arima/mlp hybrid models for stock price forecasting. *Communications in Statistics: Simulation and Computation*, *48*(9), 2625–2640. https://doi.org/10.1080/03610918.2018.1458138

Kiwamu, Y., & Goro, H. (2019). *Forecasting call arrivals at call center using dynamic linear model*. *51*, 1–7.

Klusowski, J. M. (2018). Complete Analysis of a Random Forest Model. *ArXiv*, *13*, 1063–1095.

Kumwilaisak, W., Phikulngoen, S., Piriyataravet, J., Thatphithakkul, N., & Hansakunbuntheung, C. (2022). Adaptive Call Center Workforce Management With Deep Neural Network and Reinforcement Learning. *IEEE Access*, *10*, 35712–35724. https://doi.org/10.1109/ACCESS.2022.3160452

Li, C., Ge, P., Liu, Z., & Zheng, W. (2020). Forecasting tourist arrivals using denoising and potential factors. *Annals of Tourism Research*, *83*, 102943.

Li, X., Chen, W., Zhang, Q., & Wu, L. (2020). Building Auto-Encoder Intrusion Detection System based on random forest feature selection. *Computers & Security*, *95*, 101851. https://doi.org/https://doi.org/10.1016/j.cose.2020.101851

Liao, S., Zhou, L., Di, X., Yuan, B., & Xiong, J. (2018). Large-scale short-term urban taxi demand forecasting using deep learning. *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, *2018-Janua*, 428–433.

https://doi.org/10.1109/ASPDAC.2018.8297361

Ma, D., Song, X., & Li, P. (2020). Daily Traffic Flow Forecasting Through a Contextual Convolutional Recurrent Neural Network Modeling Inter-and Intra-Day Traffic Patterns. *IEEE Transactions on Intelligent Transportation Systems*, 1–10. https://doi.org/10.1109/tits.2020.2973279

Manno, A., Rossi, F., Smriglio, S., & Cerone, L. (2023). Comparing deep and shallow neural networks in forecasting call center arrivals. *Soft Computing*, *27*(18), 12943–12957. https://doi.org/10.1007/s00500-022-07055-2

Martin, R. J., Mousavi, R., & Saydam, C. (2021). Predicting emergency medical service call demand: A modern spatiotemporal machine learning approach. *Operations Research for Health Care*, *28*, 100285.

Millán-Ruiz, D., & Hidalgo, J. I. (2013). Forecasting call centre arrivals. *Journal of Forecasting*, *32*(7), 628–638. https://doi.org/10.1002/for.2258

Mitchell, T., Buchanan, B., DeJong, G., Dietterich, T., Rosenbloom, P., & Waibel, A. (1990). Machine learning. *Annual Review of Computer Science*, *4*(1), 417–433.

Niu, D., Wang, K., Sun, L., Wu, J., & Xu, X. (2020). Short-term photovoltaic power generation forecasting based on random forest feature selection and CEEMD: A case study. *Applied Soft Computing*, *93*, 106389. https://doi.org/https://doi.org/10.1016/j.asoc.2020.106389

Paouris, E., Vourlidas, A., Papaioannou, A., & Anastasiadis, A. (2021). Assessing the Projection Correction of Coronal Mass Ejection Speeds on Time-of-Arrival Prediction Performance Using the Effective Acceleration Model. *Space Weather*, *19*(2), e2020SW002617.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. https://doi.org/10.1109/TPAMI.2005.159

Qin, Z., Wang, P., Zhang, G., & Zhang, H. (2017). A novel hybrid-Garch model based on ARIMA and SVM for PM2.5 concentrations forecasting. *Atmospheric Pollution Research*, *8*(5), 850–860. https://doi.org/http://dx.doi.org/10.1016/j.apr.2017.01.003

Rausch, T. M., Albrecht, T., & Baier, D. (2022). Beyond the beaten paths of forecasting call center arrivals: on the use of dynamic harmonic regression with predictor variables. In *Journal of Business Economics* (Vol. 92, Issue 4). Springer Berlin Heidelberg. https://doi.org/10.1007/s11573-021-01075-4

Rehman, A. U., Malik, A. K., Raza, B., & Ali, W. (2019). A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. *Multimedia Tools and Applications*, *78*(18), 26597–26613. https://doi.org/10.1007/s11042-019-07788-7

Samal, U., & Kumar, A. (2023). Enhancing Software Reliability Forecasting Through a Hybrid ARIMA-ANN Model. *Arabian Journal for Science and Engineering*.

https://doi.org/10.1007/s13369-023-08486-1

Scheidt, S., & Chung, Q. B. (2019). Making a case for speech analytics to improve customer service quality: Vision, implementation, and evaluation. *International Journal of Information Management*, *45*(November 2017), 223–232. https://doi.org/10.1016/j.ijinfomgt.2018.01.002

Shen, H., & Huang, J. Z. (2008a). Forecasting time series of inhomogeneous Poisson processes with application to call center workforce management. *Annals of Applied Statistics*, *2*(2), 601–623. https://doi.org/10.1214/08-AOAS164

Shen, H., & Huang, J. Z. (2008b). Interday forecasting and intraday updating of call center arrivals. *Manufacturing and Service Operations Management*, *10*(3), 391–410. https://doi.org/10.1287/msom.1070.0179

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, *25*.

Sprock, T., & McGinnis, L. F. (2014). Proceedings of the 2014 Winter Simulation Conference A. Tolk, S. *Winter Simulation Conference*, *1983*, 2600–2608.

Surasai, P., & Sa-ing, V. (2023). *TIME SERIES FORECAST OF CALL ARRIVALS USING MACHINE LEARNING METHODS*. 273–292.

Tartuk, M., Taha NURDAĞ, F., Acar, V., Erdem, S., Fatih AKAY, M., & Abut, F. (2022). Forecasting Call Center Arrivals Using XGBoost Combined with Consecutive and Periodic Lookback. In *Eastern Anatolian Journal of Science: Vol. VIII* (Issue 1). Ağrı İbrahim Çeçen University WT - DergiPark.

Taylor, J. W. (2012). Density forecasting of intraday call center arrivals using models based on exponential smoothing. *Management Science*, *58*(3), 534–539. https://doi.org/10.1287/mnsc.1110.1434

Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., & Troncoso, A. (2021). Deep Learning for Time Series Forecasting: A Survey. *Big Data*, *9*(1), 3–21. https://doi.org/10.1089/big.2020.0159

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. (2020). Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. *Proceedings of Machine Learning Research*, *133*, 3–26.

Victoria, A. H., & Maragatham, G. (2021). Automatic tuning of hyperparameters using Bayesian optimization. *Evolving Systems*, *12*(1), 217–223. https://doi.org/10.1007/s12530-020-09345-2

Weinberg, J., Brown, L. D., & Stroud, J. R. (2007). Bayesian forecasting of an inhomogeneous poisson process with applications to call center data. *Journal of the American Statistical Association*, *102*(480), 1185–1198. https://doi.org/10.1198/016214506000001455

Xu, W., Peng, H., Zeng, X., Zhou, F., Tian, X., & Peng, X. (2019). A hybrid modelling method for time series forecasting based on a linear regression model and deep learning. *Applied*

Intelligence, *49*(8), 3002–3015. https://doi.org/10.1007/s10489-019-01426-3

Yasin, M., Sarıgül, M., & Avci, M. (2024). Logarithmic learning differential convolutional neural network. *Neural Networks*, 106114.

Yu, M., Kollias, D., Wingate, J., Siriwardena, N., & Kollias, S. (2021). Machine learning for predictive modelling of ambulance calls. *Electronics (Switzerland)*, *10*(4), 1–12. https://doi.org/10.3390/electronics10040482

Zeroual, A., Harrou, F., Dairi, A., & Sun, Y. (2020). Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos, Solitons and Fractals*, *140*, 110121. https://doi.org/10.1016/j.chaos.2020.110121

Zhang, F., & O'Donnell, L. J. (2019). Support vector regression. In *Machine Learning: Methods and Applications to Brain Disorders*. Elsevier Inc. https://doi.org/10.1016/B978-0-12-815739-8.00007-9

CURRICULUM VITAE

Rukia Kasauli NAKKAZI earned her undergraduate degree in Information Technology from the Department of Computer Science at the Islamic University in Uganda (IUIU). She then pursued a master's degree in Information Technology (IT) at the International Islamic University, Malaysia (IIUM). Rukia completed her doctoral studies in Computer Engineering at Cukurova University in Adana, Turkey. Throughout her PhD journey, she actively engaged in numerous conferences centered on machine learning and neural networks. In addition to her academic pursuits, she also served as an IT lecturer at the Islamic University in Uganda (IUIU), Kampala campus.