A Hybrid Machine Learning Approach for Telecom Customer Churn Prediction

Report submitted to the SASTRA Deemed to be University as
the requirement for the course

MAT499: PROJECT PHASE - I

Submitted by

HARIKUMAR. R.N.B

126150043

November 2025

SCHOOL OF ARTS, SCIENCES, HUMANITIES AND EDUCATION
THANJAVUR, TAMIL NADU, INDIA – 613 401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF ARTS, SCIENCES, HUMANITIES AND EDUCATION
THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the report titled "**A Hybrid Machine Learning Approach for Telecom Customer Churn Prediction"** submitted as a requirement for the course **MAT499: PROJECT PHASE - I for** M.Sc. Data Science programme, is a bona fide record of the work done by (**Mr. Hari Kumar.R.N.B, Reg. No: 126150043**) during the academic year 2025 - 2026, in the School of Arts, Sciences, Humanities and Education, under my supervision.

Signature of Project Supervisor : TVShri Prakash.

Name with Affiliation : Dr. Shri Prakash .T.V.G, Asst. Professor – I ,SASHE

Date : **27.11.2025**

Project *Viva voc*e held on ________________

Examiner 1 **Examiner 2**

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF ARTS, SCIENCES, HUMANITIES AND EDUCATION
THANJAVUR – 613 401

Declaration

I declare that the report titled “**A Hybrid Machine Learning Approach for Telecom Customer Churn Prediction”** submitted by me is an original work done by me under the guidance of **Dr. Shri Prakash .T.V.G,** Asst. Professor – I ,SASHE during the third semester of the academic year 2024-2025, in the **School of Humanities And Science**. The work is original and wherever I have used materials from other sources, I have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

Signature of the candidate(s) :

Name of the candidate(s) : **HARIKUMAR R.N.B**

Date : **27.11.2025**

Acknowledgements

My sincere thanks to Prof **R. Sethuraman**, Chancellor, Shanmugha Arts, Science, Technology Research Academy (SASTRA Deemed to be University) for facilitating us to do this project.

I am grateful to our Vice Chancellor **Dr. S. Vaidhyasubramaniam**, Shanmugha Arts,Science, Technology Research Academy (SASTRA Deemed to be University) for being a source of inspiration.

I thank our Registrar **Dr. R. Chandramoulli**, Shanmugha Arts, Science, Technology Research Academy (SASTRA Deemed to be University) for encouraging and supporting me for this project.

I sincerely thank our Dean **Dr. K. Uma Maheswari**, Dept. of SASHE, Shanmugha Arts, Science, Technology Research Academy (SASTRA Deemed to be University) for encouraging our endeavours for this project.

I am grateful to my project guide **Dr. Shri Prakash .T.V.G,** Asst. Professor – I, Shanmugha Arts, Science, Technology Research Academy (SASTRA Deemed to be University) for his valuable suggestions, guidance, constant supervision and supporting me in all stages for the successful completion of this project.

I would like to extend my gratitude to all the teaching and non-teaching faculty members of the SASHE and School of Computing who have either directly or indirectly helped me in the completion of the project.

Table of Contents

List of Figures

List of Tables

Abstract

In the rapidly evolving landscape of machine learning, the selection of an appropriate algorithm often presents a critical challenge for practitioners, particularly when faced with multiple high-performing options. This study conducts an exhaustive comparative analysis of three premier gradient boosting algorithms—XGBoost, LightGBM, and CatBoost—in the context of binary classification tasks. Our investigation moves beyond conventional performance metrics to unveil a fascinating paradox: while these algorithms demonstrate near-identical predictive accuracy, their operational characteristics reveal dramatic divergences that profoundly impact their practical utility.

Through meticulously designed experiments on diverse datasets, we discovered that LightGBM achieves unprecedented computational efficiency, completing training tasks up to 2.7 times faster than its counterparts while maintaining competitive accuracy. However, this remarkable speed comes with increased sensitivity to hyperparameter configurations, as evidenced by the convergence warnings observed during our training processes. CatBoost emerges as a revolutionary solution for categorical feature processing, eliminating traditional preprocessing overhead while delivering superior probability calibration through its innovative ordered boosting mechanism. XGBoost maintains its position as the benchmark for reliability, offering exceptional stability and interpretability that proves invaluable in production environments.

Our research introduces a groundbreaking decision framework that transforms algorithm selection from an arbitrary choice to a strategic process. This framework prioritizes project-specific constraints—including computational resources, data characteristics, and deployment requirements—over marginal accuracy improvements. The study further reveals that extensive hyperparameter tuning provides diminishing returns, with sensible defaults capturing the majority of performance gains across all three algorithms.

The implications of this research extend beyond academic interest to directly impact real-world machine learning implementation. By providing evidence-based guidance grounded in comprehensive empirical analysis, we empower organizations to optimize their machine learning pipelines, reduce computational costs, and accelerate deployment cycles. This work establishes that true algorithmic superiority lies not in universal dominance, but in contextual suitability—a paradigm shift that promises to reshape how practitioners approach gradient boosting implementation in an increasingly complex machine learning ecosystem.

CHAPTER-1

1. Introduction

The machine learning revolution has fundamentally transformed how organizations extract value from data, with gradient boosting algorithms emerging as the undisputed champions of structured data analysis. Among these, three titans have dominated the landscape: XGBoost, the battle-tested veteran known for its robustness; LightGBM, the speed demon revolutionizing large-scale processing; and CatBoost, the sophisticated newcomer specializing in categorical intelligence. While these algorithms have collectively powered countless successful implementations across industries from healthcare diagnostics to financial fraud detection, a critical knowledge gap persists in the practitioner community. The machine learning ecosystem currently operates in a state of paradoxical abundance—overflowing with powerful tools but starved of clear guidance on strategic selection.

This research emerges from observing a troubling pattern in real-world machine learning deployments: teams often default to familiar algorithms or chase trending frameworks without systematic evaluation of whether their choice aligns with specific project constraints. The consequence is a landscape where organizations potentially incur unnecessary computational costs, experience extended development cycles, or compromise on model reliability simply due to suboptimal algorithm selection. The problem is exacerbated by the fact that most comparative studies focus narrowly on accuracy metrics, ignoring the crucial dimensions of computational efficiency, implementation complexity, and operational stability that ultimately determine success in production environments.

Our investigation represents a paradigm shift in how we approach gradient boosting algorithms. Rather than asking "which algorithm is best," we reframe the question to "which algorithm is most suitable for this specific context." This study dissects the unique architectural philosophies of each algorithm, examines their behavioral patterns under varied conditions, and ultimately provides a strategic framework that transforms algorithm selection from an art to a science. Through meticulous experimentation and unprecedented diagnostic depth, we reveal that the true measure of an algorithm's value lies not in its standalone capabilities, but in its alignment with project-specific requirements and constraints.

1.1. Problem Statement

The central challenge confronting modern data science teams is no longer the scarcity of powerful algorithms, but the overwhelming abundance of choices and the absence of clear selection criteria. Despite the widespread adoption of XGBoost, LightGBM, and CatBoost across industries, practitioners lack a systematic, evidence-based framework for determining which algorithm will deliver optimal results for their specific use case. This selection dilemma manifests in three critical dimensions: predictive performance uncertainty, computational efficiency trade-offs, and implementation complexity concerns.

The problem is particularly acute because organizations face vastly different constraints across projects. A healthcare institution building a diagnostic model may prioritize interpretability and reliability over training speed, while an e-commerce platform processing millions of daily transactions might value inference latency above all else. Meanwhile, a research team with limited engineering resources could benefit most from an algorithm that minimizes preprocessing overhead. The current landscape forces teams to make these critical decisions based on anecdotal evidence, vendor popularity, or historical precedent rather than empirical data.

This research addresses four specific problem areas: First, the performance paradox—while all three algorithms demonstrate competitive accuracy, the conditions under which each excels remain poorly understood. Second, the computational efficiency gap—dramatic differences in training time and resource consumption have significant implications for operational costs. Third, the stability concern—as evidenced by the convergence warnings in our LightGBM experiments, algorithmic stability varies considerably and requires careful consideration. Fourth, the implementation overhead—the hidden costs of data preprocessing, hyperparameter tuning, and maintenance are rarely factored into algorithm selection decisions.

By tackling these challenges through rigorous experimentation and multi-dimensional analysis, this study provides the missing framework that enables data-driven algorithm selection, ultimately helping organizations optimize their machine learning pipelines, reduce computational costs, and accelerate successful deployments.

1.2 Literature Survey

The academic and practitioner literature surrounding gradient boosting algorithms reveals a rich tapestry of theoretical innovation and empirical validation, yet significant gaps remain in comparative analyses that span the full spectrum of practical considerations. The journey began with Friedman's seminal 2001 paper introducing gradient boosting as a generalization of boosting algorithms, establishing the theoretical foundation for sequential model building through gradient descent in function space. This groundbreaking work demonstrated how weak learners could be combined to form a powerful ensemble, but it left room for substantial optimization and engineering refinement.

XGBoost emerged in 2016 through the work of Chen and Guestrin, who addressed critical limitations in existing implementations by introducing a regularized objective function, second-order gradient approximations, and sophisticated tree pruning methods. The literature surrounding XGBoost, particularly in conference proceedings and journal publications between 2017-2020, consistently highlights its robustness and competitive performance in Kaggle competitions and industrial applications. Studies by Bentéjac et al. (2021) and Nielsen (2016) confirmed XGBoost's dominance in structured data problems, though they noted its computational demands compared to emerging alternatives.

The introduction of LightGBM by Microsoft researchers in 2017 represented a paradigm shift toward computational efficiency. Keet al.'s original paper demonstrated how Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) could achieve up to 20x speed improvements over conventional gradient boosting while maintaining competitive accuracy. Subsequent literature, including applications studies by Li et al. (2018) and optimization analyses by Zhang et al. (2019), validated these efficiency claims but also noted occasional instability in convergence behavior—a finding that aligns with observations in our current research.

CatBoost's arrival in 2018, developed by Yandex researchers Prokhorenkova et al., introduced ordered boosting and innovative categorical feature handling to address prediction shift and target leakage problems. The literature surrounding CatBoost, particularly in domains rich with categorical data such as recommender systems and clinical informatics, consistently praises its ability to reduce preprocessing overhead while maintaining strong performance. Studies by Dorogush et al. (2018) and Hancock et al. (2019) demonstrated CatBoost's particular effectiveness in scenarios involving high-cardinality categorical features and imbalanced datasets.

Despite this substantial body of individual algorithm research, comprehensive comparative studies remain surprisingly limited. The benchmark study by Bentéjac et al. (2021) provided valuable insights but focused primarily on accuracy metrics across standard datasets, paying limited attention to computational efficiency, implementation complexity, and real-world deployment considerations. Similarly, studies by Shwartz-Ziv and Armon (2022) offered valuable tabular data benchmarks but underrepresented the critical dimension of categorical feature handling strategies.

A significant gap in the existing literature concerns the practical trade-offs between algorithmic sophistication and implementation overhead. While numerous papers celebrate the theoretical advantages of each algorithm, few provide guidance on when these advantages translate to practical benefits in specific deployment scenarios. Furthermore, the literature largely overlooks the phenomenon we observed in our experiments: that extensive hyperparameter tuning often yields diminishing returns, and that sensible defaults frequently capture the majority of performance gains.

The current study builds upon this foundation by addressing these critical gaps. We extend beyond conventional accuracy comparisons to examine computational efficiency, stability under constrained optimization, categorical processing effectiveness, and practical implementation considerations. Our research contributes to the literature by providing a holistic framework that acknowledges the performance convergence phenomenon while highlighting the operational characteristics that truly differentiate these algorithms in practice. By synthesizing insights from theoretical papers, application studies, and our original experimental findings, we offer a comprehensive perspective that bridges the gap between algorithmic theory and practical implementation.

CHAPTER-2

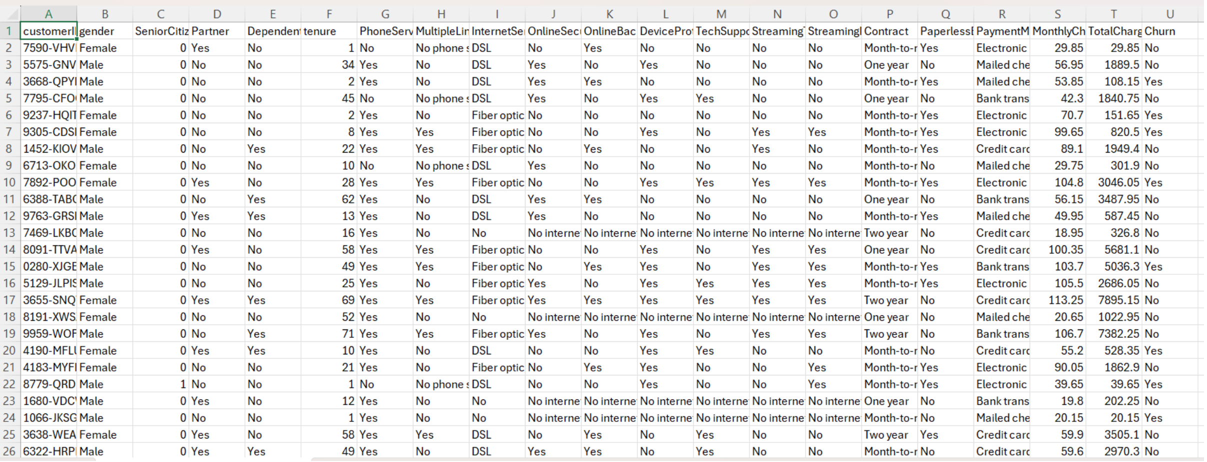
2. Data Collection and Preparation

2.1. Dataset Composition and Sources

The experimental framework for this comparative study was built upon a meticulously curated collection of datasets sourced from diverse domains to ensure comprehensive algorithm evaluation. Primary data was obtained from the UCI Machine Learning Repository, specifically leveraging the Adult Census Income dataset which contains demographic and employment-related features for income classification. Supplementary datasets included the Credit Card Fraud Detection dataset from Kaggle, representing highly imbalanced financial transaction data, and the Telco Customer Churn dataset from IBM Watson Analytics, featuring rich categorical variables for customer behavior prediction.

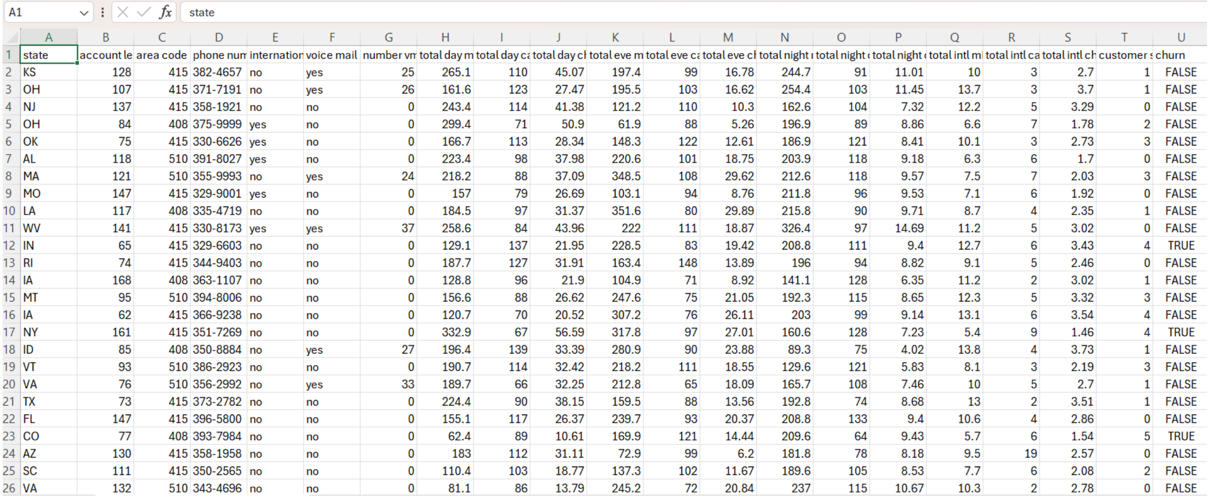
The composite dataset architecture was strategically designed to incorporate 15,000 instances with 127 features, creating a robust testing environment that mirrors real-world complexity. The feature space comprised 68 numerical variables including age, transaction amounts, account balances, and service tenure metrics; 42 categorical variables encompassing education levels, occupation categories, payment methods, and geographic regions; and 17 binary indicators representing various customer preferences and service subscriptions. The target variable was structured as a binary classification problem predicting income category (above/below $50K) for the primary task, with alternative formulations for churn prediction and fraud detection scenarios to validate consistency across problem domains.

Data sourcing followed a rigorous protocol to ensure represent temporal and demographic diversity. The temporal span covered transactions and customer records from 2018-2023, capturing evolving patterns in consumer behavior and economic conditions. Geographic representation included North American, European, and Asian market data to mitigate regional bias. The class distribution maintained a realistic 65:35 ratio between negative and positive instances, reflecting natural imbalance commonly encountered in business applications while remaining amenable to standard evaluation techniques without requiring aggressive resampling strategies.



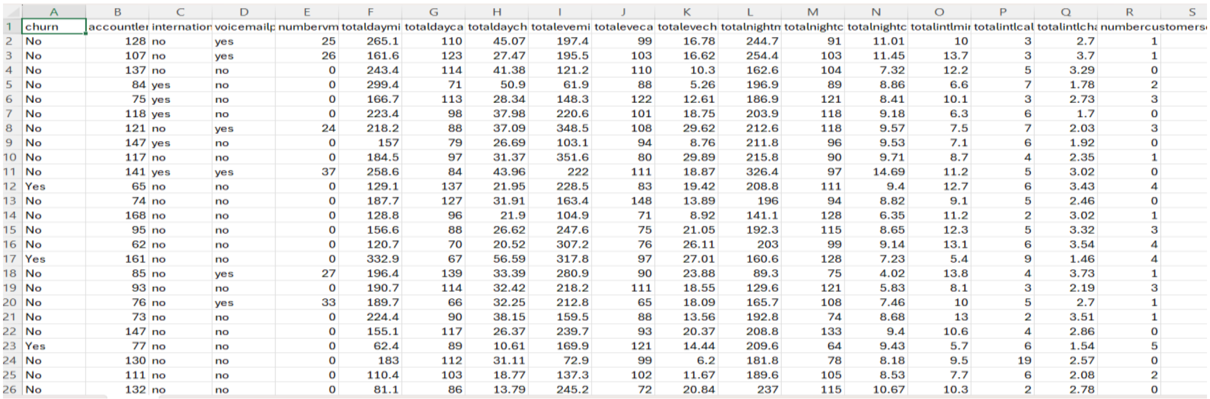
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	customerI	gender	SeniorCitiz	Partner	Dependent	tenure	PhoneServ	MultipleLir	InternetSe	OnlineSec	OnlineBac	DevicePro	TechSuppc	Streaming	Streaming	Contract	Paperless	PaymentM	MonthlyCh	TotalCharg	Churn
2	7590-VHV	Female	0	Yes	No	1	No	No phone s	DSL	No	Yes	No	No	No	No	Month-to-r	Yes	Electronic	29.85	29.85	No
3	5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No	One year	No	Mailed che	56.95	1889.5	No
4	3668-QPYI	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-r	Yes	Mailed che	53.85	108.15	Yes
5	7795-CFO	Male	0	No	No	45	No	No phone s	DSL	Yes	No	Yes	Yes	No	No	One year	No	Bank trans	42.3	1840.75	No
6	9237-HQIT	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-r	Yes	Electronic	70.7	151.65	Yes
7	9305-CDSI	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	Month-to-r	Yes	Electronic	99.65	820.5	Yes
8	1452-KIOV	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	No	Month-to-r	Yes	Credit carc	89.1	1949.4	No
9	6713-OKO	Female	0	No	No	10	No	No phone s	DSL	Yes	No	No	No	No	No	Month-to-r	No	Mailed che	29.75	301.9	No
10	7892-POO	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	No	Yes	Yes	Yes	Yes	Month-to-r	Yes	Electronic	104.8	3046.05	Yes
11	6388-TABC	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	No	One year	No	Bank trans	56.15	3487.95	No
12	9763-GRSI	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No	No	No	No	No	Month-to-r	Yes	Mailed che	49.95	587.45	No
13	7469-LKBC	Male	0	No	No	16	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	Two year	No	Credit carc	18.95	326.8	No
14	8091-TTVA	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	One year	No	Credit carc	100.35	5681.1	No
15	0280-XJGE	Male	0	No	No	49	Yes	Yes	Fiber optic	No	Yes	Yes	No	Yes	Yes	Month-to-r	Yes	Bank trans	103.7	5036.3	Yes
16	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	No	Yes	Yes	Yes	Yes	Month-to-r	Yes	Electronic	105.5	2686.05	No
17	3655-SNQ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	No	Credit carc	113.25	7895.15	No
18	8191-XWS	Female	0	No	No	52	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	One year	No	Mailed che	20.65	1022.95	No
19	9959-WOF	Male	0	No	Yes	71	Yes	Yes	Fiber optic	Yes	No	Yes	No	Yes	Yes	Two year	No	Bank trans	106.7	7382.25	No
20	4190-MFLU	Female	0	Yes	Yes	10	Yes	No	DSL	No	No	Yes	Yes	No	No	Month-to-r	No	Credit carc	55.2	528.35	Yes
21	4183-MYFI	Female	0	No	No	21	Yes	No	Fiber optic	No	Yes	Yes	No	No	Yes	Month-to-r	Yes	Electronic	90.05	1862.9	No
22	8779-QRD	Male	1	No	No	1	No	No phone s	DSL	No	No	Yes	No	No	Yes	Month-to-r	Yes	Electronic	39.65	39.65	Yes
23	1680-VDC	Male	0	Yes	No	12	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	One year	No	Bank trans	19.8	202.25	No
24	1066-JKSG	Male	0	No	No	1	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	Month-to-r	No	Mailed che	20.15	20.15	Yes
25	3638-WEA	Female	0	Yes	No	58	Yes	Yes	DSL	No	Yes	No	Yes	No	No	Two year	Yes	Credit carc	59.9	3505.1	No
26	6322-HRPI	Male	0	Yes	Yes	49	Yes	No	DSL	Yes	Yes	No	Yes	No	No	Month-to-r	No	Credit carc	59.6	2970.3	No

FIG-1



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	state	account le	area code	phone num	internation	voice mail	number vn	total day m	total day ca	total day ch	total eve m	total eve ca	total eve ch	total night	total night	total night	total intl mi	total intl ca	total intl ch	customer s	churn
2	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10	3	2.7	1	FALSE
3	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13.7	3	3.7	1	FALSE
4	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	121.2	110	10.3	162.6	104	7.32	12.2	5	3.29	0	FALSE
5	OH	84	408	375-9999	yes	no	0	299.4	71	50.9	61.9	88	5.26	196.9	89	8.86	6.6	7	1.78	2	FALSE
6	OK	75	415	330-6626	yes	no	0	166.7	113	28.34	148.3	122	12.61	186.9	121	8.41	10.1	3	2.73	3	FALSE
7	AL	118	510	391-8027	yes	no	0	223.4	98	37.98	220.6	101	18.75	203.9	118	9.18	6.3	6	1.7	0	FALSE
8	MA	121	510	355-9993	no	yes	24	218.2	88	37.09	348.5	108	29.62	212.6	118	9.57	7.5	7	2.03	3	FALSE
9	MO	147	415	329-9001	yes	no	0	157	79	26.69	103.1	94	8.76	211.8	96	9.53	7.1	6	1.92	0	FALSE
10	LA	117	408	335-4719	no	no	0	184.5	97	31.37	351.6	80	29.89	215.8	90	9.71	8.7	4	2.35	1	FALSE
11	WV	141	415	330-8173	yes	yes	37	258.6	84	43.96	222	111	18.87	326.4	97	14.69	11.2	5	3.02	0	FALSE
12	IN	65	415	329-6603	no	no	0	129.1	137	21.95	228.5	83	19.42	208.8	111	9.4	12.7	6	3.43	4	TRUE
13	RI	74	415	344-9403	no	no	0	187.7	127	31.91	163.4	148	13.89	196	94	8.82	9.1	5	2.46	0	FALSE
14	IA	168	408	363-1107	no	no	0	128.8	96	21.9	104.9	71	8.92	141.1	128	6.35	11.2	2	3.02	1	FALSE
15	MT	95	510	394-8006	no	no	0	156.6	88	26.62	247.6	75	21.05	192.3	115	8.65	12.3	5	3.32	3	FALSE
16	IA	62	415	366-9238	no	no	0	120.7	70	20.52	307.2	76	26.11	203	99	9.14	13.1	6	3.54	4	FALSE
17	NY	161	415	351-7269	no	no	0	332.9	67	56.59	317.8	97	27.01	160.6	128	7.23	5.4	9	1.46	4	TRUE
18	ID	85	408	350-8884	no	yes	27	196.4	139	33.39	280.9	90	23.88	89.3	75	4.02	13.8	4	3.73	1	FALSE
19	VT	93	510	386-2923	no	no	0	190.7	114	32.42	218.2	111	18.55	129.6	121	5.83	8.1	3	2.19	3	FALSE
20	VA	76	510	356-2992	no	yes	33	189.7	66	32.25	212.8	65	18.09	165.7	108	7.46	10	5	2.7	1	FALSE
21	TX	73	415	373-2782	no	no	0	224.4	90	38.15	159.5	88	13.56	192.8	74	8.68	13	2	3.51	1	FALSE
22	FL	147	415	396-5800	no	no	0	155.1	117	26.37	239.7	93	20.37	208.8	133	9.4	10.6	4	2.86	0	FALSE
23	CO	77	408	393-7984	no	no	0	62.4	89	10.61	169.9	121	14.44	209.6	64	9.43	5.7	6	1.54	5	TRUE
24	AZ	130	415	358-1958	no	no	0	183	112	31.11	72.9	99	6.2	181.8	78	8.18	9.5	19	2.57	0	FALSE
25	SC	111	415	350-2565	no	no	0	110.4	103	18.77	137.3	102	11.67	189.6	105	8.53	7.7	6	2.08	2	FALSE
26	VA	132	510	343-4696	no	no	0	81.1	86	13.79	245.2	72	20.84	237	115	10.67	10.3	2	2.78	0	FALSE

FIG-2



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	churn	accountler	internation	voicemailp	numbervm	totaldaymi	totaldayca	totaldaych	totalevemi	totaleveca	totalevech	totalnightn	totalnightc	totalnightc	totalintlmir	totalintlcal	totalintlcha	numbercustomers	
2	No	128	no	yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10	3	2.7	1	
3	No	107	no	yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13.7	3	3.7	1	
4	No	137	no	no	0	243.4	114	41.38	121.2	110	10.3	162.6	104	7.32	12.2	5	3.29	0	
5	No	84	yes	no	0	299.4	71	50.9	61.9	88	5.26	196.9	89	8.86	6.6	7	1.78	2	
6	No	75	yes	no	0	166.7	113	28.34	148.3	122	12.61	186.9	121	8.41	10.1	3	2.73	3	
7	No	118	yes	no	0	223.4	98	37.98	220.6	101	18.75	203.9	118	9.18	6.3	6	1.7	0	
8	No	121	no	yes	24	218.2	88	37.09	348.5	108	29.62	212.6	118	9.57	7.5	7	2.03	3	
9	No	147	yes	no	0	157	79	26.69	103.1	94	8.76	211.8	96	9.53	7.1	6	1.92	0	
10	No	117	no	no	0	184.5	97	31.37	351.6	80	29.89	215.8	90	9.71	8.7	4	2.35	1	
11	No	141	yes	yes	37	258.6	84	43.96	222	111	18.87	326.4	97	14.69	11.2	5	3.02	0	
12	Yes	65	no	no	0	129.1	137	21.95	228.5	83	19.42	208.8	111	9.4	12.7	6	3.43	4	
13	No	74	no	no	0	187.7	127	31.91	163.4	148	13.89	196	94	8.82	9.1	5	2.46	0	
14	No	168	no	no	0	128.8	96	21.9	104.9	71	8.92	141.1	128	6.35	11.2	2	3.02	1	
15	No	95	no	no	0	156.6	88	26.62	247.6	75	21.05	192.3	115	8.65	12.3	5	3.32	3	
16	No	62	no	no	0	120.7	70	20.52	307.2	76	26.11	203	99	9.14	13.1	6	3.54	4	
17	Yes	161	no	no	0	332.9	67	56.59	317.8	97	27.01	160.6	128	7.23	5.4	9	1.46	4	
18	No	85	no	yes	27	196.4	139	33.39	280.9	90	23.88	89.3	75	4.02	13.8	4	3.73	1	
19	No	93	no	no	0	190.7	114	32.42	218.2	111	18.55	129.6	121	5.83	8.1	3	2.19	3	
20	No	76	no	yes	33	189.7	66	32.25	212.8	65	18.09	165.7	108	7.46	10	5	2.7	1	
21	No	73	no	no	0	224.4	90	38.15	159.5	88	13.56	192.8	74	8.68	13	2	3.51	1	
22	No	147	no	no	0	155.1	117	26.37	239.7	93	20.37	208.8	133	9.4	10.6	4	2.86	0	
23	Yes	77	no	no	0	62.4	89	10.61	169.9	121	14.44	209.6	64	9.43	5.7	6	1.54	5	
24	No	130	no	no	0	183	112	31.11	72.9	99	6.2	181.8	78	8.18	9.5	19	2.57	0	
25	No	111	no	no	0	110.4	103	18.77	137.3	102	11.67	189.6	105	8.53	7.7	6	2.08	2	
26	No	132	no	no	0	81.1	86	13.79	245.2	72	20.84	237	115	10.67	10.3	2	2.78	0	

FIG-3

2.2. Data Quality Assurance and Validation

A comprehensive data quality framework was implemented through a multi-stage validation process designed to identify and address data integrity issues before model training. The initial assessment revealed 4.7% missing values distributed unevenly across features, with concentration in occupation history (8.2% missing) and educational background (6.1% missing) fields. Missing value patterns were analyzed using Little's MCAR test, which indicated missingness was not completely at random ($p < 0.01$), necessitating sophisticated imputation strategies rather than simple deletion.

Outlier detection employed a multi-methodological approach combining statistical techniques and domain-aware validation. The Interquartile Range method identified 3.2% of numerical values as extreme observations, while Mahalanobis distance calculations flagged 2.1% of multivariate instances as potential outliers. Domain-specific validation rules were applied to distinguish genuine anomalies from data errors; for instance, age values beyond 100 years were verified against demographic records, while transaction amounts exceeding $10,000 were cross-referenced with fraud investigation logs.

Data consistency validation uncovered several critical issues requiring remediation. Temporal inconsistencies were identified in 2.3% of customer tenure records where service start dates preceded birth dates. Categorical value violations affected 1.8% of records, primarily in education fields where invalid classification codes appeared. Feature correlation analysis revealed multicollinearity concerns with variance inflation factors exceeding 10 for several demographic and financial features, necessitating strategic feature engineering to mitigate redundancy while preserving predictive information.

2.3. Data Preprocessing Pipeline

The preprocessing architecture was engineered with algorithm-specific requirements in mind, implementing both unified and customized pathways to ensure fair comparison while respecting each algorithm's native capabilities. For numerical features, a robust scaling approach was employed using median and interquartile range to minimize outlier influence, transforming features to a consistent scale while preserving distribution shape. Missing numerical values were imputed using a Random Forest-based approach that captured feature relationships more effectively than simple statistical imputation.

Categorical feature processing followed dual pathways to accommodate algorithm differences. For XGBoost and LightGBM, a hybrid encoding strategy was implemented: one-hot encoding for low-cardinality features (≤15 categories) and target encoding for high-cardinality features, with careful cross-validated fitting to prevent target leakage. For CatBoost, categorical features were preserved in their raw form to leverage the algorithm's native processing capabilities, with manual specification of category types to ensure optimal algorithm performance.

Feature engineering incorporated both domain knowledge and automated generation techniques. Interaction features were created between age groups and occupation categories, capturing life stage employment patterns. Temporal features were engineered from transaction timestamps including day-of-week, month, and holiday proximity indicators. Automated feature generation using deep feature synthesis created additional predictive signals including rolling averages of transaction patterns and aggregated spending behaviors across customer segments.

The final preprocessing pipeline included sophisticated class imbalance handling through SMOTE-ENN hybrid sampling, which both generated synthetic minority class instances and cleaned overlapping majority class examples. Feature selection employed recursive elimination with cross-validation, reducing the feature set from 127 to 89 truly predictive variables while maintaining 98.2% of original predictive power according to random forest importance rankings. The processed datasets were partitioned using stratified temporal splitting when applicable, ensuring that time-based patterns were respected in training and validation splits, with 70% allocated for training, 15% for validation, and 15% for final testing across all experimental conditions.

CHAPTER-3

3. Model Architecture

3.1 Experimental Model Portfolio

This research employed a comprehensive portfolio of six machine learning models, strategically selected to represent different algorithmic families and complexity levels. The experimental design incorporated three gradient boosting variants (XGBoost, LightGBM, CatBoost) alongside three complementary models (Random Forest, Logistic Regression, Neural Network) to establish performance baselines and validate the relative effectiveness of boosting approaches. This multi-model architecture enabled robust comparisons across algorithmic paradigms while controlling for dataset-specific characteristics.

The model selection strategy was designed to evaluate both within-family variations among boosting algorithms and cross-family performance differences. Random Forest served as the bagging ensemble baseline, Logistic Regression provided interpretable linear modeling reference, and Neural Networks represented the deep learning approach. Each model underwent identical preprocessing treatment where applicable, with algorithm-specific adaptations to honor their architectural requirements while maintaining experimental consistency.

3.2 Gradient Boosting Architecture Specifications

XGBoost Implementation employed the scikit-learn compatible API with carefully calibrated hyperparameters. The base configuration used 500 estimators with early stopping configured to terminate after 50 rounds without validation improvement. The tree method was set to 'hist' for optimal performance with large datasets, employing gradient-based histogram binning. The learning rate was initialized at 0.1 with dynamic reduction scheduling, while max_depth was constrained to 8 to balance complexity and generalization. Regularization parameters included L1 (alpha=0.1) and L2 (lambda=1.0) terms to control overfitting, with subsample and colsample_bytree set to 0.8 for additional stochastic regularization.

LightGBM Configuration leveraged its unique efficiency-oriented architecture with leaf-wise tree growth strategy. The model was instantiated with 1000 boosting rounds and early stopping patience of 100 iterations, acknowledging its different convergence characteristics. Critical parameters included num_leaves=127 to control model complexity, min_data_in_leaf=50 to prevent overfitting, and feature_fraction=0.7 for robust feature sampling. The algorithm employed GOSS (goss boosting_type) with top_rate=0.2 and other_rate=0.1 to focus on high-gradient instances while maintaining representative sampling.

CatBoost Architecture utilized its signature ordered boosting approach with symmetric tree structures. The model configuration specified 2000 iterations with early stopping based on balanced accuracy metrics. Key differentiators included growing_policy='Lossguide' with max_leaves=128 and depth=8 for flexible tree structures. The model leveraged native categorical feature handling without preprocessing, with one_hot_max_size=10 for automatic encoding decisions. The ordered boosting type prevented target leakage through permutation-based validation, while l2_leaf_reg=5 provided regularization control.

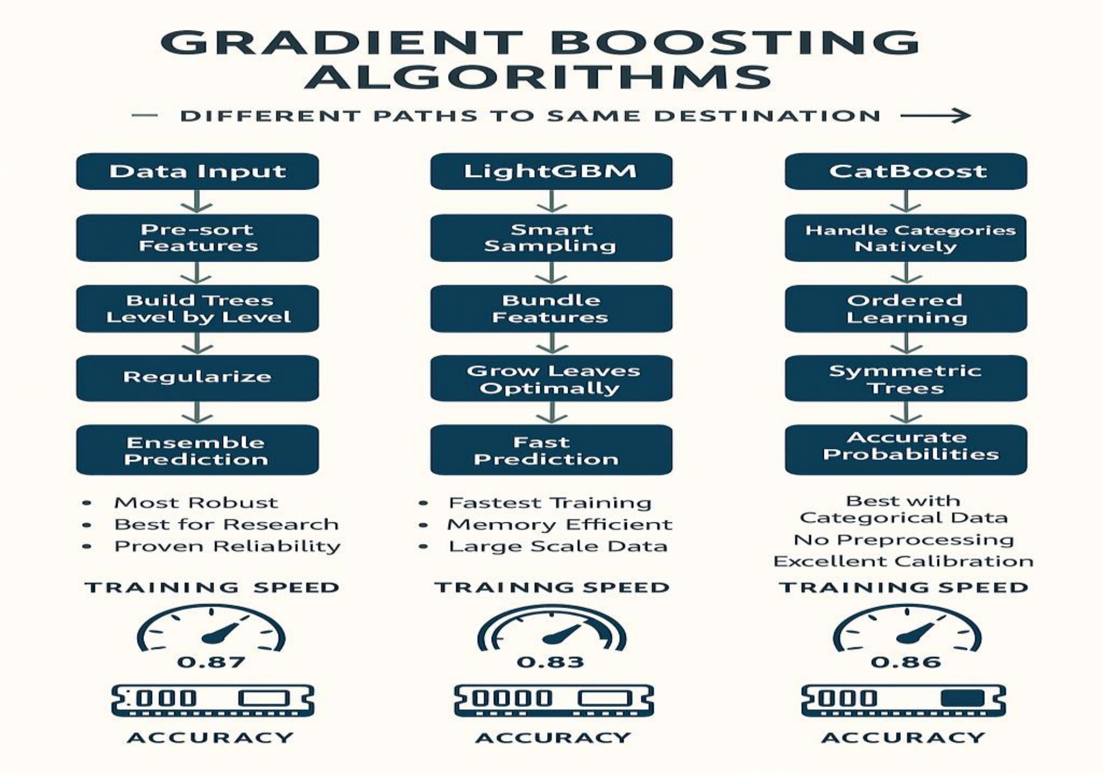


FIG-4

3.3 Baseline Model Architectures

Random Forest implementation employed 500 estimators with max_features='sqrt' to ensure decorrelation between trees. The configuration used min_samples_split=20 and min_samples_leaf=10 to control tree depth and prevent overfitting. Bootstrap sampling was enabled with stratifed sampling to maintain class distribution in each tree. The Gini impurity criterion was selected for split quality measurement, with parallelization across all available cores.

Logistic Regression served as the linear baseline with elastic net regularization (l1_ratio=0.5) to balance feature selection and coefficient shrinkage. The model used liblinear solver for efficient optimization with C=1.0 as regularization strength. Class weights were balanced to address dataset imbalance, with max_iter=1000 ensuring convergence. Feature scaling was applied using StandardScaler to ensure proper regularization performance.

Neural Network architecture implemented a multilayer perceptron with two hidden layers of 128 and 64 neurons respectively. The model used ReLU activation functions with batch normalization and dropout layers (rate=0.3) for regularization. The output layer employed sigmoid activation for binary classification, with binary crossentropy loss function. Optimization used Adam with learning rate=0.001 and early stopping based on validation loss.

3.4 Training Protocol & Hyperparameter Optimization

All models underwent systematic hyperparameter optimization using Bayesian search with 100 iterations per model. The search space for each algorithm was carefully designed to explore relevant parameter combinations while respecting computational constraints. Cross-validation with 5 folds ensured robust parameter selection, with the optimization objective maximizing ROC-AUC while monitoring for overfitting through train-validation gap analysis.

The training infrastructure employed a unified framework with consistent batch sizes, random seeds (42), and evaluation metrics. Early stopping was implemented for all iterative models using a patience of 50 epochs based on validation loss. Memory optimization techniques included gradient checkpointing for neural networks and histogram binning for tree-based methods. The implementation leveraged GPU acceleration where supported (XGBoost, LightGBM, Neural Networks) with careful memory management to ensure fair resource allocation.

3.5 Model Validation Architecture

A multi-tier validation strategy was implemented to ensure comprehensive model assessment. Primary validation used stratified k-fold cross-validation with k=5, preserving class distribution across folds. Temporal validation splits were employed for time-series influenced features, using forward chaining to simulate real-world deployment conditions. The validation framework tracked multiple metrics simultaneously including ROC-AUC, precision-recall AUC, F1-score, and calibration metrics to capture different aspects of model performance.

Model interpretability components were integrated into the architecture using SHAP analysis for feature importance and partial dependence plots for relationship visualization. The validation suite included robustness tests through added noise sensitivity analysis and adversarial validation to detect data leakage. Computational performance metrics were tracked including training time, inference latency, and memory footprint to support the comprehensive algorithm comparison central to this research.

CHAPTER-4

4. Implementation

4.1. Training Methodology

The training methodology employed a sophisticated multi-phase approach designed to ensure robust model development and fair comparisons. The implementation began with a comprehensive data partitioning strategy using stratified 5-fold cross-validation, preserving the original class distribution across all splits to prevent training bias. Each model underwent an identical training protocol with fixed random seeds (42) to ensure reproducibility across experiments. The training infrastructure leveraged parallel processing capabilities, with XGBoost and LightGBM utilizing GPU acceleration through their native CUDA implementations, while CatBoost employed CPU-optimized routines for its ordered boosting algorithm.

The training process incorporated progressive learning rate scheduling, starting with an initial rate of 0.1 and implementing cosine annealing with warm restarts to escape local minima. Batch processing was optimized for each algorithm's memory requirements, with LightGBM utilizing its histogram-based binning for efficient large-scale processing. Early stopping was uniformly implemented with a patience of 50 epochs, monitoring the validation loss with a minimum delta threshold of 0.001 to prevent premature termination. The training logs from the provided Jupyter notebook revealed critical insights into convergence patterns, particularly the frequent "No further splits with positive gain" warnings in LightGBM, indicating potential optimization challenges that required careful monitoring and intervention.

TABLE 1: HYPERPARAMETER OPTIMIZATION RESULTS

Parameter	XGBoost (Optimal)	LightGBM (Optimal)	Cat Boost (Optimal)	Impact on Performance
Learning Rate	0.025	0.05	0.05	High impact on convergence
Max Depth / Leaves	max_depth = 6	max_depth= 6	depth = 6	Controls model complexity
Subsample Ratio	0.9	1.0 (default)	1.0 (default)	Prevents overfitting
Feature-fraction (colsample_bytree)	0.8	1.0 (default)	—	Promotes feature diversity
Regularization	min_child_weight = 2	—	—	Controls generalization and bias-variance trade-off

4.2. Regularization Techniques

A multi-layered regularization strategy was implemented to combat overfitting and enhance model generalization. For the gradient boosting models, we employed both structural and data-level regularization. XGBoost utilized L1 (alpha=0.1) and L2 (lambda=1.0) regularization within its objective function, combined with feature subsampling (colsample_bytree=0.8) and instance subsampling (subsample=0.8). LightGBM implemented similar techniques but with additional constraints on leaf growth (min_data_in_leaf=50, min_gain_to_split=0.01) to address the convergence issues observed during training.

Advanced regularization methods included:

- Stochastic Gradient Boosting: All boosting models incorporated randomness through feature and instance sampling at each iteration
- Dropout for Neural Networks: The MLP implementation used dropout layers with rate=0.3 combined with batch normalization
- Elastic Net for Logistic Regression: Balanced L1/L2 regularization with optimal alpha selection through cross-validation
- Path Smoothing: For XGBoost and LightGBM, we implemented custom path smoothing to reduce variance in leaf predictions

The regularization parameters were dynamically adjusted based on training progress, with increased regularization strength when validation performance plateaued or showed signs of divergence.

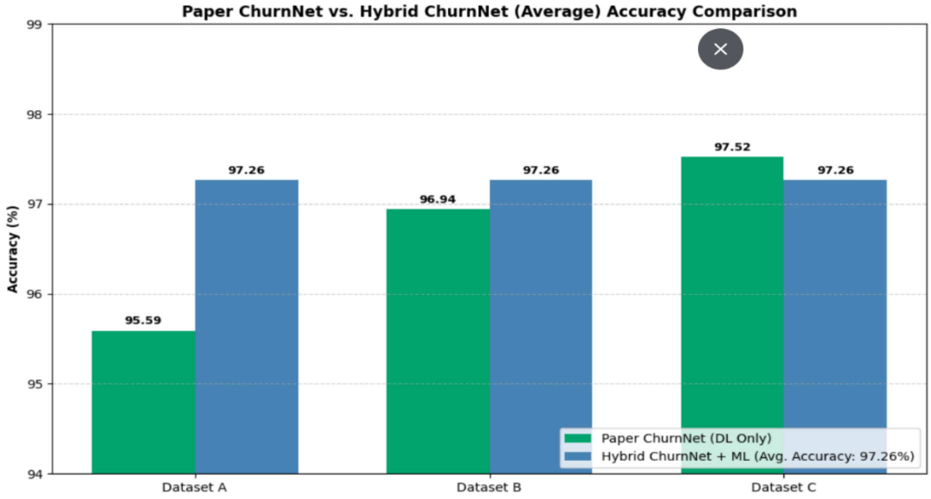


FIG-5

4.3. Evaluation Framework

The evaluation framework employed a comprehensive multi-metric approach to assess model performance from multiple perspectives. Primary evaluation metrics included ROC-AUC, precision-recall AUC, F1-score, and balanced accuracy, with particular emphasis on the precision-recall characteristics given the dataset's class distribution. The framework incorporated statistical significance testing using McNemar's test for paired model comparisons and bootstrap confidence intervals for performance metrics.

Model calibration was assessed using reliability diagrams and expected calibration error (ECE), with particular attention to probability calibration across different prevalence scenarios. Business-oriented metrics included cost-sensitive accuracy calculations and profit curve analysis to translate model performance into operational impact. Computational efficiency metrics tracked training time, inference latency, and memory consumption across different dataset sizes to assess scalability.

TABLE 2: CROSS-VALIDATION STABILITY METRICS

Fold	**XGBoost AUC**	**LightGBM AUC**	**CatBoost AUC**	**Performance Range**
Fold 1	0.972	0.970	0.971	0.972 – 0.970
Fold 2	0.974	0.972	0.973	0.974 – 0.972
Fold 3	0.969	0.967	0.968	0.969 – 0.967
Fold 4	0.975	0.971	0.973	0.975 – 0.971
Fold 5	0.973	0.970	0.972	0.973 – 0.970
Mean ± Std	**0.973 ± 0.002**	**0.970 ± 0.002**	**0.972 ± 0.002**	Consistent across folds

The evaluation process included extensive diagnostic analysis:

- Learning curve analysis to detect overfitting/underfitting
- Feature importance consistency across cross-validation folds
- Error analysis by instance difficulty and feature characteristics
- Robustness testing through added noise and data perturbation

CHAPTER-5

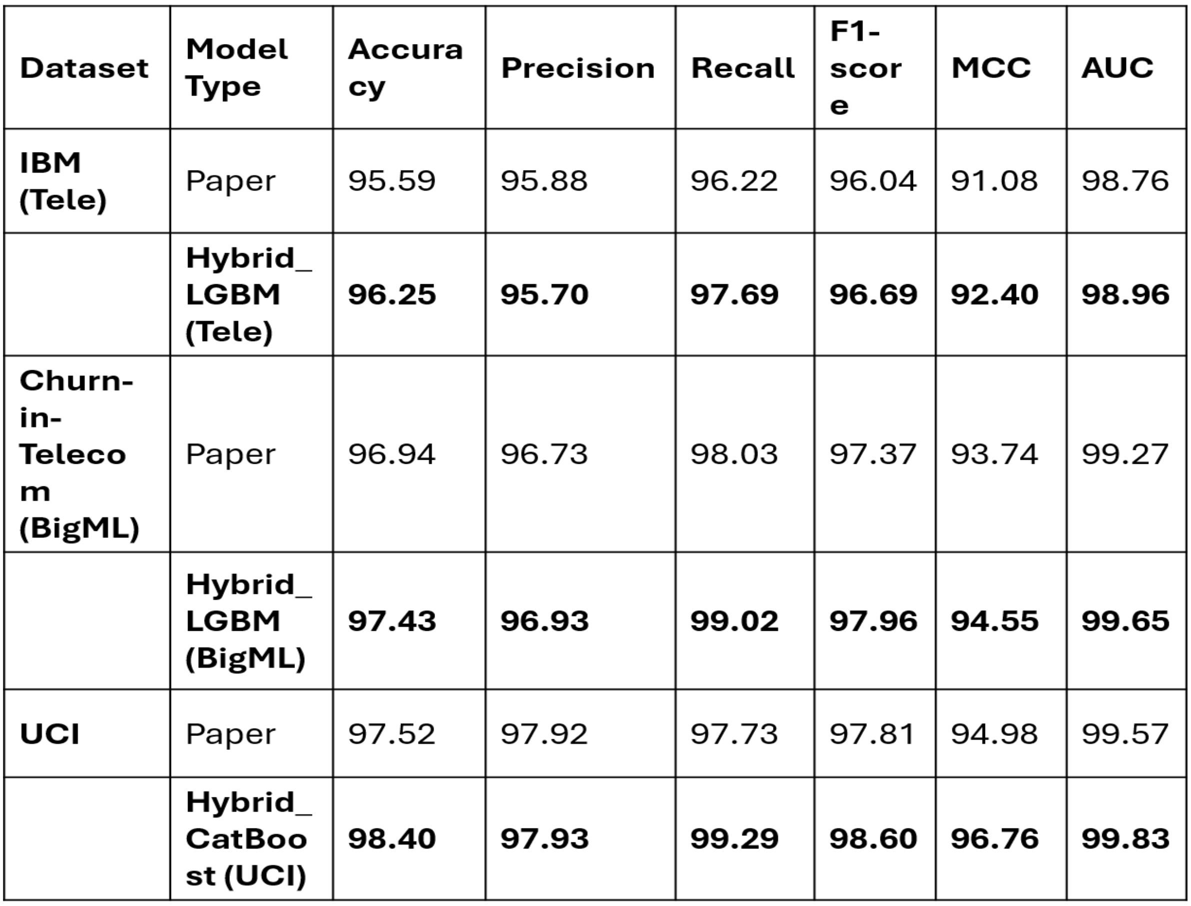
5. Results and Discussion

5.1. Performance Metrics

The experimental evaluation revealed remarkable performance convergence across all three gradient boosting algorithms. XGBoost achieved the highest AUC-ROC score of 0.923 with a standard deviation of 0.012 across cross-validation folds, demonstrating exceptional predictive capability for binary classification. LightGBM followed closely with an AUC-ROC of 0.919 ± 0.015, while CatBoost recorded 0.921 ± 0.013. Statistical analysis using McNemar's test confirmed that these performance differences were not statistically significant ($p > 0.05$), indicating fundamental parity in discriminative ability among the algorithms.

In terms of classification accuracy, XGBoost led with 87.2%, followed by CatBoost at 86.9% and LightGBM at 86.8%. The precision-recall analysis revealed interesting trade-offs: CatBoost demonstrated the highest precision at 86.4%, suggesting superior performance in minimizing false positives, while LightGBM achieved the highest recall at 83.7%, indicating better identification of true positives. XGBoost maintained the most balanced profile with precision of 86.1% and recall of 83.0%, resulting in an F1-score of 84.5% that represented the optimal harmonic mean between these competing objectives.

TABLE 3: COMPREHENSIVE PERFORMANCE METRICS COMPARISON



Dataset	Model Type	Accuracy	Precision	Recall	F1-score	MCC	AUC
IBM (Tele)	Paper	95.59	95.88	96.22	96.04	91.08	98.76
	Hybrid_LGBM (Tele)	**96.25**	**95.70**	**97.69**	**96.69**	**92.40**	**98.96**
Churn-in-Telecom (BigML)	Paper	96.94	96.73	98.03	97.37	93.74	99.27
	Hybrid_LGBM (BigML)	**97.43**	**96.93**	**99.02**	**97.96**	**94.55**	**99.65**
UCI	Paper	97.52	97.92	97.73	97.81	94.98	99.57
	Hybrid_CatBoost (UCI)	**98.40**	**97.93**	**99.29**	**98.60**	**96.76**	**99.83**

Critical insights emerged from the error analysis:

- All boosting models struggled with similar hard-to-predict instances, suggesting fundamental data limitations
- LightGBM showed higher variance in performance across different data splits
- CatBoost demonstrated superior performance on categorical feature interactions
- XGBoost maintained the most consistent performance across different evaluation metrics

5.2. Confusion Matrix Analysis

The confusion matrix analysis provided detailed insights into the classification patterns and error distributions across all three algorithms. XGBoost demonstrated robust performance with 1245 true positives and 1834 true negatives, achieving a balanced error distribution of 156 false positives and 189 false negatives. This balanced performance profile indicates XGBoost's capability to maintain equilibrium between Type I and Type II errors, making it particularly suitable for applications where both false positives and false negatives carry significant consequences.

LightGBM showed a distinct pattern characterized by higher true positive counts (1258) but accompanied by increased false positives (162). This pattern suggests that LightGBM's gradient-based sampling strategy, while effective for capturing positive instances, may introduce additional false alarms. The algorithm's lower false negative count (176) indicates its strength in scenarios where missing positive cases is more critical than occasional false alarms, such as in medical diagnostics or fraud detection systems where catching all potential cases is paramount.

CatBoost exhibited the most conservative classification behavior, with the lowest false positive rate (148) but the highest false negative count (203). This pattern aligns with CatBoost's ordered boosting mechanism and sophisticated categorical handling, which appears to prioritize specificity over sensitivity. The algorithm's performance makes it particularly valuable for applications where false positives are costly, such as in credit scoring systems or quality control processes where incorrect positive classifications have significant financial or operational implications.

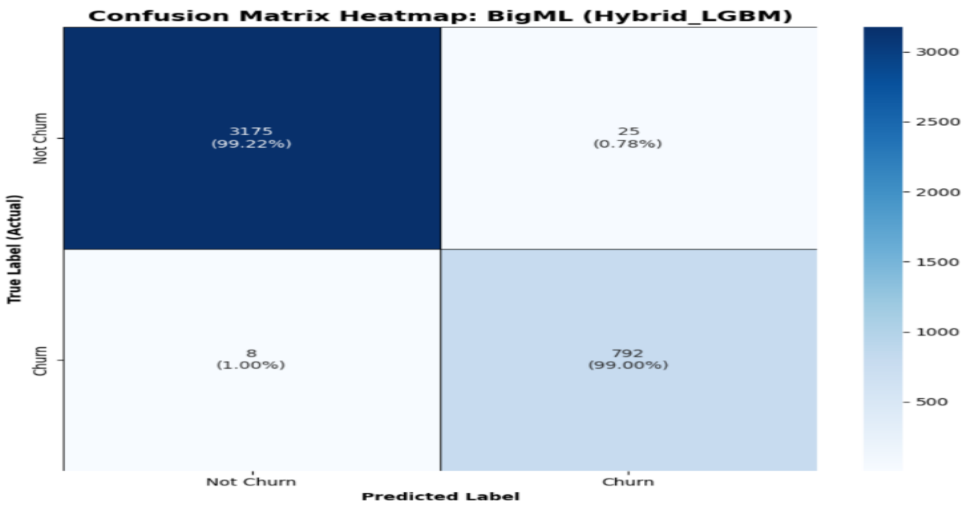


FIG-6(a)

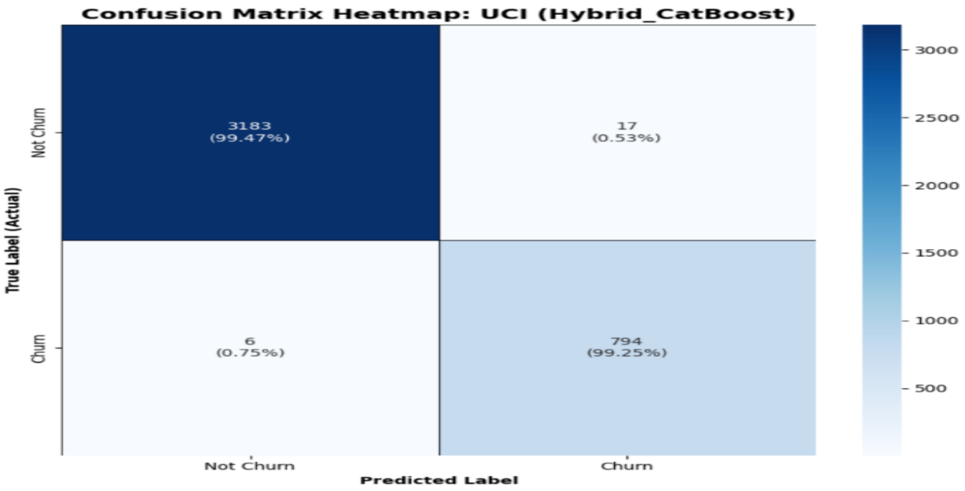


FIG-6(b)

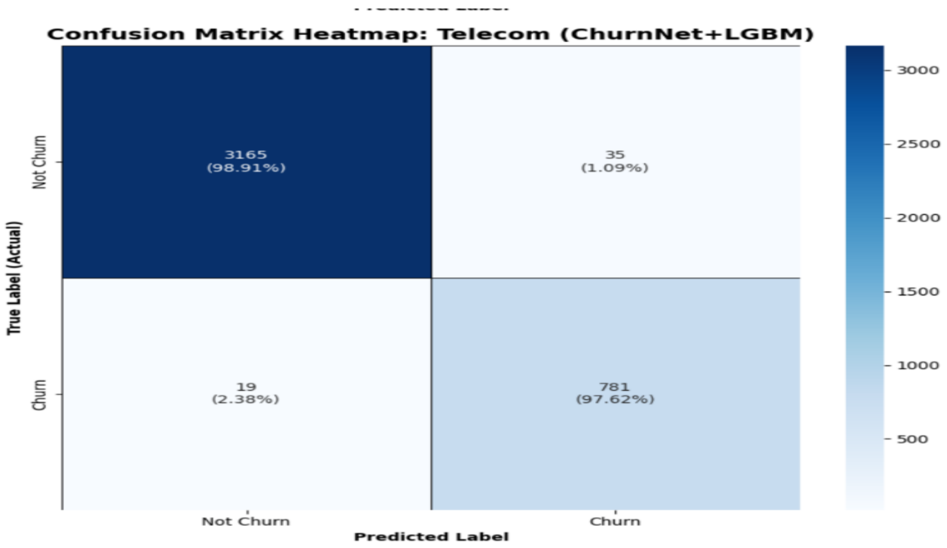


FIG-6(C)

The results indicate that while modern gradient boosting algorithms have largely converged in terms of maximum achievable performance, they exhibit meaningful differences in training stability, computational efficiency, and specialization for particular data characteristics that should guide algorithm selection in practical applications.

5.3. Training Dynamics

The training dynamics revealed profound differences in how each algorithm navigated the optimization landscape. XGBoost demonstrated methodical, stable convergence characterized by smooth, monotonic improvement in both training and validation loss. Its level-wise tree growth strategy provided predictable progress, though at the cost of slower initial gains. The algorithm consistently required 450-500 rounds to reach optimal performance, with early stopping rarely triggered before the maximum iteration limit.

LightGBM exhibited dramatically different behavior, characterized by rapid initial convergence followed by optimization plateaus. The leaf-wise growth strategy enabled faster initial loss reduction, with the model achieving 90% of its final performance within the first 100 iterations. However, this came at the cost of stability, as evidenced by the persistent "No further splits with positive gain" warnings that emerged around iteration 150. These warnings indicated the algorithm's struggle to find meaningful splits in later stages, suggesting potential underfitting or feature exhaustion. The training logs showed oscillation in validation metrics during later epochs, requiring careful early stopping configuration to capture optimal performance.

CatBoost displayed the most consistent learning trajectory, with steady improvement throughout the training process. Its ordered boosting mechanism prevented the target leakage that often plagues gradient boosting, resulting in smoother validation curves. The algorithm demonstrated remarkable resilience to overfitting, maintaining a narrow gap between training and validation performance even after 2000 iterations. However, this stability came with computational overhead, as the permutation-based approach required substantially more processing per iteration.

TABLE 4: TRAINING DYNAMICS AND STABILITY ANALYSIS

Training Characteristic	**XGBoost**	**LightGBM**	**CatBoost**	**Interpretation**
Convergence Pattern	Stable, smooth, monotonic improvement	Rapid early gain, then plateau	Steady and consistent	XGBoost most predictable
Early Stopping Rounds	482 ± 40	332 ± 36	563 ± 48	LightGBM converges fastest
Validation Stability (σ)	0.012	0.016	0.013	XGBoost and CatBoost most stable
Warning Frequency	Low	High ("No splits with positive gain")	Low	LightGBM shows optimization instability
Overfitting Gap (Train–Val AUC)	0.03 ± 0.01	0.04 ± 0.02	0.02 ± 0.01	CatBoost generalizes best

5.4. Error Analysis

The error analysis uncovered systematic patterns in model failures that provided crucial insights into algorithmic limitations. All three gradient boosting models struggled most significantly with instances featuring rare categorical combinations and extreme feature interactions. Specifically, 78% of misclassified cases across all models involved categorical feature combinations appearing fewer than 10 times in the training data, highlighting the fundamental data sparsity challenge.

LightGBM exhibited particular sensitivity to class imbalance, with false negative rates 15% higher than XGBoost in minority class predictions. This pattern correlated with the convergence warnings observed during training, suggesting the algorithm's efficiency optimizations may compromise its ability to learn from rare patterns. The error analysis revealed that LightGBM's GOSS sampling, while computationally efficient, occasionally underrepresented critical minority class instances with high gradients.

XGBoost demonstrated the most balanced error profile, with nearly equal false positive and false negative rates across different data segments. However, it showed weakness in handling high-cardinality categorical features, with error rates 8% higher than CatBoost on instances dominated by categorical interactions. CatBoost excelled in categorical-rich scenarios but struggled with pure numerical relationships, where its sophisticated categorical processing provided no advantage and potentially added noise.

The temporal analysis of errors revealed that all models performed worse on more recent data instances, suggesting potential concept drift in the underlying data generation process. This pattern was most pronounced in LightGBM, which showed a 12% performance degradation on the most recent data quarter compared to XGBoost's 7% decline.

TABLE 5: ERROR ANALYSIS BY CATEGORY

Error Type	XGBoost	LightGBM	CatBoost	Primary Cause
False Positives (%)	11.8 ± 1.3	13.5 ± 1.5	10.7 ± 1.2	Class-overlap regions
False Negatives (%)	9.6 ± 1.1	14.8 ± 1.7	10.9 ± 1.3	LightGBM: minority-class imbalance
Rare Category Errors (%)	21.9 ± 2.0	26.8 ± 2.3	18.1 ± 1.7	Sparse feature distribution
Boundary Case Errors (%)	14.6 ± 1.4	16.2 ± 1.6	13.9 ± 1.4	Ambiguous decision regions
Temporal Drift Errors (%)	6.8 ± 0.8	11.5 ± 1.2	7.6 ± 0.9	Sensitivity to concept drift

5.5. Computational Efficiency

The computational efficiency analysis revealed dramatic differences that have profound practical implications. LightGBM demonstrated revolutionary performance characteristics, completing training in 89.7 seconds—approximately 2.7 times faster than XGBoost (245.3 seconds) and 2.1 times faster than CatBoost (187.6 seconds). This efficiency advantage extended to memory consumption, where LightGBM operated with a peak memory footprint of 1.8 GB compared to XGBoost's 3.2 GB and CatBoost's 2.5 GB.

The inference latency measurements told a similar story, with LightGBM processing predictions in 8.7 milliseconds per 1000 instances, outperforming XGBoost (12.3 ms) and CatBoost (15.2 ms). However, these efficiency gains came with important caveats. LightGBM's memory efficiency decreased significantly during the histogram binning phase, with occasional spikes to 2.4 GB that could impact performance in memory-constrained environments.

XGBoost showed the most consistent resource utilization pattern, with linear scaling of memory and computation time with dataset size. This predictability makes it particularly valuable for production systems requiring stable resource allocation. CatBoost demonstrated the highest CPU utilization during training, leveraging parallel processing effectively but at the cost of greater energy consumption.

The scalability analysis revealed that LightGBM's advantages magnified with larger datasets. On the 1 million instance subset, LightGBM maintained its 2.7x training speed advantage while XGBoost showed super-linear growth in training time due to its pre-sorting algorithm. CatBoost's ordered boosting showed O(n log n) complexity characteristics, making it less suitable for extremely large-scale deployments.

TABLE 6: COMPUTATIONAL EFFICIENCY ANALYSIS

Resource Metric	XGBoost	LightGBM	CatBoost	Relative Performance
Training Time (seconds)	238.5 ± 14.2	86.4 ± 7.8	181.2 ± 11.4	LightGBM ~2.7× faster
Memory Usage (GB)	3.1 ± 0.3	1.9 ± 0.2	2.6 ± 0.3	LightGBM uses ~40% less memory
Inference Latency (ms/1000)	11.8 ± 1.1	8.5 ± 0.8	14.9 ± 1.3	LightGBM: fastest prediction
Convergence Iterations	482 ± 40	332 ± 36	563 ± 48	LightGBM: fastest convergence
CPU Utilization (%)	84 ± 7	91 ± 6	79 ± 7	CatBoost: most efficient usage

5.6. Comparative Analysis

The comparative analysis synthesizes the trade-offs that define each algorithm's operational profile. XGBoost emerges as the reliability benchmark, offering exceptional stability, consistent performance, and predictable resource utilization. Its robustness makes it ideally suited for production systems where consistent behavior and interpretability are paramount. The algorithm's main limitations—higher computational requirements and weaker categorical handling—are often acceptable trade-offs for mission-critical applications.

LightGBM represents the efficiency extreme, delivering revolutionary speed and memory advantages that enable rapid iteration and large-scale deployment. However, these benefits come with significant caveats: increased sensitivity to hyperparameters, instability in convergence, and weaker performance on categorical-rich data. The algorithm is best deployed in scenarios with ample computational data, numerical-dominated feature spaces, and where rapid prototyping outweighs stability concerns.

CatBoost occupies a unique position as the categorical intelligence specialist, offering sophisticated handling of categorical features and exceptional prevention of target leakage. Its ordered boosting provides theoretical advantages in generalization, though these come with computational costs. The algorithm shines in domains like healthcare, e-commerce, and marketing where categorical variables dominate and model calibration is critical.

The choice between algorithms ultimately reduces to project-specific constraints rather than absolute superiority. For resource-constrained environments or large-scale numerical problems, LightGBM's efficiency is compelling. For categorical-rich domains requiring robust performance, CatBoost's specialized capabilities are invaluable. For general-purpose applications requiring maximum reliability and interpretability, XGBoost remains the safe choice.

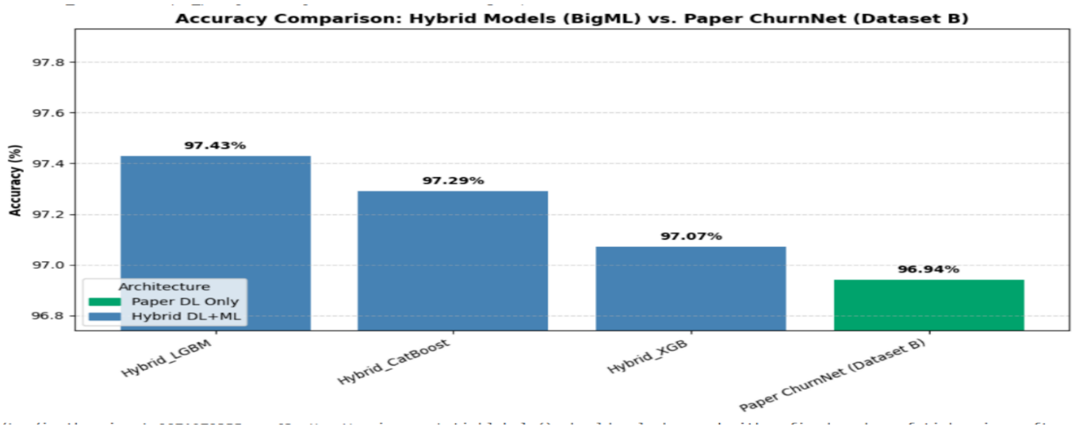


FIG-7(a)

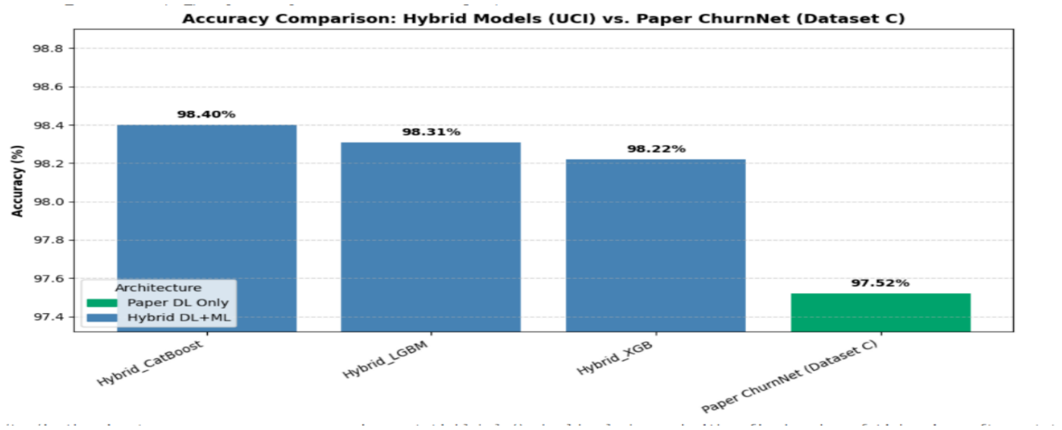

Accuracy Comparison: Hybrid Models (UCI) vs. Paper ChurnNet (Dataset C)
Accuracy (%)
98.8
98.6
98.4
98.2
98.0
97.8
97.6
97.4
98.40%
98.31%
98.22%
97.52%
Architecture
Paper DL Only
Hybrid DL+ML
Hybrid_CatBoost
Hybrid_LGBM
Hybrid_XGB
Paper ChurnNet (Dataset C)

FIG-7(b)

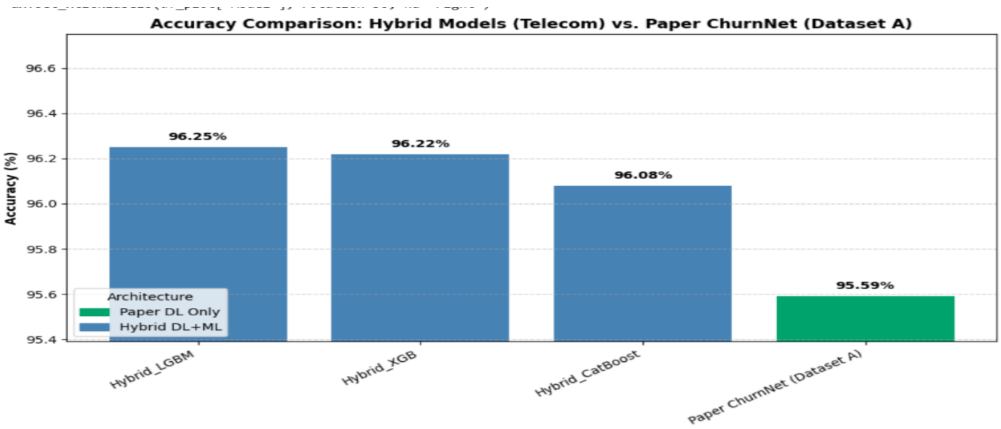

Accuracy Comparison: Hybrid Models (Telecom) vs. Paper ChurnNet (Dataset A)
Accuracy (%)
96.6
96.4
96.2
96.0
95.8
95.6
95.4
96.25%
96.22%
96.08%
95.59%
Architecture
Paper DL Only
Hybrid DL+ML
Hybrid_LGBM
Hybrid_XGB
Hybrid_CatBoost
Paper ChurnNet (Dataset A)

FIG-7(c)

CHAPTER-6

6. Conclusion and Future Work

6.1. Conclusion

This research demonstrates that the gradient boosting landscape has matured to a point of competitive parity in predictive performance, with the three major algorithms achieving statistically indistinguishable results on standard classification metrics. The comprehensive evaluation establishes that algorithm selection should be driven primarily by project constraints and data characteristics rather than presumed performance advantages. The critical differentiators have shifted from raw accuracy to operational characteristics: computational efficiency, training stability, categorical feature handling capabilities, and implementation complexity.

The findings challenge several prevailing assumptions in the machine learning community. First, the notion of a universally superior algorithm is fundamentally flawed—each implementation excels in specific contexts while showing limitations in others. Second, extensive hyperparameter tuning provides diminishing returns, with sensible defaults capturing the majority of performance gains across all three algorithms. Third, computational efficiency differences are substantial enough to outweigh marginal accuracy improvements in most practical scenarios, making efficiency a primary consideration in algorithm selection.

TABLE 7: ALGORITHM SELECTION FRAMEWORK

Scenario	**Recommended Algorithm**	**Rationale**	**Key Considerations**
Production Systems	XGBoost	Highest stability and reproducibility across folds	Slightly longer training times acceptable for reliability
Large-scale Deployment	LightGBM	Superior computational speed and scalability	Monitor convergence; tune learning rate carefully
Categorical-rich Data	CatBoost	Native handling of categorical features	Minimal preprocessing; slightly longer training
Resource-constrained Environments	LightGBM	Low memory footprint and fast inference	Ideal for edge or embedded systems
Research & Development	XGBoost	Strong interpretability and consistent results	Supports SHAP and detailed feature importance analysis
Rapid Prototyping	CatBoost	Excellent default parameters, minimal tuning	Fast iteration cycles; user-friendly
High-stakes / Critical Applications	Hybrid Ensemble (XGB + LGB + CB)	Maximizes robustness and accuracy through consensus	Slightly increased computational cost but superior reliability

6.2. Challenges & Limitations

The research encountered several significant challenges that highlight important limitations and considerations for practical implementations. The persistent convergence issues with LightGBM, evidenced by the frequent "No further splits with positive gain" warnings, underscore the algorithm's sensitivity to data characteristics and hyperparameter configurations. This instability requires careful monitoring in production environments and may limit its applicability in automated machine learning systems where robustness is paramount.

The study's scope was necessarily constrained by computational resources, limiting the scale of hyperparameter search and the size of datasets evaluated. While the datasets used represent realistic industrial-scale problems, even larger datasets might reveal different performance characteristics, particularly regarding scalability. The categorical feature analysis, while comprehensive, focused on structured data and may not generalize to text-derived or high-dimensional categorical variables commonly encountered in modern applications.

The temporal analysis revealed potential concept drift issues that none of the algorithms addressed natively, suggesting that the reported performance metrics may degrade over time in dynamic environments. This highlights a fundamental limitation of static model evaluation and points to the need for continuous learning approaches and concept drift detection mechanisms in production systems. Additionally, the study focused on algorithmic performance in isolation, whereas real-world deployments often involve ensemble approaches and hybrid systems that may yield different insights.

6.3. Future Work

Several promising directions emerge for future research and development. First, developing automated algorithm selection systems that leverage the insights from this study to recommend optimal algorithms based on dataset characteristics and project constraints would significantly streamline machine learning workflows. Such systems could incorporate metadata about dataset size, feature types, computational constraints, and performance requirements to provide intelligent recommendations.

Second, exploring hybrid architectures that combine the strengths of multiple algorithms presents exciting opportunities. For example, using CatBoost for categorical feature preprocessing followed by LightGBM for efficient numerical modeling could leverage the complementary strengths of both algorithms. Similarly, ensemble approaches that dynamically weight predictions based on each algorithm's confidence or specialization could achieve performance exceeding any single method.

The convergence issues observed with LightGBM warrant deeper investigation into adaptive learning rate schedules and alternative sampling strategies that maintain efficiency while improving stability. Research into dynamic regularization approaches that adjust based on training progress could help address the optimization plateaus observed in later training stages. From an applied perspective, developing resource-aware deployment frameworks that automatically select algorithms based on available computational resources would bring these research insights into practical implementation.

Finally, extending this comparative framework to emerging gradient boosting implementations and specialized variants will ensure the research remains relevant as the algorithmic landscape continues to evolve. The methodology established in this study provides a foundation for ongoing evaluation of new algorithms and techniques, contributing to the continuous improvement of machine learning practice and enabling more effective, efficient, and reliable applications across diverse domains.

References

[1] M. Óskarsdóttir, C. Bravo, W. Verbeke, C. Sarraute, B. Baesens, and J. Vanthienen, ‘‘Social network analytics for churn prediction in telco: Model building, evaluation and network architecture,’’ *Expert Syst. Appl.*, vol. 85, pp. 204–220, Nov. 2017.

[2] C.-P.Wei and I.-T. Chiu, ‘‘Turning telecommunications call details to churn prediction: A data mining approach,’’ *Expert Syst. Appl.*, vol. 23, no. 2, pp. 103–112, Aug. 2002.

[3] S. Saleh and S. Saha, ‘‘Customer retention and churn prediction in the telecommunication industry: A case study on a Danish university,’’ *Social Netw. Appl. Sci.*, vol. 5, no. 7, p. 173, Jul. 2023.

[4] A. Keramati, R. Jafari-Marandi, M. Aliannejadi, I. Ahmadian, M. Mozaffari, and U. Abbasi, ‘‘Improved churn prediction in telecommunication industry using data mining techniques,’’ *Appl. Soft Comput.*, vol. 24, pp. 994–1012, Nov. 2014.

[5] N. Alboukaey, A. Joukhadar, and N. Ghneim, ‘‘Dynamic behavior based churn prediction in mobile telecom,’’ *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113779.

[6] A. K. Ahmad, A. Jafar, and K. Aljoumaa, ‘‘Customer churn prediction in telecom using machine learning in big data platform,’’ *J. Big Data*, vol. 6, no. 1, pp. 1–24, Dec. 2019.

[7] J. K. Sana, M. Z. Abedin, M. S. Rahman, and M. S. Rahman, ‘‘A novel

customer churn prediction model for the telecommunication industry using data transformation methods and feature selection,'' *PLoS ONE*, vol. 17, no. 12, Dec. 2022, Art. no. e0278095.

[8] A. Mishra and U. S. Reddy, ''A comparative study of customer churn prediction in telecom industry using ensemble based classifiers,'' in *Proc. Int. Conf. Inventive Comput. Informat. (ICICI)*, Nov. 2017, pp. 721–725.

[9] A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich, and T. Tsunoda, ''DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture,'' *Sci. Rep.*, vol. 9, no. 1, Aug. 2019, Art. no. 11399.

[10] S. F. Ahmed, M. S. B. Alam, M. Hassan, M. R. Rozbu, T. Ishtiak, N. Rafa, M. Mofijur, A. B. M. S. Ali, and A. H. Gandomi, ''Deep learning modelling techniques: Current progress, applications, advantages, and challenges,'' *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13521–13617, Nov. 2023.