Studies On Probabilistic Forecasting Methods Employing Deep Learning Models

Biswaranjan Dash

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Studies On Probabilistic Forecasting Methods Employing Deep Learning Models

Progress Report - March 2025

submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Biswaranjan Dash

(Roll Number: 122cs0557)

based on research carried out

under the supervision of

Prof. Sujata Mohanty

May, 2025

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Sujata Mohanty
Professor

May 16, 2025

Supervisors' Certificate

This is to certify that the work presented in the progress report entitled *Studies On Probabilistic Forecasting Methods Employing Deep Learning Models* submitted by *Biswaranjan Dash*, Roll Number 122cs0557, is a record of original research carried out by him under our supervision and guidance in partial fulfillment of the requirements of the degree of *Bachelor of Technology* in *Computer Science and Engineering* . Neither this project report nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Sujata Mohanty
Professor

Dedication

I dedicate this thesis to my cherished family and friends, whose steadfast love and unwavering support have been my guiding force throughout this M.Tech journey. Your encouragement, understanding and patience have fueled my determination, and your belief in me has been my greatest motivation.

To my family, for being my constant pillars of strength, for the sacrifices made, and for the boundless love that knows no bounds, I owe you the world. Thank you also for providing me with a computer early in my life, igniting my passion for technology.

To my friends, the invaluable gems in my life, for the laughter, the late night discussions, and for standing by me in both the highs and lows. Thank you for being the uplifting force that makes this journey memorable.

This thesis stands as a testament to the collective efforts and sacrifices of my family and friends. Your love has been the driving force behind every achievement, and I am grateful beyond words for the privilege of having you all in my life.

With heartfelt gratitude,
Biswaranjan Dash

Declaration of Originality

I, *Biswaranjan Dash*, Roll Number *122cs0557* hereby declare that this project report entitled *Studies On Probabilistic Forecasting Methods Employing Deep Learning Models* presents my original work carried out as a student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the thesis. Works of other authors cited in this thesis have been duly acknowledged under the sections ‘‘Reference’’ or ‘‘Bibliography’’. I have also submitted my original research records to the scrutiny committee for evaluation of my thesis.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present thesis.

Oct 21, 2025
NIT Rourkela

Biswaranjan Dash

Acknowledgement

I would like to express my sincere gratitude to all those who have supported me in my ongoing efforts on this project. I am particularly indebted to our project supervisor for the final year, ***Professor Sibarama Panigrahi***, whose invaluable suggestions and unwavering support have played a pivotal role in guiding me through the process of conducting this research. His encouragement has continually inspired me to work diligently and push boundaries.

In addition, I wish to extend my heartfelt appreciation to the dedicated personnel of the ***Department of Computer Science and Engineering*** for granting me access to the essential equipment and materials necessary for this project.

Lastly, I would like to convey my profound gratitude to my parents and friends for their continuous encouragement throughout my academic journey and during this project. I am also deeply thankful to the **National Institute of Technology, Rourkela** for their backing and for awarding me the opportunity to undertake this project, along with the provision of essential facilities.

May 16, 2025 *Biswaranjan Dash*
NIT Rourkela Roll Number: 122CS0557

Abstract

This project proposes a secure remote patient monitoring system design and its subsequent implementation employing a lightweight blockchain-enabled authentication approach. The described system keeps on recording patients' health parameters like body temperature, oxygen saturation, and so on, through biomedical sensors connected to an Arduino UNO and an ESP Wi-Fi module. After the data collection, it is stored in a cloud platform (ThingSpeak) securely for the purpose of real-time storage and analysis. The use of a blockchain-based authentication scheme along with the use of Elliptic Curve Cryptography (ECC) is a method that guarantees the fact that only the devices that have been verified and hospital servers have the capability to access or transmit sensitive medical data. To this end, the integrity of the data, its confidentiality, and the privacy of the user are improved while simultaneously, the computational costs are kept low which is very important for resource-constrained IoT devices. After that the hospital server fetches the legitimate data from the cloud, thus the doctors will be able to monitor the patients remotely and make the necessary clinical decisions timely.

The proposed system demonstrates an efficient, scalable, and secure solution for healthcare IoT applications, effectively preventing common network attacks such as replay, impersonation, and man-in-the-middle attacks.

Keywords: Remote Patient Monitoring, Internet of Medical Things (IoMT), Blockchain, Lightweight Authentication, Elliptic Curve Cryptography (ECC), Secure Healthcare System.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Remote patient monitoring (RPM) is a direct consequence of the increasing usage of technology in healthcare. It is the most effective means for a patient's health to be tracked without a need for frequent hospital visits. The doctors by using IoT (Internet of Things) devices such as sensors and microcontrollers are able to gather live data such as temperature, oxygen levels or heart rate of the patients who are at home or in far-off places. This is extremely valuable because it leads to the prevention of health issues as well as the provision of medical support on time by doctors.

On the other hand, when medical data is sent through the internet, it becomes necessary to guarantee security and privacy of such data. If there is any kind of unauthorized access or data leakage, the result would be the risk of the patient's safety. Thus, the security subsystem in the system should be not only strong in terms of protection but also be efficient enough to work on small, low-power IoT devices.

Our work is to introduce a secure remote patient monitoring system implementation that would support safety and efficiency of the communication process through the use of blockchain technology and Elliptic Curve Cryptography (ECC). The Arduino UNO and the ESP Wi-Fi module that are connected with sensors make up the first part of the system, which is used to collect patient data that is then uploaded to a cloud platform (ThingSpeak). After that, the cloud sends the data to the hospital server, which is the doctor's place, and they are able to remotely monitor the patient's condition through the data received. The usage of blockchain guarantees that only authorized users and devices have access to the data, whereas ECC is the reason for quick and simple data encryption in an IoT environment.

1.2 Objectives

The primary goals envisaged through this undertaking are:

- To conceive and implement a remote patient monitoring system that gathers real-time health information through IoT sensors.
- To maintain data security and confidentiality by implementing a blockchain-enabled authentication method and using light cryptography (ECC).
- To establish good communication pathways between patient equipment, cloud and hospital server.
- To offer doctors an opportunity to remotely access patient medical records via a safe and user-friendly system.
- To trim down the computational load thereby making it possible for the system to function efficiently on hardware capable of limited resources such as Arduino and ESP modules.

1.3 Organization Of Thesis

The organization of this thesis is structured as follows to provide a comprehensive understanding of the proposed system and its implementation.

Chapter 2

Literature Review

2.1 Overview

Remote Patient Monitoring (RPM) is the vital element of the Internet of Medical Things (IoMT), a concept in which intelligent devices keep track of a patient's health continuously and the updated information is sent to the respective doctors or hospitals [?]. Different researchers throughout the years have come up with a variety of methods to ensure that such systems are not only secure but also dependable.

The first generation of systems were mainly concerned with gathering and transferring patients' data done through Wi-Fi or Bluetooth while the aspects of privacy and security were only partially taken care of [?]. Consequently, healthcare data being highly sensitive even the slightest negligence of it would lead to a situation where data leakage, impersonation, or unauthorized access would occur.

2.2 Review of Previous Works

2.2.1 Initial IoT-based Healthcare Systems

Ancestor systems were implemented on basic IoT schematics consisting of sensors and cloud platforms (example: ThingSpeak or Blynk) [?]. These frameworks were effective in monitoring vital signs; however, due to the absence of strict authentication and encryption measures, they were susceptible to hacking.

2.2.2 Password-Based Authentication Schemes

A number of later projects incorporated password-protected authentication to secure communications [?]. That said, the security provided by such methods was very weak because passwords can be easily guessed or stolen, thus leading to scenarios where the perpetrator can impersonate the victim or record/replay the communication.

2.2.3 Public Key Cryptography (RSA) Based Approaches

The next step was adoption heavy cryptographic schemes like RSA or alike to secure patient data [?]. The drawback of this approach was that these strong security measures demanded high computational power conflicting with the low-power nature of devices like Arduino or ESP modules.

2.2.4 Lightweight Cryptography & ECC-Based Methods

Researcher solved this problem by switching to Elliptic Curve Cryptography (ECC), an algorithm achieving the same security level as RSA but with smaller key sizes and faster processing speed [?]. Presently, ECC is almost the standard for IoT applications, including healthcare.

2.2.5 Blockchain-Enabled Systems

Recent studies present medical data management with the help of blockchain technology [?]. Blockchain functions as an incorruptible ledger, keeping track of data transactions and making them transparent, thus, trustworthy, and secure against hacker attacks. When combined with ECC, blockchain delivers a decentralized and lightweight authentication solution that suits real-time medical monitoring requirements [?].

2.3 Research Gap

The analyzed papers highlight that while the developed platforms facilitate data gathering and sharing, security is still an open issue and rarely is the trade-off between security and efficiency addressed. Some works call for significant computational resources, others are vulnerable to certain types of attacks. The key challenges identified include:

- Need to operate under constrained energy of IoT devices,
- Implementation of lightweight cryptographic techniques to ensure security,
- Use of blockchain technology not only for security purposes but also to facilitate user authentication, and
- Being able to guarantee privacy and integrity of data during a continuous patient monitoring scenario.

Table 2.1 presents a comparative analysis of various authentication schemes proposed for IoT-based healthcare systems over recent years. The table highlights the progression from basic password-based methods to more sophisticated approaches incorporating ECC and blockchain technology, culminating in the present work which addresses the limitations of previous approaches.

Table 2.1: Comparative Analysis of Authentication Schemes in IoT-based Healthcare Systems

Year	Authors / Paper	Technique Used	Focus Area	Limitations
2018	Adavoudi-Jolfaei et al.	Three-factor authentication using passwords and biometrics	Wireless Sensor Networks in healthcare	High computation time; not suitable for low-power devices
2019	Gupta et al.	Lightweight ECC-based user authentication	Wearable IoT devices	Lacked decentralized data protection
2020	Yu et al.	Password-based DRM authentication	IoT Applications	Weak against replay and impersonation attacks
2021	Sahoo et al.	Three-factor IoT authentication using ECC	Healthcare Systems	Secure but lacked blockchain integration
2022	Mirsaraei et al.	Secure three-factor authentication for IoT	IoT-based healthcare	Needed more scalability and efficiency
2023	Berini et al.	Hyperelliptic curve-based IoD authentication	Internet of Drones	Good security but complex math operations
2024	Zhang et al.	PUF-based lightweight authentication	Internet of Drones	Focused on drones, not healthcare
Present Work (2025)	**Current Project**	**Blockchain + ECC-based lightweight authentication**	**Remote Patient Monitoring System**	**Addresses past issues; provides both efficiency and strong security**

Chapter 3

Parametric and Non-parametric Probabilistic Forecasting Methods Employing Deep Learning Models

3.1 Motivation

Forecasting is crucial across many fields, from energy management and finance to medicine and climatology. Accurate predictions allow stakeholders to allocate resources in turn and make intelligent choices. Traditional forecasting techniques, however, provide single-point estimates that omit uncertainty, which can lead to suboptimal or even detrimental choices, especially in high-stakes applications such as weather forecasting and financial markets. In this chapter, we aim to conduct a study on the various existing parametric and non-parametric probabilistic forecasting methods using deep learning models across a variety of datasets to assess performance using various metrics.

Probabilistic forecasting is a better option because it provides confidence intervals that allow decision-makers to plan for a range of potential outcomes and control risks. The technique, however, is not yet a widespread practice in machine learning, partly due to the lack of a shared understanding of the optimum method. While parametric techniques (e.g., Gaussian, Weibull) are simpler to apply, they might not be able to capture data complexities fully; non-parametric techniques (e.g., Quantile Regression, Bootstrap based, etc.), conversely, are more adaptable but demand more comprehensive computational resources.

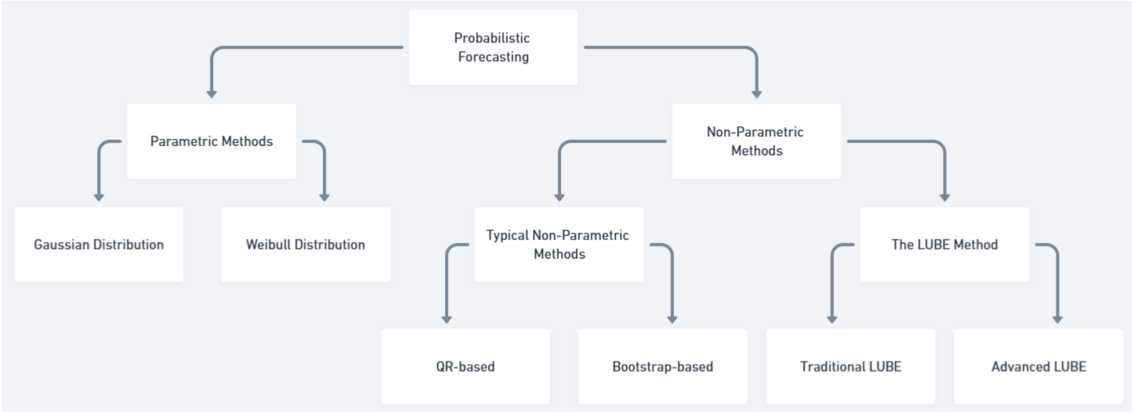


Figure 3.1: Classification of Traditional Probabilistic Methods.

Non-parametric techniques, such as the Traditional and Advanced LUBE methods, have drawn interest as well, but accuracy remains an issue compared to traditional methods. The growing significance of machine learning to probabilistic forecasting highlights the need for more research to determine the most effective methods across different applications.

Figure 3.1 presents a classification of traditional probabilistic forecasting methods and Table 3.1 represents the datasets used in this Thesis and their length.

Table 3.1: Summary of Datasets Used.

Dataset Name	**Length of Dataset**
Nifty50 Dataset (2000–2021)	
Adani Ports	3322
Asian Paints	5306
Axis Bank	5306
Web Traffic	550
Electricity Consumption	1858

3.2 Methodology

3.2.1 Categorization Of Methods

In this paper, six probabilistic forecasting methods are compared, broadly classified into parametric and non-parametric, each capable of producing prediction intervals (PIs) representing uncertainty in time series forecasting. Parametric methods are based on assumptions of underlying data distribution. The Gaussian (Normal) distribution-based method assumes symmetric, bell-shaped data and builds PIs from mean and standard deviation of forecast values and thus is optimal for datasets having normally distributed errors. The Weibull distribution-based method is more general and especially suited for skewed data and finds extensive use in reliability analysis and applications dealing with extreme values or variability in the data. Non-parametric methods make fewer distributional assumptions and thus find broader applicability for a wide variety of datasets. The Quantile Regression (QR)-based method estimates quantiles directly (e.g., 0.1 and 0.9 for a 90% confidence level) to build PIs and provides robustness to non-linearity and heteroscedasticity. The Bootstrap-based method employs resampling of training data to make multiple forecasts and estimates PIs on the basis of percentiles of the resampled outputs and hence captures model and data dependent uncertainties effectively. The Traditional LUBE method builds PIs upon a custom loss function to reduce interval width and ensure adequate coverage, trading off tightness and reliability of intervals. The Advanced LUBE method builds on this by optimizing performance using novel metrics like Absolute Coverage Error (ACE) and Absolute Width Error (AWE), providing more robustness to outliers and stability at multiple confidence levels. These methods, collectively, provide a comprehensive probabilistic forecasting framework under a wide range of data conditions and uncertainty profiles.

3.2.2 Deep Learning Models

To implement the aforementioned probabilistic forecasting techniques, four deep learning models were utilized: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN) and Bidirectional LSTM (BiLSTM). LSTM is a recurrent neural networks (RNNs) architecture specifically developed to learn long term temporal relationships in sequential data and is hence well suited for time series with intricate temporal patterns. GRU is a compact version of LSTM and hence possesses similar capabilities but lower computational complexity and is thus suitable for applications with low memory demands. CNNs, while initially developed for spatial data, have been used successfully for time series by encoding temporal sequences as structured inputs, allowing strong feature extraction through localized pattern detection. BiLSTM is an extension of LSTM with the ability to process input sequences in both directions and hence extract contextual information from past and future time steps for enhanced forecasting accuracy. These architectures were chosen based on their proven efficiency in time series forecasting tasks and their compatibility with both parametric and non-parametric probabilistic techniques.

3.2.3 Methodology

- **Dataset Preparation** In all the approaches described in the present study, the data preparation was performed consistently. Five different datasets were used, three stock price datasets of Nifty50 Dataset (2000-2021): Adani Ports, Asian Paints and Axis Bank and two other datasets: the Electricity Consumption dataset and the Web Traffic dataset 3.1. The respective target variables for prediction were 'VWAP' for the stock price datasets, 'Consumption' for the electricity dataset, and 'Views' for the web traffic dataset. The target variables were normalized first with Min-Max Scaling to provide stable convergence of the models. Next, a sliding window technique was employed to transform the time series data into a supervised learning format, thus generating several input-output pairs suitable for model training. The generated records were divided equally into training, validation, and test sets using a 70:15:15 ratio for all experimental protocols.

- **Evaluation Metrics** In this thesis, we have used the following Evaluation Metrics for fair comparison across all the methods:
  1. PICP (Prediction Interval Coverage Probability):
  Quantifies the proportion of actual values that are contained within the defined prediction intervals.

$$\text{PICP} = \frac{\text{Number of samples where } LB_i \leq y_i \leq UB_i}{n} \tag{3.1}$$

Where:
y_i: The actual observed value at time step i.
LB_i: The predicted **lower bound** of the prediction interval at time step i.
UB_i: The predicted **upper bound** of the prediction interval at time step i.

2. PINAW (Prediction Interval Normalized Average Width):
This metric assesses the precision of the intervals.

$$\text{PINAW} = \frac{\frac{1}{n}\sum_{i=1}^{n}(UB_i - LB_i)}{\max(y) - \min(y)} \tag{3.2}$$

Where:
n: Total number of forecasted data points or time steps.
$\max(y)$, $\min(y)$: The maximum and minimum values of the observations y_i across all n samples. These are used to normalize the interval width in PINAW.

3. ACE (Absolute Coverage Error):
Defines the difference between actual coverage and nominally assumed confidence.

$$\text{ACE} = |\text{PICP} - c| \tag{3.3}$$

Where c: The nominal confidence level of the prediction interval (e.g., 0.90 for a 90% confidence interval).

4. AWE (Average Weighted Error):
PICP and PINAW combined into an evaluation metric that is weighted.

$$\text{AWE} = \left|\frac{1}{n}\sum_{i=1}^{n}(UB_i - LB_i) - (\max(y) - \min(y))\right| \tag{3.4}$$

Eq. (3.1) represents the PICP evaluation metric, Eq. (3.2) represents the PINAW evaluation metric, Eq. (3.3) represents the ACE evaluation metric and finally Eq. (3.4) represents the AWE evaluation metric. These four metrics is used throughout this thesis to compare and evaluate perfromance of the different methods.

- **Output Generation** For every forecasting technique used, the resulting predicted intervals, performance metrics and mean results are systematically arranged in CSV file format for further evaluation and reproducibility. Visual verification is executed by generating graphs displaying the prediction intervals at different confidence levels (0.9, 0.8, 0.7, 0.6) with the actual values plotted alongside the corresponding interval boundaries. These visual graphs play a critical role in ascertaining the coverage and validity of the intervals. In addition, comparative plots are generated for every deep learning model i.e., LSTM, GRU, CNN and BiLSTM such that predictive accuracy as well as uncertainty estimation can be visually compared for the different probabilistic forecasting techniques. (These graphs are present in Figures 3.2 to 3.7). The Tables 3.2 to 3.11 displays the performances of all the methods used across the five different datasets.

Probabilistic Forecasting Using Traditional LUBE Method

This algorithm describes the Traditional LUBE method that trains deep learning networks (LSTM, CNN, GRU, BiLSTM) directly through a custom loss function that is given below in Eq. (3.7) consisting of lower and upper pinball loss as in Eq. (3.5) and Eq. (3.6) and a coverage penalty to produce interval boundaries independent of distributional assumptions. A few equations essential to compute Lube Loss is given below:

$$\text{Loss}_{\text{lower}} = \frac{1}{n} \sum \max(0, LB_i - y_i) \cdot q \tag{3.5}$$

$$\text{Loss}_{\text{upper}} = \frac{1}{n} \sum \max(0, y_i - UB_i) \cdot (1 - q) \tag{3.6}$$

$$\text{Loss}_{\text{LUBE}} = \text{Loss}_{\text{lower}} + \text{Loss}_{\text{upper}} + \lambda \max(0.9 - \text{PICP}, 0)^2 \tag{3.7}$$

Where:

- y_i: Actual (true) value of the i-th sample
- LB_i: Predicted lower bound of the prediction interval for the i-th sample
- UB_i: Predicted upper bound of the prediction interval for the i-th sample
- n: Total number of samples
- q: Quantile level (e.g., 0.1 for 90% prediction interval)
- λ: Regularization parameter controlling the penalty for low coverage in the LUBE loss function

Algorithm 1: Traditional LUBE Method.

Input: Time series dataset D
Output: Predicted intervals $[LB, UB]$, PICP, PINAW, ACE, AWE

```
Step 1: Data Preprocessing
Normalize D using MinMaxScaler
Generate input-output pairs with window size w
Split into train, validation, and test sets
Step 2: Define LUBE Loss
foreach c ∈ {0.9, 0.8, 0.7, 0.6} do
    q = 1 − c
    Compute LB, UB
    Compute Loss_lower (as in Eq. (3.5)) and Loss_upper (as in Eq. (3.6))
    Compute PICP = (as in Eq. (3.1))
    Compute Loss_LUBE (as in Eq. (3.7))
Step 3: Model Training foreach M ∈ {LSTM, CNN, GRU, BiLSTM} do
    Define model architecture
    Compile with LUBE loss
    Train on (X_train, y_train) for e epochs
    Validate on (X_val, y_val)
    Save LB, UB for test data
Step 4: Evaluation Metrics Compute the PINAW (as in Eq. (3.2)), ACE (as in Eq. (3.3)) and
    AWE (as in Eq. (3.4)) of the computed prediction intervals.
Step 5: Aggregate Results
Compute mean of metrics for all models
```

The Traditional LUBE (Lower Upper Bound Estimation) Method Algorithm is designed to generate reliable prediction intervals for time series forecasting using deep learning models. The process begins with data pre-processing, where the input time series is normalized using MinMaxScaler and transformed into input-output pairs based on a sliding window of size w, followed by splitting the dataset into training, validation, and test sets. A custom LUBE loss function is defined for different confidence levels $c \in \{0.9, 0.8, 0.7, 0.6\}$, where the loss penalizes predictions falling outside the predicted lower (LB) and upper (UB) bounds and encourages higher coverage through a PICP-based regularization term. For each selected deep learning model (LSTM, CNN, GRU, BiLSTM), the model is compiled using the LUBE loss and trained on the preprocessed data. The predicted bounds on the test set are evaluated using four key probabilistic metrics: Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Average Width (PINAW), Average Coverage Error (ACE), and Average Width Error (AWE). Finally, the algorithm aggregates the results across all models, computing the mean and standard deviation of each evaluation metric, thereby offering a robust assessment of the interval prediction performance.

Probabilistic Forecasting using Advance LUBE Method

This algorithm describes The Advanced LUBE method that enhances this Traditional LUBE method by introducing an extra regularization term controlled by hyperparameter μ which applies penalties on wide intervals to improve sharpness.

The Advanced LUBE Loss Function is given below in Eq. (3.8).

$$\text{Loss}_{\text{LUBE}} = \text{Loss}_{\text{upper}} + \lambda \max(0.9 - \text{PICP}, 0)^2 + \mu \cdot \frac{1}{n} \sum |UB_i - LB_i| \quad (3.8)$$

Where:

- $\text{Loss}_{\text{upper}}$: Penalty for true values exceeding the upper bound of the prediction interval. (Given in Eq. (3.6))
- UB_i: Predicted upper bound for the i-th sample
- LB_i: Predicted lower bound for the i-th sample

- n: Total number of samples
- λ: Regularization parameter penalizing insufficient coverage
- μ: Regularization parameter controlling the width of the prediction interval
- $\max(0.9 - \text{PICP}, 0)^2$: Coverage penalty term encouraging the model to achieve at least 90% PICP
- $\sum |UB_i - LB_i|$: Total width of the prediction intervals across all samples

Algorithm 2: Advanced LUBE Method.

```
Input: Time series dataset D
Output: Predicted intervals [LB, UB], PICP, PINAW, ACE, AWE
Step 1: Data Preprocessing
Normalize D using MinMaxScaler
Generate input-output pairs with window size w
Split into train, validation, and test sets
Step 2: Define Advanced LUBE Loss
foreach c ∈ {0.9, 0.8, 0.7, 0.6} do
    q = 1 − c
    Compute LB, UB
    Compute Loss_lower (as in Eq. (3.5)) and Loss_upper (as in Eq. (3.6))
    Compute PICP and PINAW (as in Eq. (3.1) and (3.2) respectively)
    Compute Loss_LUBE (as in Eq. (3.8))
Step 3: Model Training
foreach M ∈ {LSTM, CNN, GRU, BiLSTM} do
    Define model architecture
    Compile with Advanced LUBE loss
    Train on (X_train, y_train) for e epochs
    Validate on (X_val, y_val)
    Save LB, UB for test data
Step 4: Evaluation Metrics
Compute ACE (as in Eq. (3.3)) and AWE (as in Eq. (3.4))
Step 5: Aggregate Results
Compute mean of metrics for all models and confidence levels
```

The Advanced LUBE Method Algorithm enhances the traditional LUBE framework by incorporating interval width regularization into the loss function, thereby promoting both accuracy and tightness of prediction intervals. The process begins with data pre-processing where the time series dataset is normalized using MinMaxScaler, transformed into input-output pairs using a sliding window of size w, and divided into training, validation, and test subsets. The advanced LUBE loss function is defined for multiple confidence levels $c \in \{0.9, 0.8, 0.7, 0.6\}$, with penalties for predictions outside the bounds and additional terms for poor coverage (via PICP) and overly wide intervals (via PINAW). Specifically, the loss includes a regularization term proportional to the average width of the intervals, controlled by the hyperparameter μ. For each deep learning model (LSTM, CNN, GRU, BiLSTM), the model is trained using this advanced loss formulation. The predicted intervals are evaluated using standard probabilistic metrics including PICP, PINAW, Average Coverage Error (ACE), and Average Width Error (AWE). Finally, the results across all models and confidence levels are aggregated to compute the mean and standard deviation of each metric, offering a robust and comprehensive assessment of the method's performance.

Probabilistic Forecasting using QR-based Method

This algorithm describes the Quantile Regression based algorithm that uses quantile loss functions to estimate lower and upper quantiles, summing pinball losses and under-coverage penalties and hence providing statistically well-motivated alternatives to LUBE methods. The following equations are needed to construct the QR Loss Function. Eq. (3.9) shows the Lower quantile threshold while Eq. (3.10) shows the Upper Quantile threshold. Eq. (3.11) and (3.12) calculates the Lower and Upper losses, i.e Penalty for predicted lower or upper bounds exceeding or under flowing true values. Finally, Eq. (3.13) shows the QR Loss function.

$$q_{\text{lower}} = \frac{1-c}{2} \tag{3.9}$$

$$q_{\text{upper}} = 1 - q_{\text{lower}} \tag{3.10}$$

$$\text{Loss}_{\text{lower}} = \frac{1}{n} \sum \max(0, LB_i - y_i) \cdot q_{\text{lower}} \tag{3.11}$$

$$\text{Loss}_{\text{upper}} = \frac{1}{n} \sum \max(0, y_i - UB_i) \cdot q_{\text{upper}} \tag{3.12}$$

$$\text{Loss}_{\text{QR}} = \text{Loss}_{\text{lower}} + \text{Loss}_{\text{upper}} + \lambda \cdot \max(0.9 - \text{PICP}, 0)^2 + \mu \cdot \frac{1}{n} \sum |UB_i - LB_i| \tag{3.13}$$

Where:

- c – Confidence level (e.g: 0.9, 0.8, 0.7 and 0.6)
- q_{lower} – Lower quantile threshold.
- q_{upper} – Upper quantile threshold.
- LB_i – Predicted lower bound of the i-th interval
- UB_i – Predicted upper bound of the i-th interval
- y_i – Actual target value at index i
- n – Total number of samples
- $\text{Loss}_{\text{lower}}$ – Penalty for predicted lower bounds exceeding true values.
- $\text{Loss}_{\text{upper}}$ – Penalty for predicted upper bounds below true values.
- λ – Penalty coefficient for inadequate coverage
- μ – Penalty coefficient for interval width
- PICP – Prediction Interval Coverage Probability

– Loss_{QR} – Total quantile regression-based loss function.

Algorithm 3: Quantile Regression (QR)-based Method.

Input: Time series dataset D
Output: Predicted intervals $[LB, UB]$, PICP, PINAW, ACE, AWE

```
Step 1: Data Preprocessing
Normalize D using MinMaxScaler
Generate input-output pairs with window size w
Split into train, validation, and test sets
Step 2: Define QR Loss Function
foreach c ∈ {0.9, 0.8, 0.7, 0.6} do
    Compute q_lower (as in Eq. (3.9)) and q_upper (as in Eq. (3.10))
    Compute LB, UB
    Compute Loss_lower (as in Eq. (3.11)) and Loss_upper (as in Eq. (3.12))
    Compute PICP (as in Eq.(3.1)) and PINAW (as in Eq. (3.2))
    Compute Loss_QR (as in Eq. (3.13))
Step 3: Model Training
foreach M ∈ {LSTM, CNN, GRU, BiLSTM} do
    Define model architecture
    Compile with QR-based loss
    Train on (X_train, y_train) for e epochs
    Validate on (X_val, y_val)
    Save LB, UB for test data
Step 4: Evaluation Metrics
Compute ACE (as in (3.3) and AWE (as in Eq. (3.4))
Step 5: Aggregate Results
Compute mean of metrics for all models and confidence levels
```

The Quantile Regression (QR)-based Method Algorithm applies quantile regression principles to construct asymmetric prediction intervals for time series forecasting. Initially, the time series dataset is normalized using MinMaxScaler, and transformed into supervised input-output pairs using a sliding window of size w, followed by splitting into training, validation, and testing subsets. For each confidence level $c \in \{0.9, 0.8, 0.7, 0.6\}$, the algorithm calculates the lower and upper quantiles as $q_{\text{lower}} = \frac{1-c}{2}$ and $q_{\text{upper}} = 1 - q_{\text{lower}}$, which define the predicted interval bounds $[LB, UB]$. The QR loss function combines penalties for underestimation and overestimation, weighted by their respective quantiles. Additional terms for improving interval quality include a PICP regularization component and a width regularization term, scaled by hyperparameters λ and μ. Deep learning models (LSTM, CNN, GRU, BiLSTM) are then trained using this composite loss. The performance of each model is assessed using PICP, PINAW, Average Coverage Error (ACE), and Average Width Error (AWE). Finally, results across all models and confidence levels are aggregated by computing the mean and standard deviation of each metric, ensuring a comprehensive evaluation of the prediction interval reliability and efficiency.

Probabilistic Forecasting using Bootstrap-based Method

This algorithm describes the Bootstrap-Based method that estimates uncertainty by training models over multiple resampled datasets and estimating empirical prediction intervals from the predictions ensemble, and this provides it with robustness without assuming a residual distribution.

Algorithm 4: Bootstrap-based Method.

Input: Time series dataset D

Output: Predicted intervals $[LB, UB]$, PICP, PINAW, ACE, AWE

```
Step 1: Data Preprocessing
Normalize D using MinMaxScaler
Generate input-output pairs with window size w
Split into train, validation, and test sets
Step 2: Bootstrap Prediction Intervals
foreach c ∈ {0.9, 0.8, 0.7, 0.6} do
    Generate B bootstrap samples from training data
    foreach bootstrap sample do
        Train model M and predict on test set
    foreach test sample y^i_test do
        Compute LB_i and UB_i as empirical quantiles at q_lower and q_upper as in Eq. (3.9) and
          (3.10)
Step 3: Evaluation Metrics Compute the PICP (as in Eq. (3.1)), PINAW (as in Eq. (3.2)),
  ACE (as in Eq. (3.3)) and AWE (as in Eq. (3.4)) of the computed prediction intervals.
Step 4: Aggregate Results
Compute mean of metrics for all models and confidence levels
```

The Bootstrap-Based Method Algorithm leverages the statistical resampling technique of bootstrapping to generate robust prediction intervals for time series forecasting. The procedure begins with data pre-processing, where the dataset is normalized using MinMaxScaler, converted into input-output pairs using a sliding window of size w, and partitioned into training, validation, and testing subsets. For each confidence level $c \in \{0.9, 0.8, 0.7, 0.6\}$, B bootstrap samples are generated from the training data. A deep learning model M (e.g., LSTM, CNN, GRU, BiLSTM) is trained on each sample and used to generate predictions on the test set. For each test instance, the lower and upper bounds of the prediction interval (LB_i and UB_i) are computed as empirical quantiles corresponding to $q_{\text{lower}} = \frac{1-c}{2}$ and $q_{\text{upper}} = 1 - q_{\text{lower}}$ from the bootstrap distribution. The intervals are evaluated using probabilistic forecasting metrics: Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Average Width (PINAW), Average Coverage Error (ACE), and Average Width Error (AWE). The final step aggregates the performance metrics across all models and confidence levels by computing their mean and standard deviation, thereby providing a comprehensive assessment of interval reliability and sharpness.

Probabilistic Forecasting using Gaussian Distribution based Method

This algorithm uses the target variable as normally distributed and computes symmetric intervals around the mean using standard normal distribution z-scores, providing it with ease of implementation but lack of flexibility in dealing with non-Gaussian datasets. Eq. (3.14) provides the Z-score while Eq. (3.15) and Eq. (3.16) provides the Lower and Upper bounds of the prediction interval.

$$z_c = \Phi^{-1}\left(\frac{1+c}{2}\right) \tag{3.14}$$

$$LB = \mu - z_c \cdot \sigma \tag{3.15}$$

$$UB = \mu + z_c \cdot \sigma \tag{3.16}$$

Where:

- c: Confidence level (e.g., 0.9 for 90% confidence)
- Φ^{-1}: Inverse cumulative distribution function (quantile function) of the standard normal distribution.
- z_c: Z-score corresponding to the confidence level c.
- μ: Mean (expected value) of the distribution.
- σ: Standard deviation of the distribution.
- LB: Lower bound of the prediction interval.
- UB: Upper bound of the prediction interval.

Algorithm 5: Gaussian Distribution based Method.

Input: Time series dataset D with target variable y

Output: Predicted intervals $[LB, UB]$, PICP, PINAW, ACE, AWE

```
1  Step 1: Data Preprocessing
2  Remove missing values to obtain cleaned target y
3  Step 2: Define Confidence Levels
4  C = {0.9, 0.8, 0.7, 0.6}
5  Step 3: Estimate Distribution Parameters
6  Compute mean μ and standard deviation σ of y
7  Step 4: Generate Prediction Intervals
8  foreach c ∈ C do
9      Compute z_c (as in Eq. (3.14)) Compute LB as in Eq. (3.15) and UB as in Eq. (3.16)
10 Step 5: Evaluation Metrics Compute the PICP (as in Eq. (3.1)), PINAW (as in Eq. (3.2)),
     ACE (as in Eq. (3.3)) and AWE (as in Eq. (3.4)) of the computed prediction intervals.
```

The Gaussian Distribution-Based Method Algorithm estimates prediction intervals under the assumption that the target variable follows a normal distribution. The algorithm begins by pre-processing the time series data D to remove missing values, producing a clean target variable y. It then defines a set of confidence levels $C = \{0.9, 0.8, 0.7, 0.6\}$ for interval estimation. For each confidence level $c \in C$, the algorithm calculates the corresponding z-score using the inverse cumulative distribution function Φ^{-1}. Using the mean μ and standard deviation σ of the target variable, the lower and upper bounds of the prediction interval are computed as $LB = \mu - z_c \cdot \sigma$ and $UB = \mu + z_c \cdot \sigma$, respectively. The intervals are then evaluated using standard probabilistic metrics: Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Average Width (PINAW), Average Coverage Error (ACE), and Average Width Error (AWE). Finally, the method aggregates and stores the computed intervals and corresponding metric values for each confidence level, providing a baseline assessment of uncertainty using parametric Gaussian assumptions.

Probabilistic Forecasting Weibull Distribution based Method

This algorithm describes the Weibull Distribution method that employs a fitted Weibull distribution by maximum likelihood to model the target, facilitating interval estimation with the percent point function and showing better compliance with skewed or heavy-tailed data sets. Eq. (3.17) shows the Tail Probability. Eq. (3.18) and Eq. (3.19) shows the formulae for UB and LB calculation of Prediction Intervals.

$$\alpha = \frac{1-c}{2} \tag{3.17}$$

$$LB = \text{PPF}(\alpha, \kappa, \lambda) \tag{3.18}$$

$$UB = \text{PPF}(1-\alpha, \kappa, \lambda) \tag{3.19}$$

Where:

- c — Confidence level (e.g., 0.6, 0.7, 0.8, or 0.9) used to construct the prediction interval.
- α — Tail probability, representing the area in each tail of the distribution outside the prediction interval.

- PPF — Percent-Point Function (inverse of the CDF), used to obtain the value (quantile) corresponding to a given cumulative probability from the Weibull distribution.
- κ — Shape parameter of the Weibull distribution.
- λ — Scale parameter of the Weibull distribution.
- LB — Lower bound of the prediction interval, derived from the α quantile of the Weibull distribution.
- UB — Upper bound of the prediction interval, derived from the $(1-\alpha)$ quantile of the Weibull distribution.

Algorithm 6: Weibull Distribution based Method.

Input: Time series dataset D with target variable y
Output: Predicted intervals $[LB, UB]$, PICP, PINAW, ACE, AWE

Step 1: Data Preprocessing
Remove missing values to obtain cleaned target y
Step 2: Define Confidence Levels
$C = \{0.9, 0.8, 0.7, 0.6\}$
Step 3: Estimate Weibull Parameters
Fit Weibull distribution to y using Maximum Likelihood Estimation (MLE)
Obtain shape κ, scale λ, and location θ (fixed to 0)
Step 4: Generate Prediction Intervals
foreach $c \in C$ **do**
    Compute α (as in Eq. (3.17))
    Compute LB (as in Eq. (3.18)) and UB (as in Eq. (3.19))
Step 5: Evaluation Metrics Compute the PICP (as in Eq. (3.1)), PINAW (as in Eq. (3.2)), ACE (as in Eq. (3.3)) and AWE (as in Eq. (3.4)) of the computed prediction intervals.

The Weibull Distribution-Based Method Algorithm models uncertainty in time series forecasting by fitting a Weibull distribution to the target variable. Initially, the dataset D undergoes pre-processing to remove missing values, yielding a clean target y. The method then defines a set of confidence levels $C = \{0.9, 0.8, 0.7, 0.6\}$ for probabilistic prediction. Using Maximum Likelihood Estimation (MLE), the Weibull distribution is fitted to y to estimate its shape parameter κ, scale parameter λ, and a fixed location parameter $\theta = 0$. For each confidence level $c \in C$, the method computes the lower and upper prediction bounds as the α and $1 - \alpha$ percentiles from the Weibull percent point function (PPF). These prediction intervals are assessed using four standard metrics: Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Average Width (PINAW), Average Coverage Error (ACE) and Average Width Error (AWE). Finally, the algorithm stores the intervals and corresponding metric values across all defined confidence levels, offering a flexible, non-Gaussian parametric approach to interval forecasting.

All the six algorithms start with typical data preprocessing tasks, including normalization using MinMaxScaler and time series data transformation to input-output pairs using a sliding window approach. Each algorithm is set up to produce prediction intervals for different confidence levels (90%, 80%, 70%, 60%) and is compared to measures like Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Average Width (PINAW), Absolute Coverage Error (ACE) and Average Width Error (AWE), and their results are logged and plotted to compare against.
Overall, the collection of algorithms provides a diverse array of methods for probabilistic time series forecasting, balancing model-driven learning with statistical as well as distributional methods.

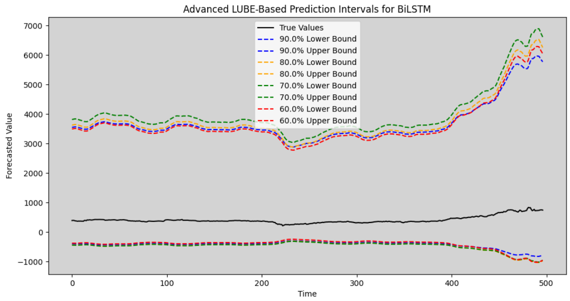
3.3 Results and Discussions

This section displays the evaluation of the six different parametric and non-parametric methods across five datasets. The performance of each method is assessed using metrics PICP, PINAW, ACE and AWE. The results are visualized through prediction interval plots and summarized in tables.

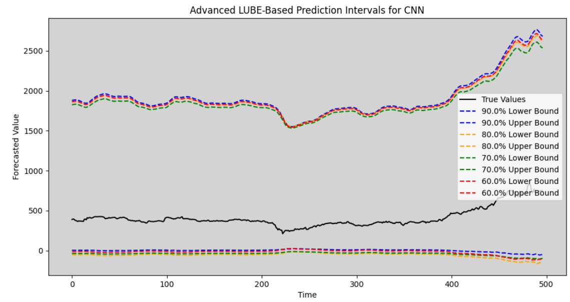
Figure 3.2 shows Prediction Intervals for Adani Ports dataset obtained using Advanced LUBE and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models. Figure 3.3 shows Prediction Intervals for Asian Paints dataset obtained using Traditional LUBE and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models. Figure 3.4 shows Prediction Intervals for Axis Bank dataset obtained using QR-based method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models. Figure 3.5 shows Prediction Intervals for Electricity Consumption dataset obtained using Bootstrap-based method and (a) BiLSTM, (b)

CNN, (c) GRU, (d) LSTM Models. Figures 3.6 and 3.7 shows Prediction Intervals for (a) Adani Ports, (b) Asian Paints, (c) Axis Bank, (d) Electricity Consumption Load, (e) Web Traffic datasets obtained using Gaussian Distribution and Weibull Distribution based methods respectively.

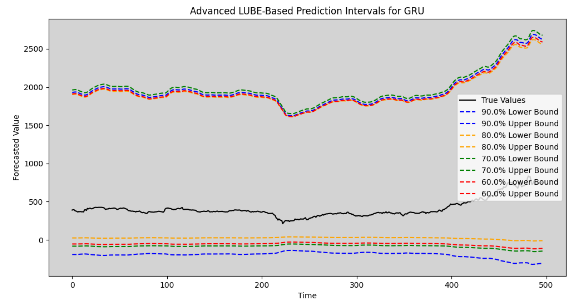
Tables 3.2, 3.3, 3.4, 3.5 and 3.6 shows the comparative performance of Traditional LUBE, Advanced LUBE, QR-based and Bootstrap-based methods on metrics PICP, PINAW, ACE and AWE across multiple confidence levels (90%, 80%, 70% and 60%) using four DL Models BiLSTM, CNN, GRU and LSTM obtained on Adani Ports, Asian Paints, Axis Bank, Electricity Consumption and Web Traffic datasets respectively whereas Tables 3.7, 3.8, 3.9, 3.10 and 3.11 shows the performance of Gaussian Distribution and Weibull Distribution on metrics PICP, PINAW, ACE and AWE across multiple confidence levels (90%, 80%, 70% and 60%) obtained on Adani Ports, Asian Paints, Axis Bank, Electricity Consumption and Web Traffic datasets respectively.



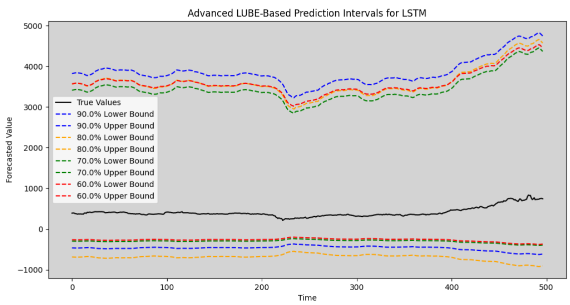
(a) BiLSTM.



(b) CNN.



(c) GRU.



(d) LSTM.

Figure 3.2: Prediction Intervals for Adani Ports dataset obtained using Advanced LUBE method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.

Probabilistic forecasting methodologies have been tested on five disparate datasets: Adani Ports, Asian Paints, Axis Bank, Electricity Consumption Load, and Web Traffic. The six methodologies like Traditional LUBE, Advanced LUBE, QR-based, Bootstrap-based, Gaussian Distribution, and Weibull Distribution were used in all cases of these analyses. The performance was evaluated using the four critical metrics, which included Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Average Width (PINAW), Average Coverage Error (ACE), and Average Width Error (AWE). The following is a more detailed summary of the results and insights learned from the analysis of all five datasets:

3.3.1 Performance Of Each Method Across Datasets

The Traditional LUBE approach consistently achieved near-optimal PICP across all confidence levels and datasets, demonstrating reliable interval coverage. However, it generally produced wider prediction intervals, reflected in higher PINAW values—for instance, a PINAW of 10.01 at the 90% confidence level with CNN DL model for the Adani Ports dataset 3.2, indicating a highly conservative prediction. Similar trends were observed in the Asian Paints dataset, where the LSTM model showed a PINAW of 5.87 at 90% confidence level. 3.3 Furthermore, across all datasets, the Traditional LUBE method exhibited higher AWE values compared to other methods, pointing to less efficient interval calibration. This trade-off suggests that while Traditional LUBE is dependable, it tends to produce overly cautious intervals, particularly in models like BiLSTM. In contrast, the Advanced LUBE method demonstrated superior performance across all datasets by generating narrower yet highly covered intervals. For example, in the Electricity Consumption dataset 3.5, the GRU model achieved a PICP of 100% with a significantly narrower PINAW of 1.35 at 90% confidence level when compared to Traditional LUBE's 2.25. Similarly, in the Web Traffic dataset 3.6, CNN attained a PINAW of 1.69 with a PICP of 99.19% at 90% confidence level. Additionally, the Advanced LUBE method consistently yielded lower ACE and AWE values, confirming its capability to deliver more dependable and efficient prediction intervals. The QR-based method, while producing the narrowest intervals among non-parametric approaches, often suffered from under-coverage. For instance, in the Axis Bank dataset at the 90% confidence level, the BiLSTM model reached 96.65%

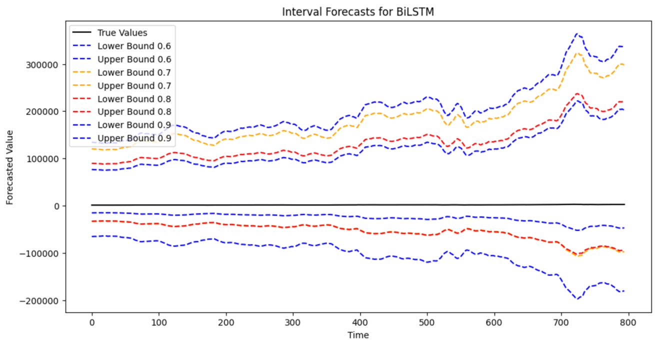
PICP with a very low PINAW of 0.15 3.4. Thus, despite its precision, the QR-based method falls short in applications requiring high reliability. The Bootstrap-based method held an intermediate location on interval coverage and width, with competitive PINAW and PICP values but not entirely on par with consistency of LUBE or Advanced LUBE. On the Adani Ports dataset, for example, the GRU model had a PICP of 68.43% supported by a PINAW of 0.28 at 70% confidence level which is a desirable output, while on the Electricity Consumption dataset, the GRU model held a PICP of only 28.20% supported by a PINAW of 0.10 at 70% confidence level 3.5 which is not at all desirable and shows major undercoverage. The Gaussian distribution method consistently produced the most narrow intervals but often failed to achieve the targeted PICP thresholds. On the Asian Paints dataset, for example, it had a PINAW of 0.42 at confidence level 70% but a PICP of 83.58%, supported by high ACE value of 13.58 indicating under coverage 3.7. On the other hand, the Weibull distribution showed better PICP performance than Gaussian at 92.90% at 90% confidence on the Web Traffic dataset but at the cost of wider intervals (PINAW = 0.76) 3.11. Even though Weibull had better reliability, its relatively higher AWE values compared to Gaussian indicated lower efficiency and hence was better suited for application where coverage takes precedence over narrow intervals. Advanced LUBE hence proved to be the best overall and reliable methodology with high efficiency and reliability on a range of datasets and models. The Gaussian and Weibull Distribution Tables are shown in Tables 3.7 to 3.11.

3.3.2 Dataset Specific Observations

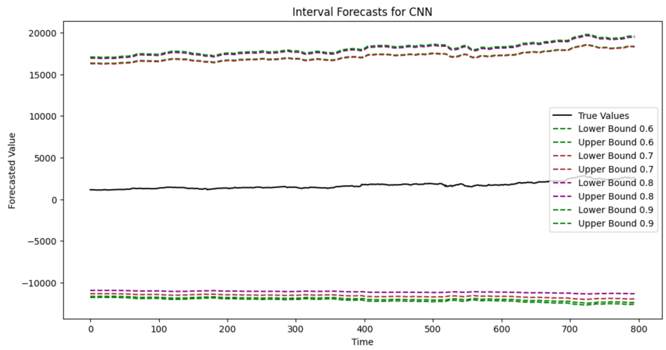
The stock prices data for Adani Ports, Asian Paints and Axis Bank indicated consistent trends, with both the Traditional and Advanced LUBE methods indicating high PICP values. Fig 3.2, Fig. 3.3 Significant differences, however, were observed among the widths of the intervals, with the Advanced LUBE technique showing a tendency to generate more tightened intervals. The QR-based and Bootstrap methods generated more efficient (narrow) intervals;Fig. 3.3 Fig. 3.4 however, these came at the cost of reliability as the coverage levels were lower. The data for Electricity Consumption Load indicated the applicability and efficiency of the Advanced LUBE methodology, which obtained high coverage along with efficient interval lengths, while the QR-based and Bootstrap methods indicated poor performance with very low PICP values. A similar trend was also observed for the Web Traffic dataset, where both the Advanced LUBE and Bootstrap methods indicated relatively higher effectiveness by balancing coverage and interval widths.

3.3.3 Recommendations

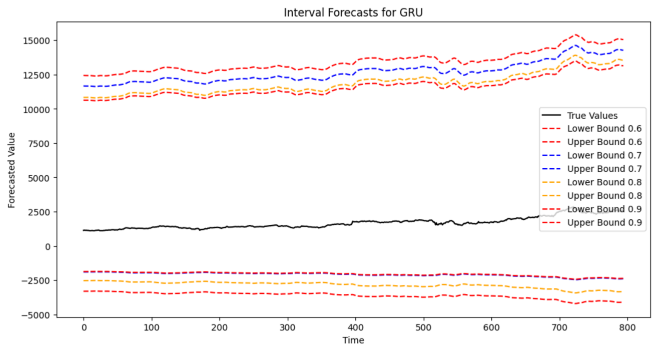
The Advanced LUBE method is best suited for applications that demand high reliability alongside efficient interval widths as it consistently outperformed other approaches across all datasets and evaluation metrics. The Bootstrap-based method serves well in scenarios where a moderate level of reliability is acceptable, offering a balanced trade-off between coverage and interval width. Among the parametric methods, the Gaussian approach yields very narrow and precise intervals but suffers from under-coverage, making it less reliable Fig. 3.6. In contrast, the Weibull distribution delivers better coverage but tends to produce wider intervals, reducing its efficiency for applications requiring tightly bound predictions Fig. 3.7.



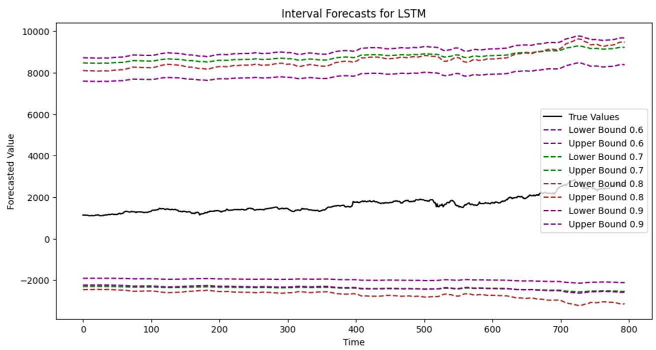
(a) BiLSTM.



(b) CNN.

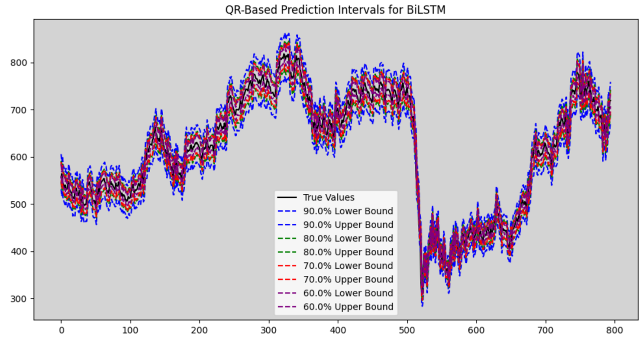


(c) GRU.

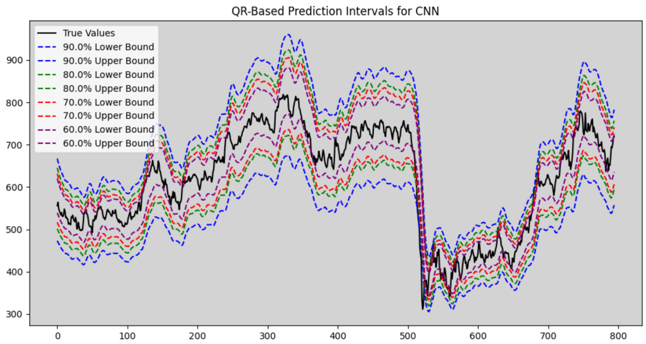


(d) LSTM.

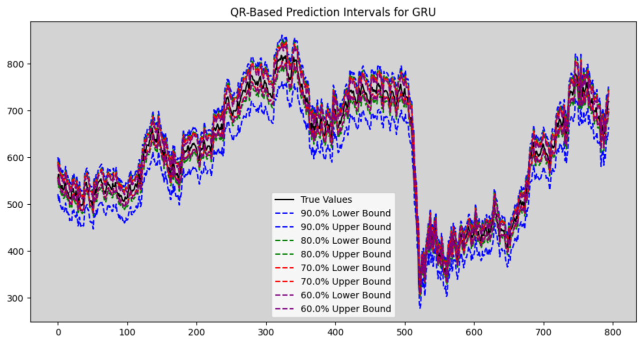
Figure 3.3: Prediction Intervals for Asian Paints dataset obtained using Traditional LUBE method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.



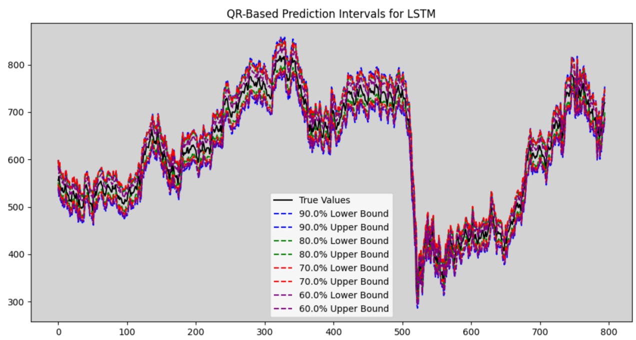
(a) BiLSTM.



(b) CNN.

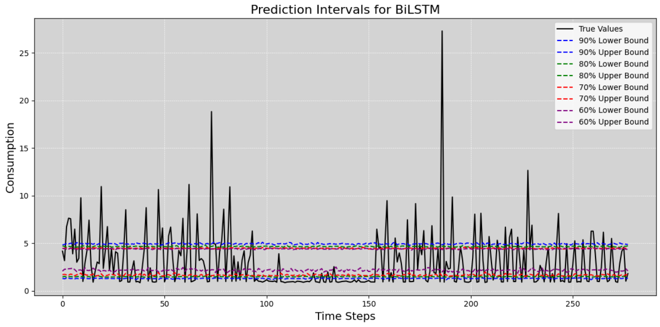


(c) GRU.

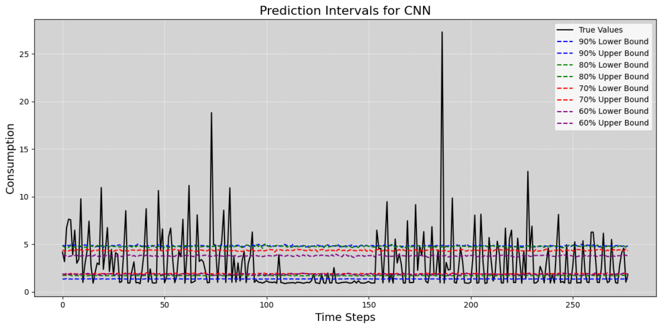


(d) LSTM.

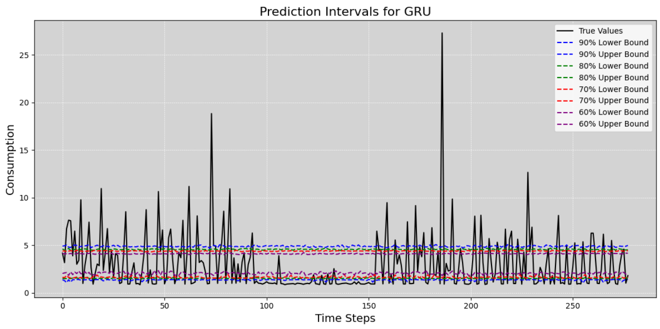
Figure 3.4: Prediction Intervals for Axis Bank dataset obtained using QR-based method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.



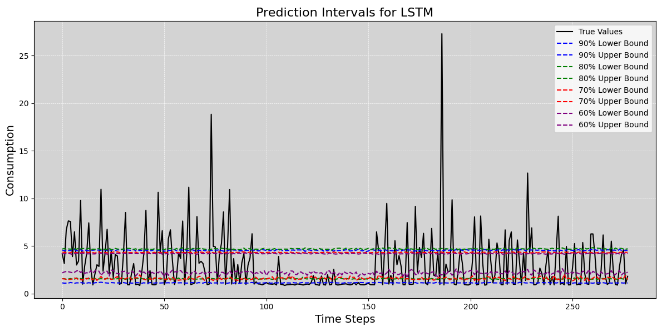
(a) BiLSTM.



(b) CNN.

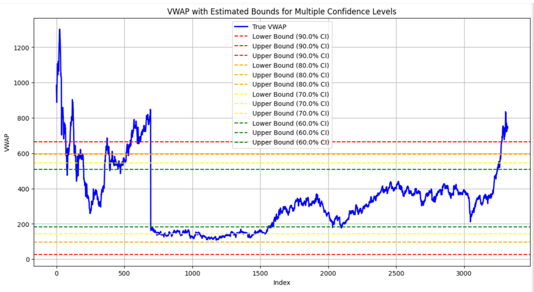


(c) GRU.

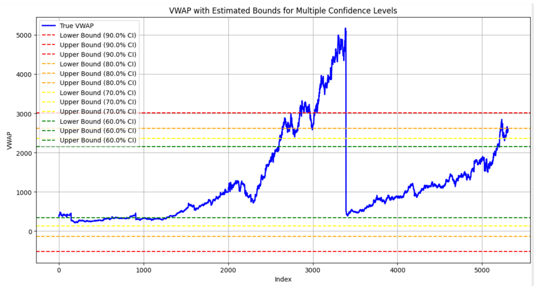


(d) LSTM.

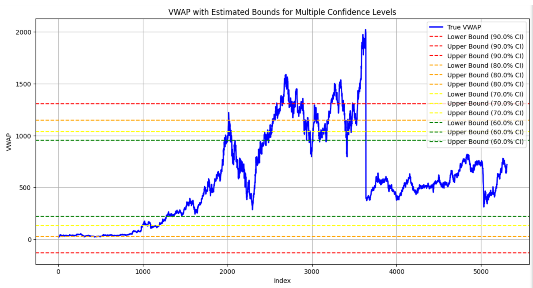
Figure 3.5: Prediction Intervals for Electricity Consumption dataset obtained using Bootstrap-based method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.



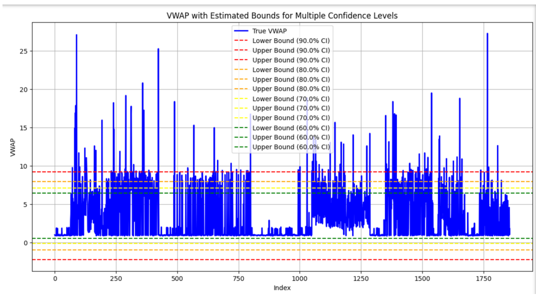
(a) Adani Ports Dataset.



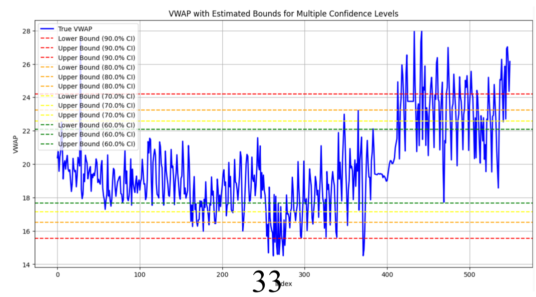
(b) Asian Paints Dataset.



(c) Axis Bank Dataset.

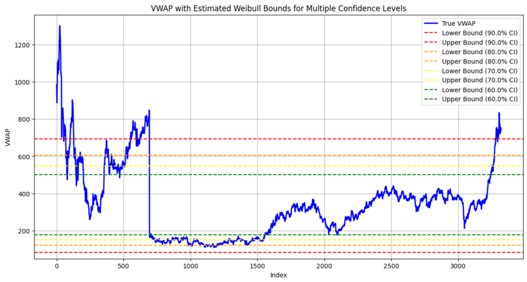


(d) Electricity Consumption Dataset.

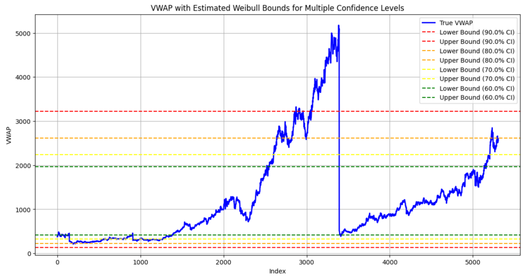


(e) Web Traffic Dataset.

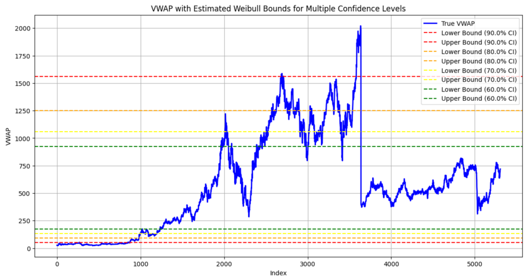
Figure 3.6: Prediction Intervals for (a) Adani Ports, (b) Asian Paints, (c) Axis Bank, (d) Electricity Consumption Load, (e) Web Traffic datasets obtained using Gaussian Distribution method.



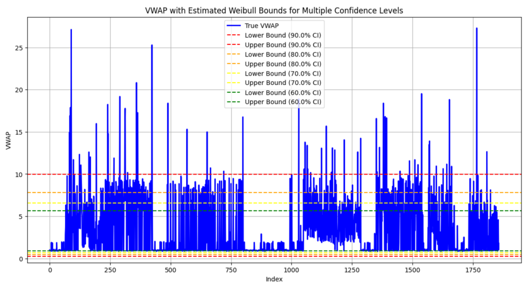
(a) Adani Ports Dataset.



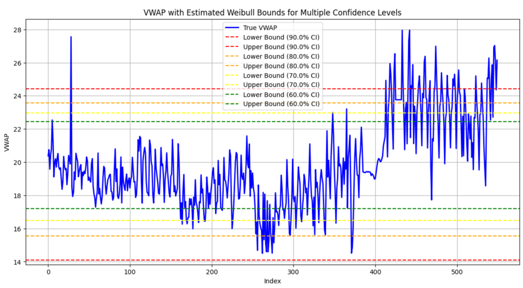
(b) Asian Paints Dataset.



(c) Axis Bank Dataset.



(d) Electricity Consumption Dataset.



(e) Web Traffic Dataset.

Figure 3.7: Prediction Intervals for (a) Adani Ports, (b) Asian Paints, (c) Axis Bank, (d) Electricity Consumption Load, (e) Web Traffic datasets obtained using Weibull Distribution method.

3.3.4 Discussion

The results of this study are quite enlightening with respect to the dynamics that characterize probabilistic forecasting as well as the interactions among various methodologies and datasets. Notable in this respect is that non-parametric techniques had a tendency to vary widely in performance between datasets, indicating their effectiveness in modeling uncertainty without requiring strict assumptions about distributions. For instance, the Advanced LUBE method consistently reached an optimal balance between interval coverage and precision, hence it proved to be reliable. Nevertheless, these methods are computationally intensive and may limit their scalability in a real-time application. In contrast, the parametric approaches presented an interesting duality: they were good at handling structured data, such as electricity consumption, but struggled to capture the complexity of noisy data, such as stock prices and web traffic. This weakness means that these methods are more suitable for domains where the underlying distributions are well defined. The Gaussian distribution was computationally efficient but struggled with coverage accuracy, whereas the Weibull distribution was more versatile across the different datasets.

Interestingly, the model architectures also had a significant role in the forecasting results. CNN and LSTM models performed very well because they are adept at capturing non-linear and temporal dependencies, while GRU and BiLSTM have mixed results, probably because of the complexity of the dataset. These findings bring out the importance of the alignment of model architecture and method selection with the data characteristics.

This research highlights the necessity for hybrid methodologies that combine the accuracy of non-parametric techniques with the effectiveness of parametric frameworks. These strategies have the potential to alleviate the computational limitations associated with non-parametric approaches while preserving adaptability. Furthermore, subsequent investigations might examine adaptive techniques that can fluidly transition between forecasting methods in response to real-time data patterns, thereby facilitating more flexible applications.

Tables 3.2 to 3.6 displays the performance of the four non-parametric methods Traditional LUBE, Advanced LUBE, QR-based and Bootstrap-based methods on the five different datasets namely Adani Ports, Asian Paints, Axis Bank, Electricity

Consumption and Web Traffic. It displays the results across four evaluation metrics and four different confidence levels (90%, 80%, 70% and 60%) for each method.

Tables 3.7 to 3.11 displays the performance of two parametric methods Gaussian and Weibull Distribution based methods across four different confidence levels and four different evaluation metrics on five different datasets namely Adani Ports, Asian Paints, Axis Bank, Electricity Consumption and Web Traffic.

Table 3.2: Comparative Performance of LUBE, Advanced LUBE, QR-based, Bootstrap Based Methods on Adani Ports dataset.

Method Used	**Confidence Level**	**Model**	**Avg PICP**	**Avg PINAW**	**Avg ACE**	**Avg AWE**
Traditional LUBETraditional LUBETraditional LUBEpt< -Traditional LUBEpt>	0.6	BiLSTM	100	31.03	40	18724.39
	0.6	CNN	100	10.07	40	5661.41
	0.6	GRU	100	7.65	40	4148.68
	0.6	LSTM	100	4.94	40	2460.54
	0.7	BiLSTM	100	405.30	30	252109.41
	0.7	CNN	100	9.93	30	5571.24
	0.7	GRU	100	6.63	30	3513.81
	0.7	LSTM	100	4.80	30	2369.84
	0.8	BiLSTM	100	25.62	20	15358.03
	0.8	CNN	100	9.96	20	5592.60
	0.8	GRU	100	7.62	20	4128.26
	0.8	LSTM	100	6.85	20	3652.41
	0.9	BiLSTM	100	624.92	10	389055.75
	0.9	CNN	100	10.01	10	5619.03
	0.9	GRU	100	7.32	10	3942.56
	0.9	LSTM	100	5.10	10	2557.74
Advanced LUBEAdvanced LUBEAdvanced LUBEpt< -Advanced LUBEpt>	0.6	BiLSTM	100	6.54	40	3453.78
	0.6	CNN	100	3.08	40	1297.77
	0.6	GRU	100	3.21	40	1376.41
	0.6	LSTM	100	6.21	40	3247.55
	0.7	BiLSTM	100	7.18	30	3854.64
	0.7	CNN	100	3.05	30	1278.93
	0.7	GRU	100	3.34	30	1459.42
	0.7	LSTM	100	6.02	30	3129.68
	0.8	BiLSTM	100	6.85	20	3649.99
	0.8	CNN	100	3.14	20	1333.73
	0.8	GRU	100	3.06	20	1287.54
	0.8	LSTM	100	6.90	20	3676.17
	0.9	BiLSTM	100	6.61	10	3497.79
	0.9	CNN	100	3.08	10	1297.10
	0.9	GRU	100	3.47	10	1537.40
	0.9	LSTM	100	6.94	10	3702.24
QR-basedQR-basedQR-basedpt< -QR-basedpt>	0.6	BiLSTM	67.87	0.03	12.36	607.25
	0.6	CNN	79.07	0.10	21.86	562.45
	0.6	GRU	58.87	0.02	21.24	608.68
	0.6	LSTM	70.68	0.04	26.97	600.36
	0.7	BiLSTM	60.72	0.03	21.29	606.88
	0.7	CNN	86.96	0.11	16.96	551.85
	0.7	GRU	72.49	0.04	21.80	601.08
	0.7	LSTM	67.38	0.03	21.79	602.06
	0.8	BiLSTM	86.72	0.06	8.42	587.38
	0.8	CNN	86.74	0.13	12.12	539.85
	0.8	GRU	82.27	0.04	9.11	598.12
	0.8	LSTM	80.08	0.06	13.03	589.16
	0.9	BiLSTM	94.55	0.08	5.69	572.07
	0.9	CNN	95.73	0.19	5.75	507.37
	0.9	GRU	89.80	0.07	9.49	577.76
	0.9	LSTM	97.08	0.10	7.30	561.57
Bootstrap-basedBootstrap-basedBootstrap-basedpt< -Bootstrap-basedpt>	0.6	BiLSTM	57.48	0.17	2.55	519.87
	0.6	CNN	56.58	0.16	3.92	521.28
	0.6	GRU	58.95	0.17	2.17	519.90
	0.6	LSTM	57.81	0.17	2.31	519.69
	0.7	BiLSTM	67.97	0.27	2.08	453.09
	0.7	CNN	67.34	0.28	3.03	450.21
	0.7	GRU	68.43	0.28	1.65	451.96
	0.7	LSTM	65.88	0.27	4.17	454.46
	0.8	BiLSTM	76.62	0.45	3.38	340.08
	0.8	CNN	77.04	0.45	3.13	341.60
	0.8	GRU	78.29	0.46	1.71	337.02
	0.8	LSTM	78.27	0.46	1.87	338.39
	0.9	BiLSTM	87.53	0.68	2.47	199.80
	0.9	CNN	88.19	0.69	2.08	190.55
	0.9	GRU	88.05	0.68	1.95	198.01
	0.9	LSTM	87.67	0.68	2.33	199.72

Table 3.3: Comparative Performance of LUBE, Advanced LUBE, QR-based, Bootstrap Based Methods on Asian Paints dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
Traditional LUBETraditional LUBETraditional LUBEpt< -Traditional LUBEpt>	0.6	BiLSTM	100	176.2462	40	306120.048
	0.6	CNN	100	16.7941	40	27589.0975
	0.6	GRU	100	8.7152	40	13476.9307
	0.6	LSTM	100	6.3460	40	9338.4101
	0.7	BiLSTM	100	134.8771	30	233856.5168
	0.7	CNN	100	16.4525	30	26992.4830
	0.7	GRU	100	8.4597	30	13030.5229
	0.7	LSTM	100	6.3955	30	9424.8555
	0.8	BiLSTM	100	107.5393	20	186102.8086
	0.8	CNN	100	16.6753	20	27381.5852
	0.8	GRU	100	8.4415	20	12998.7874
	0.8	LSTM	100	6.4796	20	9571.7360
	0.9	BiLSTM	100	82.2330	10	141897.7734
	0.9	CNN	100	17.2459	10	28378.3252
	0.9	GRU	100	8.8828	10	13769.6257
	0.9	LSTM	100	5.8771	10	8519.3438
Advanced LUBEAdvanced LUBEAdvanced LUBEpt< -Advanced LUBEpt>	0.6	BiLSTM	100	21.38	40	35595.10
	0.6	CNN	100	4.87	40	6758.33
	0.6	GRU	100	6.44	40	9493.89
	0.6	LSTM	100	11.99	40	19192.37
	0.7	BiLSTM	100	19.89	30	32988.48
	0.7	CNN	100	5.12	30	7205.28
	0.7	GRU	100	6.58	30	9740.53
	0.7	LSTM	100	12.73	30	20486.20
	0.8	BiLSTM	100	18.56	20	30669.12
	0.8	CNN	100	5.04	20	7062.18
	0.8	GRU	100	6.77	20	10078.24
	0.8	LSTM	100	12.21	20	19584.56
	0.9	BiLSTM	100	16.27	10	26678.04
	0.9	CNN	100	5.24	10	7401.71
	0.9	GRU	100	6.20	10	9086.57
	0.9	LSTM	100	13.03	10	21021.77
QR-basedQR-basedQR-basedpt< -QR-basedpt>	0.6	BiLSTM	49.45	0.03	22.91	1702.05
	0.6	CNN	81.00	0.13	23.66	1518.25
	0.6	GRU	35.04	0.01	33.01	1720.67
	0.6	LSTM	39.30	0.02	35.72	1709.96
	0.7	BiLSTM	57.99	0.03	17.33	1689.08
	0.7	CNN	95.28	0.20	25.28	1402.63
	0.7	GRU	52.10	0.02	26.34	1706.18
	0.7	LSTM	74.25	0.04	18.81	1670.70
	0.8	BiLSTM	68.73	0.04	20.23	1670.69
	0.8	CNN	94.19	0.21	16.91	1380.19
	0.8	GRU	81.57	0.05	9.77	1661.94
	0.8	LSTM	79.16	0.06	19.18	1638.10
	0.9	BiLSTM	88.19	0.06	10.38	1637.48
	0.9	CNN	98.68	0.32	8.68	1185.35
	0.9	GRU	84.93	0.07	9.97	1632.32
	0.9	LSTM	71.18	0.06	23.80	1645.33
Bootstrap-basedBootstrap-basedBootstrap-basedpt< -Bootstrap-basedpt>	0.6	BiLSTM	59.51	0.37	1.12	1102.74
	0.6	CNN	56.96	0.37	4.45	1098.73
	0.6	GRU	58.70	0.36	1.80	1111.23
	0.6	LSTM	57.26	0.36	3.85	1110.84
	0.7	BiLSTM	68.73	0.49	1.50	884.40
	0.7	CNN	67.72	0.49	3.99	892.03
	0.7	GRU	69.01	0.49	1.35	885.19
	0.7	LSTM	69.07	0.49	1.46	885.54
	0.8	BiLSTM	77.19	0.63	2.81	646.05
	0.8	CNN	79.03	0.65	1.50	611.55
	0.8	GRU	78.91	0.64	1.35	627.15
	0.8	LSTM	79.01	0.64	1.07	621.64
	0.9	BiLSTM	89.08	0.79	1.28	369.51
	0.9	CNN	88.31	0.78	2.19	383.92
	0.9	GRU	88.70	0.79	1.30	373.43
	0.9	LSTM	89.48	0.79	0.97	369.39

Table 3.4: Comparative Performance of LUBE, Advanced LUBE, QR-based, Bootstrap Based Methods on Axis Bank dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
Traditional LUBETraditional LUBETraditional LUBEpt< -Traditional LUBEpt>	0.6	BiLSTM	100	142.23	40	71594.44
	0.6	CNN	100	21.70	40	10492.69
	0.6	GRU	100	13.41	40	6290.57
	0.6	LSTM	100	10.36	40	4746.86
	0.7	BiLSTM	100	84.23	30	42192.68
	0.7	CNN	100	21.98	30	10635.94
	0.7	GRU	100	14.67	30	6932.03
	0.7	LSTM	100	8.15	30	3623.44
	0.8	BiLSTM	100	30683.37	20	15554427.52
	0.8	CNN	100	22.58	20	10939.10
	0.8	GRU	100	15.97	20	7587.76
	0.8	LSTM	100	9.98	20	4553.34
	0.9	BiLSTM	100	110.06	10	55289.93
	0.9	CNN	100	22.11	10	10700.12
	0.9	GRU	100	16.70	10	7957.61
	0.9	LSTM	100	10.35	10	4741.71
Advanced LUBEAdvanced LUBEAdvanced LUBEpt< -Advanced LUBEpt>	0.6	BiLSTM	100	19.02	40	9136.65
	0.6	CNN	100	6.97	40	3027.38
	0.6	GRU	100	7.53	40	3311.35
	0.6	LSTM	100	15.30	40	7248.18
	0.7	BiLSTM	100	17.66	30	8445.44
	0.7	CNN	100	7.11	30	3098.77
	0.7	GRU	100	7.49	30	3287.88
	0.7	LSTM	100	14.99	30	7091.83
	0.8	BiLSTM	100	19.64	20	9449.07
	0.8	CNN	100	6.86	20	2972.82
	0.8	GRU	100	7.77	20	3432.81
	0.8	LSTM	100	15.69	20	7448.36
	0.9	BiLSTM	100	21.82	10	10554.26
	0.9	CNN	100	6.64	10	2861.38
	0.9	GRU	100	7.10	10	3094.89
	0.9	LSTM	100	16.27	10	7743.09
QR-basedQR-basedQR-basedpt< -QR-basedpt>	0.6	BiLSTM	63.97	0.05	16.28	481.88
	0.6	CNN	78.04	0.17	25.41	423.17
	0.6	GRU	74.13	0.06	16.84	476.30
	0.6	LSTM	81.17	0.09	24.06	460.57
	0.7	BiLSTM	82.44	0.08	18.72	466.64
	0.7	CNN	93.64	0.24	23.64	384.16
	0.7	GRU	78.98	0.07	11.27	472.55
	0.7	LSTM	87.99	0.11	20.52	453.45
	0.8	BiLSTM	86.69	0.09	8.15	459.56
	0.8	CNN	96.35	0.30	16.35	355.55
	0.8	GRU	88.24	0.09	8.89	462.92
	0.8	LSTM	84.69	0.09	14.30	461.12
	0.9	BiLSTM	96.65	0.15	6.74	429.74
	0.9	CNN	98.25	0.41	8.25	299.18
	0.9	GRU	95.96	0.17	7.53	423.20
	0.9	LSTM	89.23	0.13	9.45	442.80
Bootstrap-basedBootstrap-basedBootstrap-basedpt< -Bootstrap-basedpt>	0.6	BiLSTM	59.80	0.47	2.19	268.87
	0.6	CNN	61.04	0.47	2.96	267.34
	0.6	GRU	60.68	0.47	1.79	269.28
	0.6	LSTM	59.50	0.47	2.31	269.05
	0.7	BiLSTM	70.53	0.56	1.91	224.65
	0.7	CNN	71.46	0.56	2.96	220.82
	0.7	GRU	72.65	0.56	2.97	224.17
	0.7	LSTM	68.44	0.55	2.84	226.77
	0.8	BiLSTM	79.60	0.63	1.36	188.36
	0.8	CNN	73.03	0.62	7.17	191.75
	0.8	GRU	80.23	0.63	0.91	186.70
	0.8	LSTM	79.46	0.63	1.17	186.19
	0.9	BiLSTM	88.88	0.72	1.12	142.82
	0.9	CNN	87.50	0.72	2.50	140.65
	0.9	GRU	88.87	0.72	1.13	142.78
	0.9	LSTM	85.02	0.71	4.98	148.03

Table 3.5: Comparative Performance of LUBE, Advanced LUBE, QR-based, Bootstrap Based Methods on Electricity Consumption dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
Traditional LUBETraditional LUBETraditional LUBEpt< -Traditional LUBEpt>	0.6	BiLSTM	100.00	3.69	40	71.06
	0.6	CNN	100.00	4.15	40	83.35
	0.6	GRU	100.00	2.44	40	38.01
	0.6	LSTM	100.00	2.24	40	32.70
	0.7	BiLSTM	100.00	3.80	30	74.14
	0.7	CNN	100.00	4.00	30	79.19
	0.7	GRU	100.00	1.70	30	18.59
	0.7	LSTM	100.00	2.68	30	44.36
	0.8	BiLSTM	100.00	5.71	20	124.41
	0.8	CNN	100.00	4.10	20	82.06
	0.8	GRU	100.00	2.81	20	47.74
	0.8	LSTM	100.00	2.75	20	46.31
	0.9	BiLSTM	100.00	4.16	10	83.67
	0.9	CNN	100.00	4.19	10	84.42
	0.9	GRU	100.00	2.25	10	33.16
	0.9	LSTM	100.00	1.97	10	25.68
Advanced LUBEAdvanced LUBEAdvanced LUBEpt< -Advanced LUBEpt>	0.6	BiLSTM	100.00	2.02	40	27.09
	0.6	CNN	99.68	0.99	39.68	0.90
	0.6	GRU	100.00	1.39	40	10.31
	0.6	LSTM	100.00	2.07	40	28.23
	0.7	BiLSTM	100.00	1.82	30	21.70
	0.7	CNN	99.67	0.99	29.67	0.65
	0.7	GRU	100.00	1.48	30	12.79
	0.7	LSTM	100.00	1.98	30	25.95
	0.8	BiLSTM	100.00	1.99	20	26.15
	0.8	CNN	99.67	0.98	19.67	1.08
	0.8	GRU	100.00	1.36	20	9.52
	0.8	LSTM	100.00	2.32	20	35.01
	0.9	BiLSTM	100.00	1.66	10	17.32
	0.9	CNN	99.69	1.01	9.69	0.92
	0.9	GRU	100.00	1.35	10	9.32
	0.9	LSTM	100.00	2.21	10	32.04
QR-basedQR-basedQR-basedpt< -QR-basedpt>	0.6	BiLSTM	46.69	0.1443	13.31	22.63
	0.6	CNN	56.94	0.1667	10.44	22.03
	0.6	GRU	50.04	0.1494	10.70	22.49
	0.6	LSTM	65.47	0.1605	12.68	22.20
	0.7	BiLSTM	70.76	0.1980	4.37	21.21
	0.7	CNN	73.78	0.2004	4.11	21.14
	0.7	GRU	65.61	0.1838	10.45	21.58
	0.7	LSTM	74.86	0.1997	12.18	21.16
	0.8	BiLSTM	78.92	0.2372	6.69	20.17
	0.8	CNN	82.73	0.2484	5.94	19.87
	0.8	GRU	81.98	0.2278	5.22	20.42
	0.8	LSTM	81.51	0.2473	10.86	19.90
	0.9	BiLSTM	88.35	0.2900	4.82	18.77
	0.9	CNN	87.52	0.3027	5.66	18.44
	0.9	GRU	90.65	0.3133	3.58	18.16
	0.9	LSTM	83.96	0.3342	15.01	17.60
Bootstrap-basedBootstrap-basedBootstrap-basedpt< -Bootstrap-basedpt>	0.6	BiLSTM	24.42	0.0846	35.58	24.20
	0.6	CNN	24.42	0.0734	35.58	24.50
	0.6	GRU	19.50	0.0777	40.50	24.39
	0.6	LSTM	21.01	0.0762	38.99	24.43
	0.7	BiLSTM	28.06	0.1064	41.94	23.63
	0.7	CNN	32.23	0.0930	37.77	23.98
	0.7	GRU	28.20	0.1040	41.80	23.69
	0.7	LSTM	33.92	0.1029	36.08	23.72
	0.8	BiLSTM	34.60	0.1198	45.40	23.27
	0.8	CNN	32.70	0.1157	47.30	23.38
	0.8	GRU	31.76	0.1152	48.24	23.39
	0.8	LSTM	37.70	0.1184	42.30	23.31
	0.9	BiLSTM	45.04	0.1376	44.96	22.80
	0.9	CNN	43.88	0.1316	46.12	22.96
	0.9	GRU	41.40	0.1335	48.60	22.91
	0.9	LSTM	52.59	0.1297	37.41	23.01

Table 3.6: Comparative Performance of LUBE, Advanced LUBE, QR-based, Bootstrap Based Methods on Web Traffic dataset.

Method Used	**Confidence Level**	**Model**	**Avg PICP**	**Avg PINAW**	**Avg ACE**	**Avg AWE**
Traditional LUBETraditional LUBETraditional LUBEpt< -Traditional LUBEpt>	0.6	BiLSTM	100	10.5871	40	89.2555
	0.6	CNN	100	5.3765	40	40.7450
	0.6	GRU	100	3.9920	40	27.8559
	0.6	LSTM	100	3.7757	40	25.8415
	0.7	BiLSTM	100	10.0964	30	84.6871
	0.7	CNN	100	5.2022	30	39.1221
	0.7	GRU	100	4.0640	30	28.5260
	0.7	LSTM	100	2.8798	30	17.5013
	0.8	BiLSTM	100	10.1790	20	85.4565
	0.8	CNN	100	5.2952	20	39.9882
	0.8	GRU	100	4.1002	20	28.8626
	0.8	LSTM	100	4.0289	20	28.1994
	0.9	BiLSTM	100	10.4536	10	88.0132
	0.9	CNN	100	5.2567	10	39.6294
	0.9	GRU	100	4.0724	10	28.6041
	0.9	LSTM	100	3.7530	10	25.6306
Advanced LUBEAdvanced LUBEAdvanced LUBEpt< -Advanced LUBEpt>	0.6	BiLSTM	100	5.1715	40	38.8365
	0.6	CNN	99.9628	1.8441	39.9628	7.8588
	0.6	GRU	100	2.9044	40	17.7303
	0.6	LSTM	100	4.5065	40	32.6452
	0.7	BiLSTM	100	4.6309	30	33.8035
	0.7	CNN	98.9218	1.7859	28.9218	7.3169
	0.7	GRU	100	3.1460	30	19.9797
	0.7	LSTM	100	4.1576	30	29.3974
	0.8	BiLSTM	100	5.0116	20	37.3477
	0.8	CNN	99.4170	1.7916	19.4170	7.3702
	0.8	GRU	100	3.0457	20	19.0454
	0.8	LSTM	100	4.0881	20	28.7499
	0.9	BiLSTM	100	4.9196	10	36.4912
	0.9	CNN	99.1939	1.6862	9.1939	6.3889
	0.9	GRU	100	3.1565	10	20.0771
	0.9	LSTM	100	4.2867	10	30.5994
QR-basedQR-basedQR-basedpt< -QR-basedpt>	0.6	BiLSTM	21.4634	0.2639	38.5366	6.8529
	0.6	CNN	15.1220	0.2586	44.8780	6.9025
	0.6	GRU	33.7805	0.3303	26.2195	6.2347
	0.6	LSTM	19.6341	0.2617	40.3659	6.8733
	0.7	BiLSTM	25.8537	0.3031	44.1463	6.4879
	0.7	CNN	18.4146	0.3330	51.5854	6.2101
	0.7	GRU	43.2927	0.3703	26.7073	5.8625
	0.7	LSTM	25.2439	0.2937	44.7561	6.5756
	0.8	BiLSTM	50.2439	0.4557	29.7561	5.0670
	0.8	CNN	24.3902	0.4074	55.6098	5.5174
	0.8	GRU	52.9268	0.4617	27.0732	5.0117
	0.8	LSTM	40.6098	0.4011	39.3902	5.5761
	0.9	BiLSTM	58.7805	0.5750	31.2195	3.9564
	0.9	CNN	19.5122	0.4978	70.4878	4.6753
	0.9	GRU	50.3659	0.4454	39.6341	5.1632
	0.9	LSTM	45.0000	0.4244	45.0000	5.3588
Bootstrap-basedBootstrap-basedBootstrap-basedpt< -Bootstrap-basedpt>	0.6	BiLSTM	19.5122	0.2041	40.4878	7.4098
	0.6	CNN	28.1707	0.2416	31.8293	7.0603
	0.6	GRU	20.1220	0.2004	39.8780	7.4442
	0.6	LSTM	19.1463	0.2147	40.8537	7.3116
	0.7	BiLSTM	20.3659	0.2337	49.6341	7.1339
	0.7	CNN	35.3659	0.2972	34.6341	6.5429
	0.7	GRU	26.8293	0.2754	43.1707	6.7694
	0.7	LSTM	21.8049	0.2436	48.1951	7.0651
	0.8	BiLSTM	38.7073	0.3137	41.2927	5.8068
	0.8	CNN	43.1707	0.3291	26.8293	5.9821
	0.8	GRU	33.7805	0.3092	36.2195	5.9619
	0.8	LSTM	24.3902	0.2824	45.6098	6.1043
	0.9	BiLSTM	45.3659	0.4005	34.6341	4.4732
	0.9	CNN	51.7073	0.3894	18.2927	4.4840
	0.9	GRU	45.2195	0.3735	34.7805	4.5675
	0.9	LSTM	46.4634	0.3952	33.5366	4.5153

Table 3.7: Comparative Performance of Gaussian Distribution and Weibull Distribution based Methods on Adani Ports dataset.

Method Used	**Confidence Level**	**Avg PICP**	**Avg PINAW**	**Avg ACE**	**Avg AWE**
Gaussian DistributionGaussian DistributionGaussian Distributionpt< -Gaussian Distributionpt>	0.6	55.75	0.09	4.25	163.11
	0.7	71.19	0.12	1.19	200.87
	0.8	88.32	0.14	8.32	248.38
	0.9	91.36	0.19	1.36	318.79
Weibull DistributionWeibull DistributionWeibull Distributionpt< -Weibull Distributionpt>	0.6	56.08	0.27	3.92	323.55
	0.7	65.77	0.33	4.23	395.97
	0.8	86.66	0.40	6.66	485.05
	0.9	92.50	0.51	2.50	612.33

Table 3.8: Comparative Performance of Gaussian Distribution and Weibull Distribution based Methods on Asian Paints dataset.

Method Used	Confidence Level	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
Gaussian DistributionGaussian DistributionGaussian Distributionpt< -Gaussian Distributionpt>	0.6	63.38	0.34	3.38	904.70
	0.7	83.58	0.42	13.58	1114.12
	0.8	86.78	0.52	6.78	1377.61
	0.9	90.93	0.66	0.93	1768.14
Weibull DistributionWeibull DistributionWeibull Distributionpt< -Weibull Distributionpt>	0.6	56.26	0.31	3.74	1558.04
	0.7	68.69	0.38	1.31	1923.42
	0.8	86.28	0.48	6.28	2388.31
	0.9	93.35	0.62	3.35	3092.62

Table 3.9: Comparative Performance of Gaussian Distribution and Weibull Distribution based Methods on Axis Bank dataset.

Method Used	Confidence Level	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
Gaussian DistributionGaussian DistributionGaussian Distributionpt< -Gaussian Distributionpt>	0.6	53.13	0.14	6.87	367.43
	0.7	60.25	0.17	9.75	452.47
	0.8	84.41	0.21	4.41	559.48
	0.9	92.82	0.27	2.82	718.10
Weibull DistributionWeibull DistributionWeibull Distributionpt< -Weibull Distributionpt>	0.6	52.90	0.37	7.10	749.62
	0.7	61.23	0.46	8.77	928.27
	0.8	71.70	0.58	8.30	1157.98
	0.9	82.72	0.76	7.28	1511.76

Table 3.10: Comparative Performance of Gaussian Distribution and Weibull Distribution based Methods on Electricity Consumption dataset.

Method Used	Confidence Level	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
Gaussian DistributionGaussian DistributionGaussian Distributionpt< -Gaussian Distributionpt>	0.6	79.60	0.003	19.60	2.94
	0.7	83.36	0.004	13.36	3.62
	0.8	87.41	0.005	7.41	4.48
	0.9	94.30	0.006	4.30	5.74
Weibull DistributionWeibull DistributionWeibull Distributionpt< -Weibull Distributionpt>	0.6	64.10	0.18	4.10	4.71
	0.7	80.30	0.22	10.30	5.86
	0.8	86.54	0.27	6.54	7.36
	0.9	96.77	0.37	6.77	9.73

Table 3.11: Comparative Performance of Gaussian Distribution and Weibull Distribution based Methods on Web Traffic dataset.

Method Used	Confidence Level	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
Gaussian DistributionGaussian DistributionGaussian Distributionpt< -Gaussian Distributionpt>	0.6	62.54	0.008	2.54	2.21
	0.7	71.27	0.009	1.27	2.72
	0.8	78.73	0.01	1.27	3.37
	0.9	89.45	0.15	0.55	4.32
Weibull DistributionWeibull DistributionWeibull Distributionpt< -Weibull Distributionpt>	0.6	70	0.39	10	5.25
	0.7	77.45	0.48	7.45	6.48
	0.8	84.54	0.60	4.54	8.02
	0.9	92.90	0.76	2.90	10.31

3.4 Summary

Using an ensemble of datasets (Stock prices: Adani Ports, Asian Paints, Axis Bank; Electricity Consumption Load and Web Traffic), this paper has examined and compared different probabilistic techniques for time series forecasting, combining parametric and non-parametric approaches. Using a fundamental framework of machine learning models including LSTM, CNN, GRU and BiLSTM. This paper concentrated on creating and contrasting six distinct approaches: Traditional LUBE, Advanced LUBE, QR-based, Bootstrap-based, Gaussian Distribution and Weibull Distribution based methods.

According to the findings, non-parametric techniques—in particular, the Advanced LUBE method is more effective at balancing the PICP and producing compact and trustworthy prediction intervals with a smaller PINAW. Because they could generate precise and effective predictions for a variety of datasets, CNN and LSTM were determined to be the most effective models among them. This is very evident from the results displayed in Tables 3.2 to 3.6.

In datasets where data variability matches its assumptions, parametric techniques such as the Weibull Distribution demonstrated promise by surpassing the Gaussian Distribution in terms of coverage probability while preserving manageable interval lengths. For example the Weibull Distribution fared better than the Gaussian Distribution in the Electricity Consumption dataset, obtaining a greater coverage probability (PICP of 96.77% versus 94.29%) at the 90% confidence level while keeping interval lengths (PINAW of 0.368) that were manageable. This is evident form Table 2.9. This illustrates how Weibull can more effectively handle the variability of the dataset, providing better coverage without appreciably sacrificing interval precision. The results of Weibull and Gaussian Distributions are displayed in Tables 3.7 to 3.11.

Overall, the results show each method's advantages and disadvantages. While parametric approaches offer more straightforward, distribution-based options that work well in certain situations, non-parametric approaches are best suited for applications that demand flexibility and few assumptions. This study adds to the expanding corpus of research in probabilistic forecasting by offering a comparative framework that can direct future investigations and real-world applications in domains including web analytics, energy, and finance.

Chapter 4

LUBE-Weibull Based Hybrid Method For Probabilistic Time Series Forecasting

4.1 Motivation

Forecasting is critical in applications ranging from energy and finance to medicine and climatology, to enable informed decision-making. Traditional forecasting techniques yield single-point forecasts, disregarding uncertainty, which can result in misleading forecasts and wasteful resource usage in high-stakes applications. Probabilistic forecasting circumvents this shortcoming by generating prediction intervals (PIs) that capture uncertainty, enabling risk estimation. The issue remains, however, to find an optimal technique. Parametric techniques (e.g., Gaussian, Weibull) are efficient in computation but could be ineffective in detecting complex data patterns. Non-parametric techniques (e.g., Quantile Regression, Bootstrap) are more general but are computationally demanding.

The Lower Upper Bound Estimation (LUBE) algorithm, a deep learning-based non-parametric algorithm, learns PIs from data but lacks reliability in highly dynamic environments. Weibull distribution modeling, in contrast, models residual errors accurately but lacks adaptability to time-dependent forecast changes.

This chapter presents the LUBE-Weibull Based Hybrid Method, a technique that improves PI reliability through the combination of deep learning-based LUBE prediction and Weibull-based residual correction.

The technique consists of using deep learning models (LSTM, CNN, GRU, BiLSTM) to produce initial prediction intervals, and residual error modeling through a Weibull distribution. The Weibull-based corrections improve the intervals, which improves accuracy and stability at various confidence levels.

Through the combination of the flexibility of deep learning and statistical error modeling, the technique presents stronger and more reliable probabilistic predictions.

4.2 Methodology

4.2.1 Data Preprocessing

The time series datasets are preprocessed to ensure stable training and effective modeling. First, the data is normalized to the range [0,1] using MinMaxScaler. Next, a sliding window approach with a window size of 12 is applied to generate input-output pairs, where the past 12 time steps are used to predict the next step. Finally, the dataset is divided into three subsets: 70% for training, 15% for validation, and 15% for testing, ensuring a balanced and robust evaluation of the model.

4.2.2 Model Selection

The hybrid method employs different deep learning architectures to generate first-stage prediction intervals (PIs). Long Short-Term Memory (LSTM) is well suited to learning long-term dependencies of sequential data, while Convolutional Neural Networks (CNN) are best at detecting local patterns and trends. Gated Recurrent Units (GRU) offer a computationally cheaper option that still retains the ability to learn sequences. Additionally, Bidirectional LSTM (BiLSTM) enhances feature extraction through learning information in both directions. Each model delivers two outputs for each prediction, which represent the lower and upper bounds of the PI and hence enable extensive estimation of uncertainty.

4.2.3 LUBE Loss Function

The LUBE method directly learns interval bounds using a custom loss function. The Advanced LUBE loss function is defined as:

$$\mathcal{L}_{\text{LUBE}} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}\max(0, \hat{y}_i^{\text{lower}} - y_i)\cdot q}_{\text{Lower Loss}} + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\max(0, y_i - \hat{y}_i^{\text{upper}})\cdot(1-q)}_{\text{Upper Loss}} + \lambda\cdot\max(0, \tau - \text{PICP})^2 \tag{4.1}$$

- $\mathcal{L}_{\text{LUBE}}$: Custom LUBE loss function.
- $\hat{y}_i^{\text{lower}}$: Predicted lower bound.
- $\hat{y}_i^{\text{upper}}$: Predicted upper bound.
- y_i : True value.
- q : Quantile parameter (0.05).
- N : Number of samples.
- PICP : Prediction Interval Coverage Probability.
- τ : Desired coverage probability (0.9).
- λ : Scaling factor (5).

The Advanced LUBE loss function has some components that are designed to promote robust and consistent interval estimation. The Lower Loss penalizes those cases where the lower bound estimated is higher than the actual value, and the Upper Loss penalizes those cases where the upper bound estimated is less than the actual value. Additionally, the Prediction Interval Coverage Probability (PICP) enforces that the actual values be inside the estimated bounds with a high probability, thus improving the calibration of the uncertainty estimates.

4.2.4 Weibull-based Residual Correction

Although the LUBE method is a highly effective method of estimating prediction intervals (PIs), it might not be completely capturing residual errors, which undermines its reliability. To overcome this, a Weibull model is used for residual modeling, i.e., absolute prediction errors produced by deep learning models. The process starts by calculating the residuals through the absolute difference between predicted mean interval and observed value. Secondly, Maximum Likelihood Estimation (MLE) is used to fit the Weibull distribution and estimate its shape and scale parameters. Finally, lower and upper LUBE bounds are corrected using corrections obtained from the Weibull, thus ensuring more accurate and reliable prediction intervals.

$$Correction = \lambda \cdot \text{Weibull}^{-1}\left(1 - \frac{1-\alpha}{2}, k, \sigma\right) \tag{4.2}$$

- λ : Scaling factor controlling the impact of Weibull correction.
- Weibull^{-1} : The inverse cumulative distribution function (quantile function) of the Weibull distribution.
- k and σ : Estimated shape and scale parameters of the Weibull distribution respectively.
- α : Confidence level (0.9, 0.8, 0.7, 0.6).

4.2.5 Confidence Levels and Performance Metrics

The hybrid method examines prediction intervals for four confidence levels: 90%, 80%, 70%, and 60%, giving a complete uncertainty estimation evaluation. Performance is measured via several key indicators. Prediction Interval Coverage Probability (PICP) estimates the percentage of actual values in the predicted interval, assessing reliability. Prediction Interval Normalized Average Width (PINAW) estimates interval sharpness as a function of data range, sacrificing precision for coverage. Average Coverage Error (ACE) estimates PICP deviation from the target confidence level, measuring calibration accuracy. Lastly, Average Width Error (AWE) evaluates how far interval width is from expected bounds, ensuring proper uncertainty quantification.

4.2.6 Probabilistic Forecasting using Hybrid LUBE-Weibull based Method

The LUBE-Weibull Hybrid method algorithm provided below (Algorithm 7) demonstrates the application of the Hybrid LUBE–Weibull Method for time series forecasting. It begins with simple pre-processing and trains deep learning models (LSTM, CNN, GRU, BiLSTM) with a specified LUBE loss function to generate initial prediction intervals. Residuals are obtained by subtracting predicted from actual means, and a Weibull distribution is fitted to these residuals by maximum likelihood estimation. Prediction intervals are adjusted by expanding the initial limits based on Weibull distribution corrections for every confidence level. The adjusted intervals are then evaluated using standard metrics (PICP, PINAW, ACE, AWE) and averaged over ten runs.

Algorithm 7: Hybrid LUBE-Weibull Method.

Input: Time series dataset D

Output: Predicted intervals $[LB, UB]$, PICP, PINAW, ACE, AWE

Step 1: Data Preprocessing
Normalize D using MinMaxScaler
Generate input-output pairs with window size w
Split into train, validation, and test sets
Step 2: Define Advanced LUBE Loss
foreach $c \in \{0.9, 0.8, 0.7, 0.6\}$ **do**
    $q = 1 - c$
    Compute LB, UB
    Compute $\text{Loss}_{\text{lower}}$ and $\text{Loss}_{\text{upper}}$
    Compute PICP (as in Eq. (3.1)) and PINAW (as in Eq. (3.2))
    Compute $\text{Loss}_{\text{LUBE}}$ (as in Eq. (4.1))
Step 3: Model Training
foreach $M \in \{$*LSTM, CNN, GRU, BiLSTM*$\}$ **do**
    Define model architecture
    Compile with Advanced LUBE loss
    Train on $(X_{\text{train}}, y_{\text{train}})$ for e epochs
    Validate on $(X_{\text{val}}, y_{\text{val}})$
    Predict LB, UB for test data
Step 4: Weibull Distribution Fitting on Residuals
Compute residuals r
Estimate Weibull parameters $(\hat{k}, \hat{\lambda})$ using MLE
Step 5: Adjust Prediction Intervals Using Weibull Correction
foreach $c \in \{0.9, 0.8, 0.7, 0.6\}$ **do**
    Compute Weibull-based correction factor δ_c (as in Eq. (4.2))
Step 6: Evaluation Metrics
Compute PICP (as in Eq. (3.1)), PINAW (as in Eq. (3.2)), ACE (as in Eq. (3.3)) and AWE (as in Eq. (3.4)) of the computed prediction intervals.
Step 7: Aggregate Results
Compute mean of metrics for all models

The Hybrid LUBE–Weibull Method integrates the strengths of the Lower Upper Bound Estimation (LUBE) approach with probabilistic correction using the Weibull distribution. The algorithm begins by pre-processing the time series dataset and preparing input-output windows. For each confidence level ($c \in \{0.9, 0.8, 0.7, 0.6\}$), the LUBE loss function is defined and used to train deep learning models (LSTM, CNN, GRU, BiLSTM). After training, residuals between actual values and predicted means are computed, and a Weibull distribution is fitted to these residuals using Maximum Likelihood Estimation (MLE). Correction factors based on the Weibull quantiles are then calculated and applied to refine the initial prediction intervals. Finally, the method evaluates the performance using PICP, PINAW, ACE and AWE metrics and aggregates the results over all models and confidence levels.

4.3 Results and Discussions

This section displays the evaluation of the LUBE-Weibull Based Hybrid Method across five datasets. The performance of the hybrid approach is assessed using metrics PICP, PINAW, ACE and AWE. The results are visualized through prediction interval plots and summarized in tables.

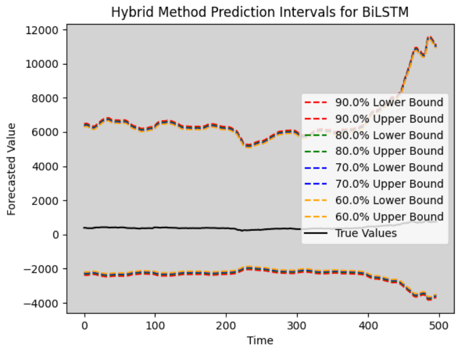
Figures 4.1, 4.2, 4.3, 4.4 and 4.5 shows Prediction Intervals for all the five different datasets obtained using proposed LUBE-Weibull based Hybrid Method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models respectively.

Tables 4.1, 4.2, 4.3, 4.4 and 4.5 shows the performance of the proposed LUBE-Weibull based Hybrid Method on all the five datasets respectively across the four metrics PICP, PINAW, ACE and AWE for four different confidence levels 60%, 70%, 80% and 90%.

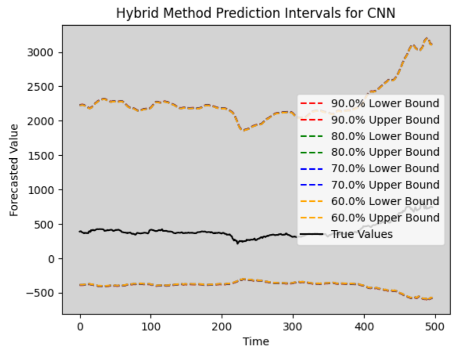
It is visible from Figures 4.1 to 4.5 and from Tables 4.1 to 4.5 that the results obtained from his Hybrid LUBE-Weibull method achieves near 100% PICP across all datasets and confidence levels which may not be ideal in every scenario as the prediction interval width is high but it can be particularly helpful in those applications needing high reliability and assured coverage. Such applications include energy demand forecasting, stock market volatility, and meteorological forecasting, where mis-capture of the true value within the prediction interval can mean large operational or financial risk. In addition, in regulated sectors or risk-averse environments such as healthcare and finance, in which high compliance or risk-averse decision-making is critical, the capability of this hybrid method for assured coverage of prediction intervals about the true outcome is critical to the assurance of trust, safety, and compliance. This method being very computationally efficient is also an added advantage and can be considered as an alternative to Traditional LUBE or other computationally demanding methods.

4.3.1 Visualization of Prediction Intervals

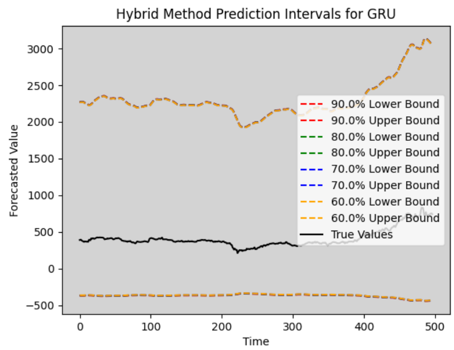
Figures 4.1 to 4.5 illustrate the probabilistic forecasts for each dataset. The black line represents the true values, while the colored dashed lines depict the lower and upper prediction bounds across the 4 confidence levels 90%, 80%, 70% and 60% . These figures represents the capability of the hybrid method to generate adaptive prediction intervals, balancing coverage and sharpness across different confidence levels.



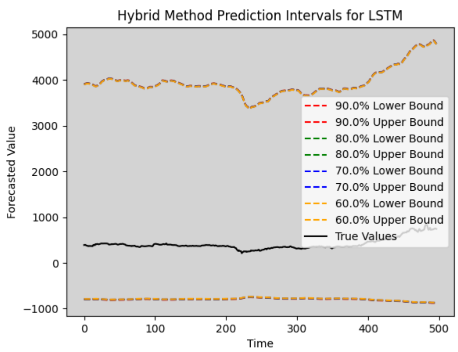
(a) BiLSTM.



(b) CNN.

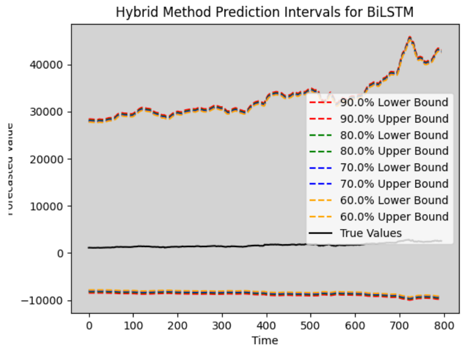


(c) GRU.

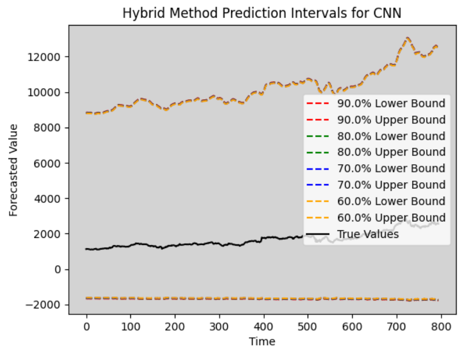


(d) LSTM.

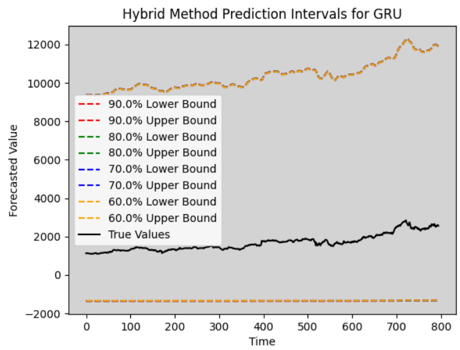
Figure 4.1: Prediction Intervals for Adani Ports dataset obtained using proposed LUBE-Weibull based Hybrid Method and (a)BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.



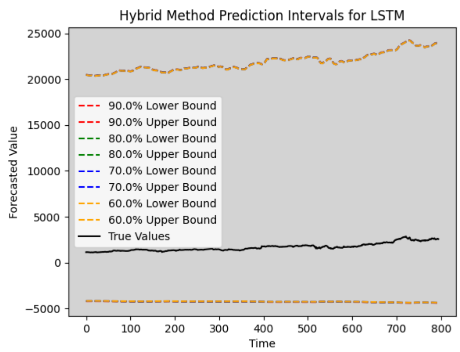
(a) BiLSTM.



(b) CNN.

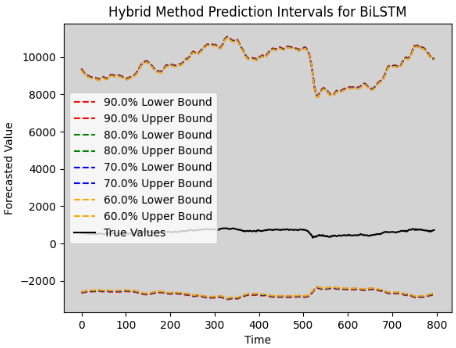


(c) GRU.

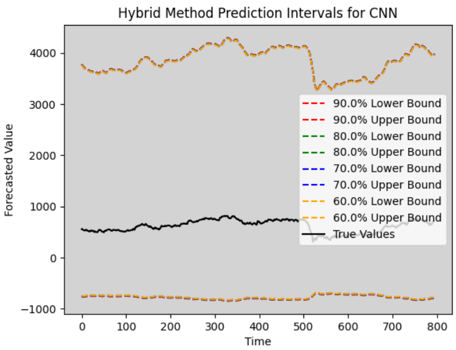


(d) LSTM.

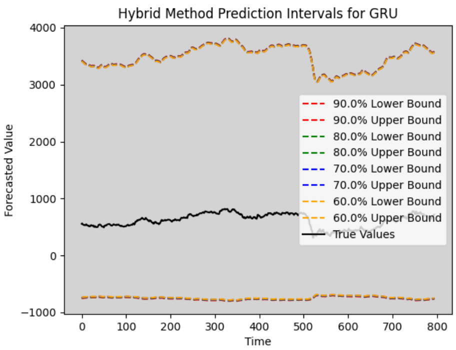
Figure 4.2: Prediction Intervals for Asian Paints dataset obtained using proposed LUBE-Weibull based Hybrid Method and (a)BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.



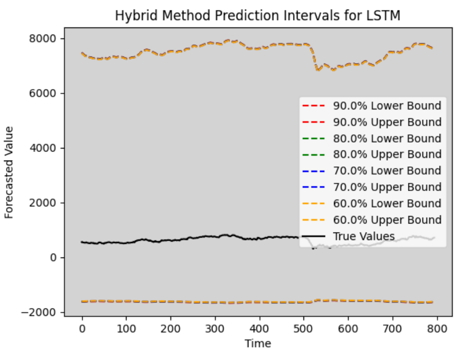
(a) BiLSTM.



(b) CNN.

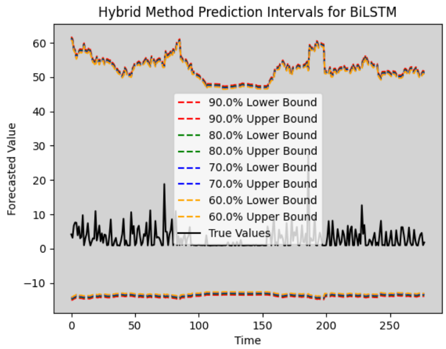


(c) GRU.

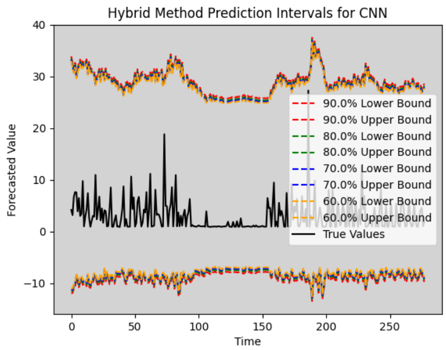


(d) LSTM.

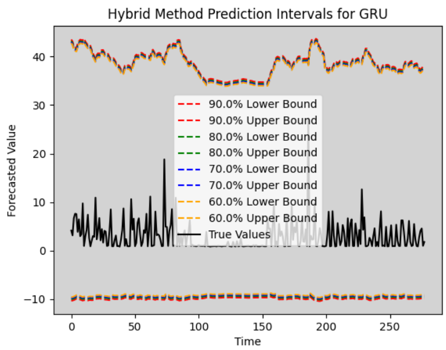
Figure 4.3: Prediction Intervals for Axis Bank dataset obtained using proposed LUBE-Weibull based Hybrid Method and (a)BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.



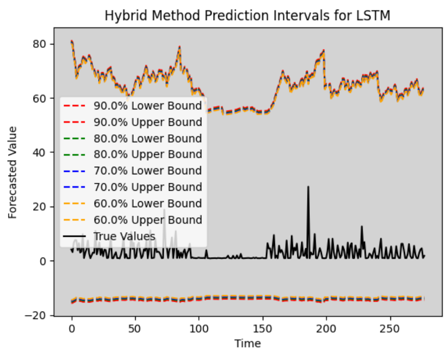
(a) BiLSTM.



(b) CNN.

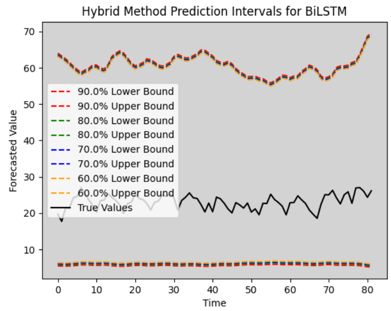


(c) GRU.

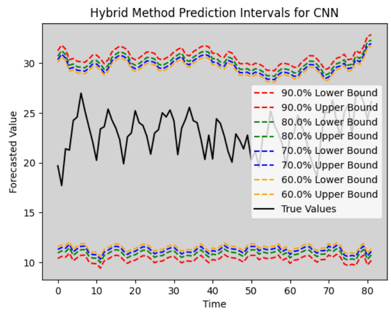


(d) LSTM.

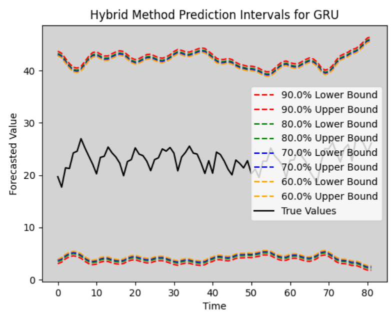
Figure 4.4: Prediction Intervals for Electricity Consumption dataset obtained using proposed LUBE-Weibull based Hybrid Method and (a)BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.



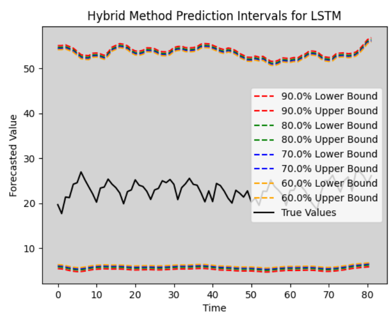
(a) BiLSTM.



(b) CNN.



(c) GRU.



(d) LSTM.

Figure 4.5: Prediction Intervals for Web Traffic dataset obtained using proposed LUBE-Weibull based Hybrid Method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models.

Table 4.1: Performance of LUBE-Weibull Hybrid Method on Adani Ports dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
LUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid Methodpt< -LUBE-Weibull Hybrid Methodpt>	0.6	BiLSTM	100	9.01	40	4995.50
	0.6	CNN	100	4.15	40	1961.88
	0.6	GRU	100	4.17	40	1979.80
	0.6	LSTM	100	7.76	40	4218.70
	0.7	BiLSTM	100	9.08	30	5039.80
	0.7	CNN	100	4.16	30	1968.03
	0.7	GRU	100	4.18	30	1984.13
	0.7	LSTM	100	7.78	30	4227.40
	0.8	BiLSTM	100	9.17	20	5093.81
	0.8	CNN	100	4.17	20	1975.35
	0.8	GRU	100	4.19	20	1989.29
	0.8	LSTM	100	7.80	20	4237.76
	0.9	BiLSTM	100	9.29	10	5170.50
	0.9	CNN	100	4.18	10	1985.47
	0.9	GRU	100	4.20	10	1996.40
	0.9	LSTM	100	7.81	10	4252.00

Table 4.2: Performance of LUBE-Weibull Hybrid Method on Asian Paints dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
LUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid Methodpt< -LUBE-Weibull Hybrid Methodpt>	0.6	BiLSTM	100	23.37	40	39072.46
	0.6	CNN	100	6.84	40	10198.06
	0.6	GRU	100	7.41	40	11189.72
	0.6	LSTM	100	15.27	40	24928.48
	0.7	BiLSTM	100	23.56	30	39400.89
	0.7	CNN	100	6.85	30	10219.74
	0.7	GRU	100	7.42	30	11206.59
	0.7	LSTM	100	15.28	30	24950.86
	0.8	BiLSTM	100	23.78	20	39800.18
	0.8	CNN	100	6.87	20	10245.51
	0.8	GRU	100	7.43	20	11226.99
	0.8	LSTM	100	15.30	20	24977.32
	0.9	BiLSTM	100	24.11	10	40365.09
	0.9	CNN	100	6.89	10	10280.97
	0.9	GRU	100	7.44	10	11255.71
	0.9	LSTM	100	15.32	10	25013.46

Table 4.3: Performance of LUBE-Weibull Hybrid Method on Axis Bank dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
LUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid Methodpt< -LUBE-Weibull Hybrid Methodpt>	0.6	BiLSTM	100	25.69	40	12517.84
	0.6	CNN	100	8.70	40	3905.90
	0.6	GRU	100	9.38	40	4250.76
	0.6	LSTM	100	20.32	40	9793.09
	0.7	BiLSTM	100	25.78	30	12562.57
	0.7	CNN	100	8.72	30	3914.95
	0.7	GRU	100	9.40	30	4259.08
	0.7	LSTM	100	20.34	30	9805.74
	0.8	BiLSTM	100	25.89	20	12616.01
	0.8	CNN	100	8.74	20	3925.73
	0.8	GRU	100	9.42	20	4269.04
	0.8	LSTM	100	20.37	20	9820.72
	0.9	BiLSTM	100	26.03	10	12689.96
	0.9	CNN	100	8.77	10	3940.60
	0.9	GRU	100	9.45	10	4282.84
	0.9	LSTM	100	20.41	10	9841.23

Table 4.4: Performance of LUBE-Weibull Hybrid Method on Electricity Consumption dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
LUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid Methodpt< -LUBE-Weibull Hybrid Methodpt>	0.6	BiLSTM	100	2.54	40	40.62
	0.6	CNN	100	1.37	40	9.85
	0.6	GRU	100	1.86	40	22.64
	0.6	LSTM	100	2.84	40	48.53
	0.7	BiLSTM	100	2.55	30	41.07
	0.7	CNN	100	1.39	30	10.30
	0.7	GRU	100	1.87	30	23.01
	0.7	LSTM	100	2.86	30	49.06
	0.8	BiLSTM	100	2.57	20	41.60
	0.8	CNN	100	1.41	20	10.85
	0.8	GRU	100	1.89	20	23.46
	0.8	LSTM	100	2.88	20	49.71
	0.9	BiLSTM	100	2.60	10	42.35
	0.9	CNN	100	1.44	10	11.65
	0.9	GRU	100	1.91	10	24.10
	0.9	LSTM	100	2.91	10	50.61

Table 4.5: Performance of LUBE-Weibull Hybrid Method on Web Traffic dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
LUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid MethodLUBE-Weibull Hybrid Methodpt< -LUBE-Weibull Hybrid Methodpt>	0.6	BiLSTM	100	6.04	40	46.91
	0.6	CNN	100	2.16	40	10.82
	0.6	GRU	100	3.65	40	24.64
	0.6	LSTM	100	5.35	40	40.52
	0.7	BiLSTM	100	6.09	30	47.37
	0.7	CNN	100	2.21	30	11.27
	0.7	GRU	100	3.70	30	25.11
	0.7	LSTM	100	5.40	30	40.96
	0.8	BiLSTM	100	6.15	20	47.94
	0.8	CNN	100	2.27	20	11.86
	0.8	GRU	100	3.76	20	25.73
	0.8	LSTM	100	5.46	20	41.52
	0.9	BiLSTM	100	6.24	10	48.76
	0.9	CNN	100	2.37	10	12.79
	0.9	GRU	100	3.87	10	26.69
	0.9	LSTM	100	5.55	10	42.35

4.3.2 Discussion

The results demonstrate that the LUBE-Weibull Hybrid Method maintains 100% PICP like the standalone LUBE method without compromising acceptable PINAW values. The ACE values reflect that the method achieves confidence levels that are essentially similar to the target measures. Additionally, the AWE values confirms that the Weibull-based adjustment prevents the over-broadening of prediction intervals. We have certainly achieved better results than the traditional LUBE method while also significantly reducing the required time and computational requirements by incorporating the Weibull method.

The last section provides a summary of the research and suggests potential avenues for future research.

4.4 Summary

This chapter introduced the LUBE-Weibull Based Hybrid Approach to probabilistic time series forecasting, which integrates deep learning-driven prediction interval estimation with Weibull-based residual correction. Experimental results demonstrate that the hybrid approach outperforms the Traditional LUBE Method in terms of improving coverage probability (PICP) with an acceptable balance between interval sharpness (PINAW) and accuracy (ACE). Deployment of Weibull-based correction effectively sharpens prediction intervals by accounting for residual uncertainties, thus improving forecast reliability.

Comparison with the Advanced LUBE Method reveals that the hybrid approach has a marginally increased AWE, indicating slightly wider prediction intervals. However, this feature may offer a potential benefit: the Hybrid Method is computationally more efficient, as it removes extra complexity in the loss function while performing post-hoc corrections to sharpen interval estimation. This trade-off suggests that the Hybrid Method offers a viable and scalable alternative to Advanced LUBE, especially in situations requiring tradeoff between computational cost and forecasting quality.

Future research can explore adaptive scaling of Weibull corrections to further improve interval sharpness without compromising coverage. Moreover, the use of uncertainty-aware deep learning architectures or ensembling of heterogeneous architectures may potentially improve predictive quality.

Chapter 5

LUBE-QR Based Hybrid Method For Probabilistic Time Series Forecasting

5.1 Motivation

Forecasting is very important in domains like energy, finance, healthcare, and climatology, as accurate predictions allow for well-informed and efficient decision making. While traditional forecasting methodologies tend to produce point estimates, they have the disadvantage of excluding uncertainty, an oversight that can lead to erroneous conclusions and inefficient use of resources, especially in risky scenarios. Probabilistic forecasting rectifies this by providing prediction intervals (PIs) that estimate uncertainty and allow for risk informed decision making. However, choice of the most appropriate method is a major problem. Parametric methods, including Gaussian and Weibull Distribution based methods, are computationally efficient but can fall short of capturing complex patterns in the data. Non-parametric methods, like Quantile Regression and Bootstrap-based methods are more data adaptive and versatile but are computationally demanding.

The Lower Upper Bound Estimation (LUBE) algorithm, a deep learning-based non-parametric algorithm, learns PIs from data but lacks reliability in highly dynamic environments. Weibull distribution modeling, in contrast, models residual errors accurately but lacks adaptability to time-dependent forecast changes.

This chapter presents the LUBE-QR Based Hybrid Method, a technique that improves PI reliability through the combination of deep learning-based LUBE prediction and QR-based residual correction.

The technique consists of using deep learning models (LSTM, CNN, GRU, BiLSTM) to produce initial prediction intervals, and residual error modeling through QR method. The QR-based corrections improve the intervals, which improves accuracy and stability at various confidence levels.

Through the combination of the flexibility of deep learning and statistical error modeling, the technique presents stronger and more reliable probabilistic predictions.

5.2 Methodology

5.2.1 Data Pre-processing

The time series datasets are preprocessed to ensure stable training and effective modeling. First, the data is normalized to the range [0,1] using MinMaxScaler. Next, a sliding window approach with a window size of 12 is applied to generate input-output pairs, where the past 12 time steps are used to predict the next step. Finally, the dataset is divided into three subsets: 70% for training, 15% for validation, and 15% for testing, ensuring a balanced and robust evaluation of the model.

5.2.2 Model Selection

The hybrid method employs different deep learning architectures to generate first-stage prediction intervals (PIs). Long Short-Term Memory (LSTM) is well suited to learning long-term dependencies of sequential data, while Convolutional Neural Networks (CNN) are best at detecting local patterns and trends. Gated Recurrent Units (GRU) offer a computationally cheaper option that still retains the ability to learn sequences. Additionally, Bidirectional LSTM (BiLSTM) enhances feature extraction through learning information in both directions. Each model delivers two outputs for each prediction, which represent the lower and upper bounds of the PI and hence enable extensive estimation of uncertainty.

5.2.3 LUBE Loss Function

The LUBE method directly learns interval bounds using a custom loss function. The LUBE loss function is defined as:

$$\mathcal{L}_{\text{LUBE}} = \text{PINAW} + \lambda \cdot \max\left(0, (1 - \text{PICP}_{\text{target}}) - \text{PICP}\right)^2 \tag{5.1}$$

Where:

- PINAW is Prediction Interval Normalized Average Width, which penalizes wide intervals.
- PICP is Prediction Interval Coverage Probability, representing the fraction of true values that lie within the predicted intervals.
- λ is a regularization hyperparameter that controls the trade-off between narrow intervals and sufficient coverage.
- UB_i, LB_i are the predicted upper and lower bounds for the i^{th} data point.
- y_i is the true value of the i^{th} sample.
- $\text{PICP}_{\text{target}}$ is the desired confidence level, such as 0.9, 0.8, 0.7 or 0.6.
- R is the range of the training target values, used to normalize the interval width.
- $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition inside is true, and 0 otherwise.

The loss aims to produce prediction intervals that are as narrow as possible (minimizing PINAW), while ensuring they capture the true target values with the desired coverage level (maximizing PICP).

5.2.4 QR-Based Residual Correction

After generating initial prediction intervals $[LB_i, UB_i]$ using the Advanced LUBE method, the Hybrid LUBE–QR framework applies a correction based on quantile regression to refine the interval bounds using residual information.

Let the residuals be defined as:

$$r_i = |y_i - \hat{y}_i| \tag{5.2}$$

where y_i is the true target value and $\hat{y}_i = \frac{LB_i + UB_i}{2}$ is the midpoint (mean prediction) of the LUBE interval.

A Quantile Regression (QR) model is then trained to estimate specific conditional quantiles of the residuals, denoted as $Q_\tau(r|x)$, where τ represents the desired quantile level (e.g., $\tau = 0.9, 0.95$). The model learns to predict asymmetric quantiles based on the input features x.

For a target confidence level c, we determine the corresponding quantile correction δ_c such that:

$$\delta_c = Q_{1-\frac{1-c}{2}}(r|x) \tag{5.3}$$

The final corrected prediction interval is:

$$LB_i^{\text{corrected}} = \hat{y}_i - \delta_c \tag{5.4a}$$

$$UB_i^{\text{corrected}} = \hat{y}_i + \delta_c \tag{5.4b}$$

The Eq. 5.3 shows the quantile correction formula while the Eq. 5.4 shows the corrected Lower Bound and Upper Bound obtained. This residual-based QR correction enables the prediction intervals to better account for heteroscedasticity and non-Gaussian uncertainty structures, resulting in intervals that are not only statistically valid but also adaptive to local data variability.

5.2.5 Confidence Levels and Performance Metrics

The hybrid method examines prediction intervals for four confidence levels: 90%, 80%, 70%, and 60%, giving a complete uncertainty estimation evaluation. Performance is measured via several key indicators. Prediction Interval Coverage Probability (PICP) estimates the percentage of actual values in the predicted interval, assessing reliability. Prediction Interval Normalized Average Width (PINAW) estimates interval sharpness as a function of data range, sacrificing precision for coverage. Average Coverage Error (ACE) estimates PICP deviation from the target confidence level, measuring calibration accuracy. Lastly, Average Width Error (AWE) evaluates how far interval width is from expected bounds, ensuring proper uncertainty quantification.

5.2.6 Probabilistic Forecasting using Hybrid LUBE-QR Based Method

The crux of this algorithm begins with training deep-learning models (LSTM, BiLSTM, CNN, GRU) on a tailored LUBE loss function that is specifically tailored to optimize interval width and violation of coverage. For each pre-specified confidence level (0.6, 0.7, 0.8, 0.9), the LUBE loss is dynamically scaled to penalize the failure of prediction intervals to cover the true values. This training produces initial lower and upper bounds for the target variable, which are the initial prediction intervals.

Lastly, the method estimates the midpoint of the LUBE-generated intervals and the absolute residuals between the predicted midpoint and observed target values are estimated. These residuals are the systematic forecasting errors not explained by the original model. Quantile Regression is applied to the residuals to estimate the lower and upper quantiles at each confidence level. This helps to model the residual distribution in a non-parametric and data-adaptive manner. The quantile forecasts are subsequently combined with the forecasted midpoint to provide improved lower and upper limits for the final prediction intervals. The two-stage adjustment process enables the model to improve its interval estimates by exploiting both interval-targeted loss optimization and quantile-based residual learning.

For each run and model, the following metrics are computed for each run and model to measure the quality of the prediction intervals: Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Average Width (PINAW), Average Coverage Error (ACE) and Absolute Width Error (AWE). The whole process is repeated for 10 independent runs for all types of models and confidence levels to provide statistical robustness. Finally, the results are aggregated and saved in structured CSV formats, and graphical visualizations are generated for all models, showing the predicted intervals over the test data for different confidence levels.

5.3 Results and Discussions

This section displays the evaluation of the LUBE-QR Based Hybrid Method across five datasets. The performance of the hybrid approach is assessed using metrics PICP, PINAW, ACE and AWE. The results are visualized through prediction interval plots

Algorithm 8: Hybrid LUBE–QR Method.

Input: Time series dataset D
Output: Predicted intervals $[LB', UB']$, PICP, PINAW, ACE, AWE

```
Step 1: Data Preprocessing
Normalize D using MinMaxScaler
Generate input-output pairs with window size w
Split into train, validation, and test sets
Step 2: Define Advanced LUBE Loss
foreach c ∈ {0.9, 0.8, 0.7, 0.6} do
    q = 1 − c
    Compute LB, UB
    Compute Loss_lower
    Compute Loss_upper
    Compute PICP (as in Eq. (3.1)) and PINAW (as in Eq. (3.2))
    Compute Loss_LUBE (as in Eq. (5.1)).
Step 3: Model Training
foreach M ∈ {LSTM, CNN, GRU, BiLSTM} do
    Define model architecture
    Compile with Advanced LUBE loss
    Train on (X_train, y_train) for e epochs
    Validate on (X_val, y_val)
    Predict LUBE bounds LB, UB on test data
    Compute midpoint ŷ
Step 4: Fit Quantile Regression on Residuals
Compute residuals (as in Eq. (5.2))
foreach c ∈ {0.9, 0.8, 0.7, 0.6} do
    Train Quantile Regression models to estimate:
    Lower quantile and Upper quantile
    Predict residual quantiles r̂_lower, r̂_upper
Step 5: Adjust Prediction Intervals Using QR Correction
foreach c ∈ {0.9, 0.8, 0.7, 0.6} do
    LB' = ŷ + r̂_lower
    UB' = ŷ + r̂_upper
Step 6: Evaluation Metrics Compute the PICP (as in Eq. (3.1)), PINAW (as in Eq.
  (3.2)), ACE (as in Eq. (3.3)) and AWE (as in Eq. (3.4)) of the computed prediction
  intervals.
Step 7: Aggregate Results
Compute mean of metrics for each case
```

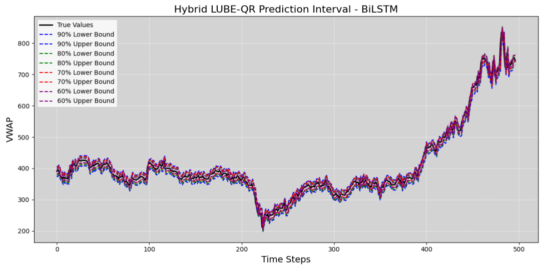
and summarized in tables.

Figures 5.1, 5.2, 5.3, 5.4 and 5.5 shows Prediction Intervals for all the five different datasets obtained using proposed LUBE-QR based Hybrid Method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models respectively.

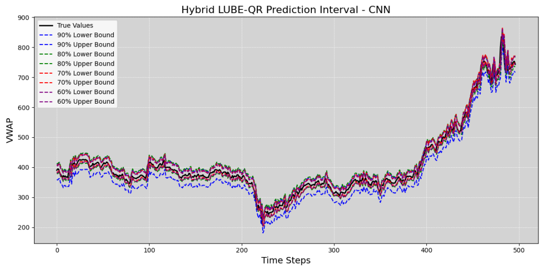
Tables 5.1, 5.2, 5.3, 5.4 and 5.5 shows the performance of the proposed LUBE-QR based Hybrid Method on all the five datasets respectively across the four metrics PICP, PINAW, ACE and AWE for four different confidence levels 60%, 70%, 80% and 90%.

It is visible from Figures 5.1 to 5.5 and from Tables 5.1 to 5.5 that the results obtained from this Hybrid LUBE-QR method produces crisp and sharp intervals while maintaining low values of PINAW, ACE and AWE. This makes this hybrid model a good alternative to existing traditional models for real time probabilistic forecasting tasks.

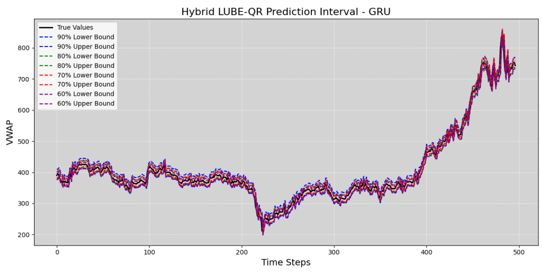
5.3.1 Visualization Of Prediction Intervals



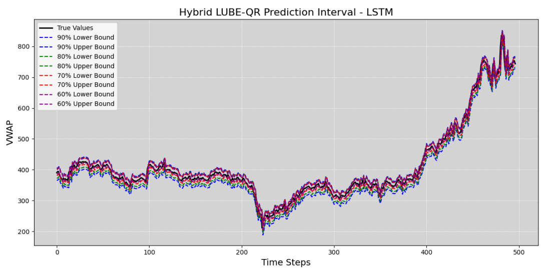
(a) BiLSTM.



(b) CNN.

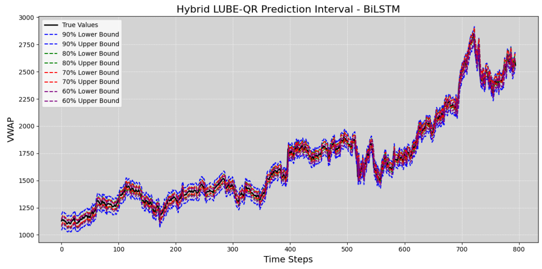


(c) GRU.

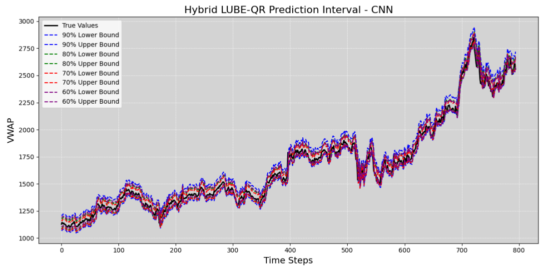


(d) LSTM.

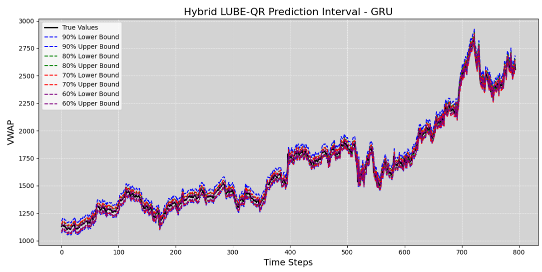
Figure 5.1: Prediction Intervals for Adani Ports dataset obtained using proposed LUBE-QR based Hybrid Method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models respectively.



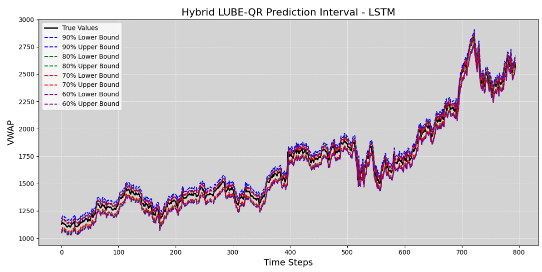
(a) BiLSTM.



(b) CNN.

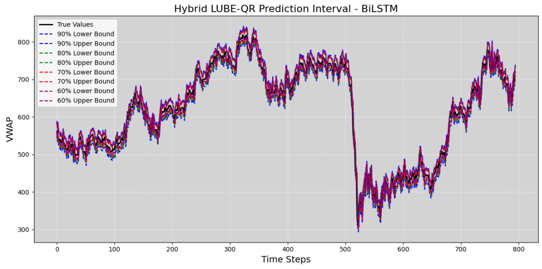


(c) GRU.

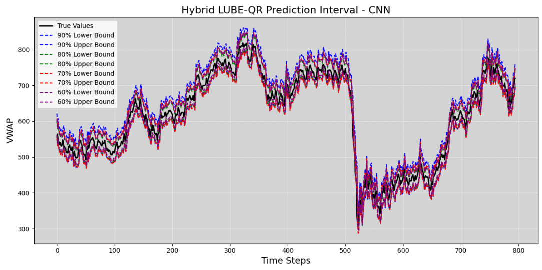


(d) LSTM.

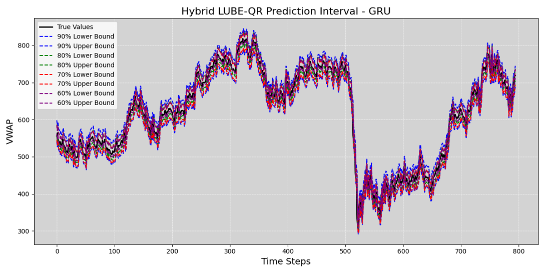
Figure 5.2: Prediction Intervals for Asian Paints dataset obtained using proposed LUBE-QR based Hybrid Method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models respectively.



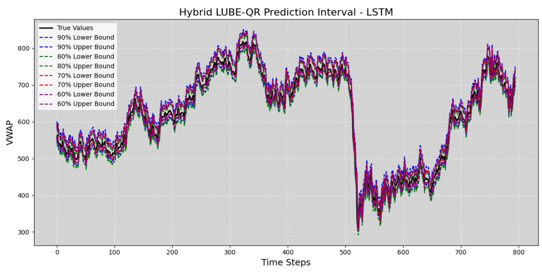
(a) BiLSTM.



(b) CNN.

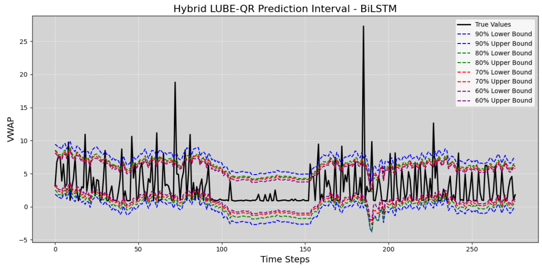


(c) GRU.

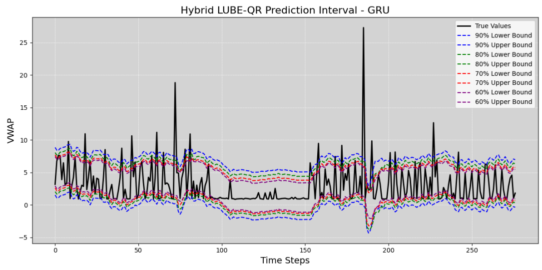


(d) LSTM.

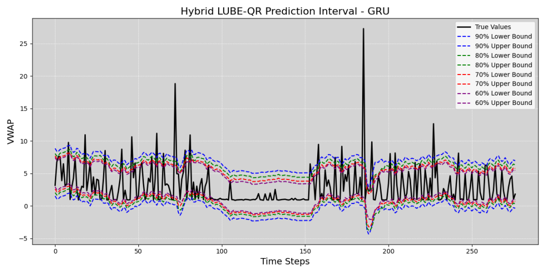
Figure 5.3: Prediction Intervals for Axis Bank dataset obtained using proposed LUBE-QR based Hybrid Method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models respectively.



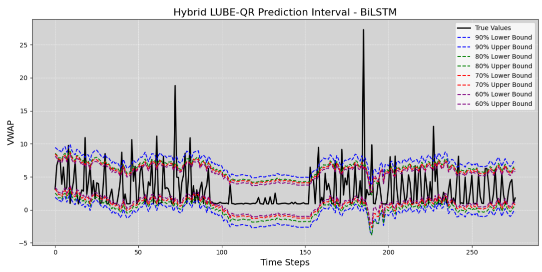
(a) BiLSTM.



(b) CNN.

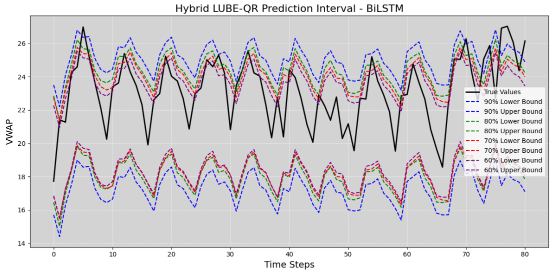


(c) GRU.

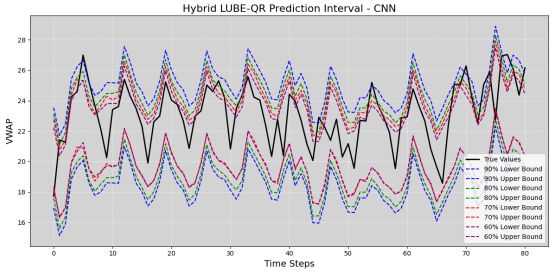


(d) LSTM.

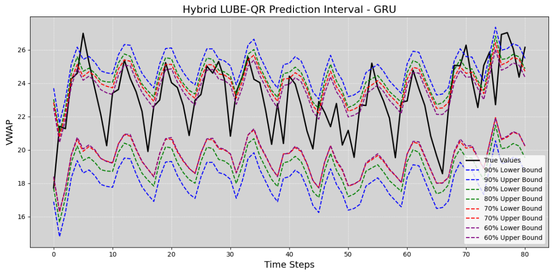
Figure 5.4: Prediction Intervals for Electricity Consumption dataset obtained using proposed LUBE-QR based Hybrid Method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models respectively.



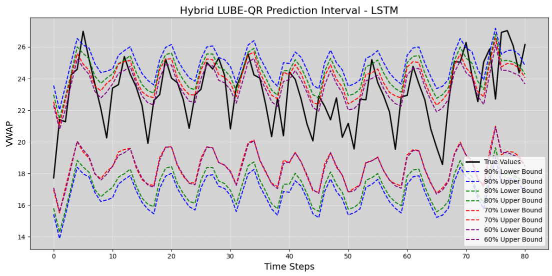
(a) BiLSTM.



(b) CNN.



(c) GRU.



(d) LSTM.

Figure 5.5: Prediction Intervals for Web Traffic Load dataset obtained using proposed LUBE-QR based Hybrid Method and (a) BiLSTM, (b) CNN, (c) GRU, (d) LSTM Models respectively.

Table 5.1: Performance of LUBE-QR Hybrid Method on Adani Ports dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
LUBE-QR Hybrid MethodLUBE-QR Hybrid MethodLUBE-QR Hybrid Methodpt< -LUBE-QR Hybrid Methodpt>	0.6	BiLSTM	59.96	0.02	0.04	608.92
	0.6	CNN	59.96	0.04	0.04	600.10
	0.6	GRU	59.96	0.03	0.04	604.20
	0.6	LSTM	59.96	0.02	0.04	608.53
	0.7	BiLSTM	70.02	0.03	0.02	602.85
	0.7	CNN	70.02	0.04	0.02	600.50
	0.7	GRU	70.02	0.03	0.02	602.87
	0.7	LSTM	70.02	0.03	0.02	602.49
	0.8	BiLSTM	79.88	0.04	0.12	599.96
	0.8	CNN	79.88	0.05	0.12	590.15
	0.8	GRU	79.88	0.04	0.12	600.54
	0.8	LSTM	79.88	0.05	0.12	593.65
	0.9	BiLSTM	89.94	0.05	0.06	591.02
	0.9	CNN	89.94	0.08	0.06	572.53
	0.9	GRU	89.94	0.05	0.06	590.39
	0.9	LSTM	89.94	0.06	0.06	583.94

Table 5.2: Performance of LUBE-QR Hybrid Method on Asian Paints dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
LUBE-QR Hybrid MethodLUBE-QR Hybrid MethodLUBE-QR Hybrid Methodpt< -LUBE-QR Hybrid Methodpt>	0.6	BiLSTM	60.00	0.04	0.00	1679.25
	0.6	CNN	60.00	0.05	0.00	1660.44
	0.6	GRU	60.00	0.02	0.00	1704.50
	0.6	LSTM	60.00	0.06	0.00	1639.36
	0.7	BiLSTM	69.94	0.03	0.06	1687.10
	0.7	CNN	69.94	0.04	0.06	1668.43
	0.7	GRU	69.94	0.04	0.06	1675.49
	0.7	LSTM	69.94	0.04	0.06	1670.81
	0.8	BiLSTM	80.00	0.04	0.00	1679.75
	0.8	CNN	80.00	0.05	0.00	1659.36
	0.8	GRU	80.00	0.04	0.00	1685.31
	0.8	LSTM	80.00	0.05	0.00	1662.02
	0.9	BiLSTM	89.94	0.08	0.06	1603.47
	0.9	CNN	89.94	0.08	0.06	1608.31
	0.9	GRU	89.94	0.06	0.06	1637.27
	0.9	LSTM	89.94	0.07	0.06	1620.58

Table 5.3: Performance of LUBE-QR Hybrid Method on Axis Bank dataset.

Method Used	**Confidence Level**	**Model**	**Avg PICP**	**Avg PINAW**	**Avg ACE**	**Avg AWE**
LUBE-QR Hybrid MethodLUBE-QR Hybrid MethodLUBE-QR Hybrid Methodpt< -LUBE-QR Hybrid Methodpt>	0.6	BiLSTM	60.00	0.05	0.00	482.85
	0.6	CNN	60.00	0.09	0.00	463.85
	0.6	GRU	60.00	0.05	0.00	483.88
	0.6	LSTM	60.00	0.06	0.00	476.72
	0.7	BiLSTM	69.94	0.05	0.06	482.26
	0.7	CNN	69.94	0.11	0.06	452.96
	0.7	GRU	69.94	0.08	0.06	466.92
	0.7	LSTM	69.94	0.06	0.06	474.35
	0.8	BiLSTM	80.00	0.06	0.00	477.10
	0.8	CNN	80.00	0.09	0.00	462.02
	0.8	GRU	80.00	0.06	0.00	475.72
	0.8	LSTM	80.00	0.07	0.00	468.94
	0.9	BiLSTM	89.94	0.08	0.06	467.31
	0.9	CNN	89.94	0.13	0.06	441.30
	0.9	GRU	89.94	0.11	0.06	450.19
	0.9	LSTM	89.94	0.09	0.06	458.83

Table 5.4: Performance of LUBE-QR Hybrid Method on Electricity Consumption dataset.

Method Used	**Confidence Level**	**Model**	**Avg PICP**	**Avg PINAW**	**Avg ACE**	**Avg AWE**
LUBE-QR Hybrid MethodLUBE-QR Hybrid MethodLUBE-QR Hybrid Methodpt< -LUBE-QR Hybrid Methodpt>	0.6	BiLSTM	59.93	0.18	0.07	21.71
	0.6	CNN	59.93	0.17	0.07	22.00
	0.6	GRU	59.93	0.17	0.07	21.92
	0.6	LSTM	59.93	0.16	0.07	22.10
	0.7	BiLSTM	70.04	0.20	0.04	21.11
	0.7	CNN	70.04	0.21	0.04	20.96
	0.7	GRU	70.04	0.19	0.04	21.33
	0.7	LSTM	70.04	0.19	0.04	21.51
	0.8	BiLSTM	79.78	0.23	0.22	20.40
	0.8	CNN	79.78	0.25	0.22	19.89
	0.8	GRU	79.78	0.23	0.22	20.43
	0.8	LSTM	79.78	0.22	0.22	20.61
	0.9	BiLSTM	89.89	0.29	0.11	18.88
	0.9	CNN	89.89	0.32	0.11	17.87
	0.9	GRU	89.89	0.28	0.11	19.11
	0.9	LSTM	89.89	0.29	0.11	18.90

Table 5.5: Performance of LUBE-QR Hybrid Method on Web Traffic dataset.

Method Used	Confidence Level	Model	Avg PICP	Avg PINAW	Avg ACE	Avg AWE
LUBE-QR Hybrid MethodLUBE-QR Hybrid MethodLUBE-QR Hybrid Methodpt< -LUBE-QR Hybrid Methodpt>	0.6	BiLSTM	60.25	0.58	0.54	3.88
	0.6	CNN	60.00	0.44	0.59	5.23
	0.6	GRU	60.37	0.44	0.52	5.19
	0.6	LSTM	60.49	0.56	0.49	4.12
	0.7	BiLSTM	69.75	0.64	0.62	3.40
	0.7	CNN	69.63	0.48	0.67	4.88
	0.7	GRU	70.00	0.48	0.52	4.82
	0.7	LSTM	69.75	0.60	0.62	3.75
	0.8	BiLSTM	79.14	0.69	0.91	2.91
	0.8	CNN	79.38	0.59	0.77	3.77
	0.8	GRU	79.75	0.57	0.54	3.96
	0.8	LSTM	79.26	0.77	0.84	2.35
	0.9	BiLSTM	89.01	0.84	1.01	1.51
	0.9	CNN	89.14	0.71	0.91	2.72
	0.9	GRU	89.01	0.73	1.01	2.50
	0.9	LSTM	89.38	0.87	0.72	1.39

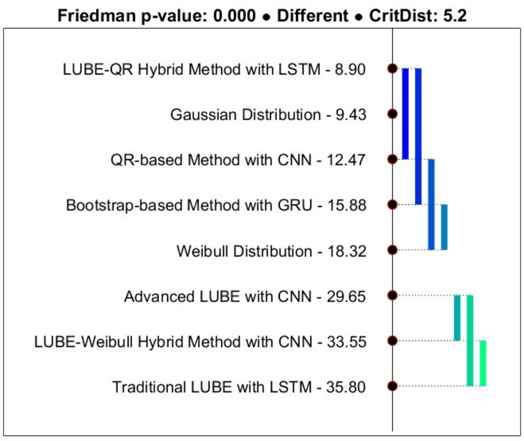


Figure 5.6: Statistical Analysis on PINAW Metric performed on all the five datasets together.

5.3.2 Statistical Analysis

Figures 5.6 and 5.7 shows the Friedman-Nemenyi hypothesis results obtained on the PINAW and ACE Metrics respectively across all the five datasets and all the eight different methods (Each method is coupled with a Deep Learning Model with which it had performed best). It proves that the proposed LUBE-QR based Hybrid Method is a statistically better or equivalent method to other existing Traditional Probabilistic Forecasting methods.

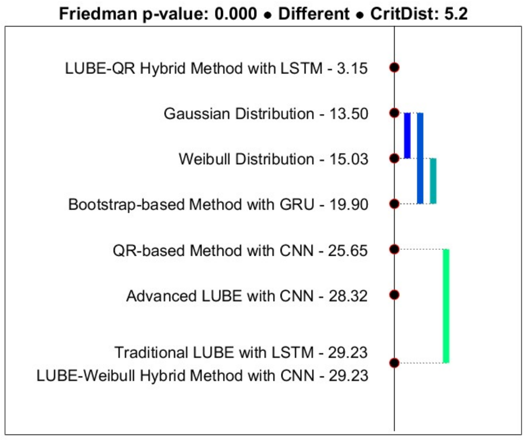


Figure 5.7: Statistical Analysis on ACE Metric performed on all the five datasets together.

5.3.3 Discussion

The hybrid method based on LUBE–QR that has been proposed introduces a new solution for increasing the reliability and sharpness of prediction intervals for probabilistic time series forecasting. By capitalizing on the respective strengths of both the Advanced LUBE technique and Quantile Regression (QR), the approach is able to overcome the limitations within each individual technique. The LUBE component achieves direct interval prediction using deep learning with self-defined loss optimization, and QR supplies data-driven correction of residual errors to enhance the adaptiveness of predicted bounds.

Experimental results show that the hybrid approach persistently yields smaller ACE (Absolute Coverage Error) and PINAW (Prediction Interval Normalized Average Width) values than individual LUBE, QR, or other baseline models on several deep learning architectures (LSTM, BiLSTM, CNN, GRU). This shows that the hybrid intervals are both narrow and well-calibrated, the best combination for good uncertainty quantification. The Friedman–Nemenyi hypothesis test also verifies the statistical performance superiority of the hybrid method, with the method being ranked first with all evaluation metrics in various confidence levels (0.6 to 0.9).

In general, the hybridization of LUBE and QR is a strong and effective approach to generating high-quality prediction intervals and one that can chart a promising path forward for uncertainty-aware forecasting in the future.

5.4 Summary

In this chapter, we proposed a novel LUBE–QR hybrid technique that seamlessly blends the interval forecasting capabilities of the Advanced LUBE method with quantile regression on residuals' flexibility. By joining these two approaches, the proposed method provided more precise and sharper prediction intervals at different levels of confidence as verified through detailed experiments and evaluation metrics. In contrast to traditional approaches which tend to have a compromise between interval width and coverage, the hybrid approach presents a better balanced trade-off. The uniformity of performance in various deep learning models and statistical superiority as witnessed by the Friedman–Nemenyi test reflect its strength. In the future, follow-up work may investigate extending this hybrid model to multivariate time series forecasting and real-time adaptive interval updates and incorporating sophisticated ensemble methods or probabilistic Bayesian layers to further boost uncertainty quantification in dynamic high-noise conditions.

Chapter 6

Conclusion and Future Work

In this thesis we have first conducted a thorough study of Traditional Probabilistic Forecasting methods divided into two groups: Parametric under which we've covered Gaussian Distribution based Method and Weibull Distribution based Method and Non-parametric under which we've covered Traditional LUBE, Advanced LUBE, QR-based Method and Bootstrap-based Method. We have evaluated the non-parametric methods using four different Deep Learning Models: LSTM, CNN, GRU and BiLSTM across four different confidence levels 90%, 80%, 70% and 60% and also run simulations for ten times to obtain the average results and then plotted the graphs using the obtained results. We have then compared the results to conclude the best performing Traditional Probabilistic Forecasting Method.

Then, we developed our first Hybrid method using two different existing methods Traditional LUBE Method and Weibull Distribution based Method. We used the same four DL models to assess it's performance across the four different confidence levels and obtained average results and graphs in a similar fashion. The key takeaway from this Hybrid Method was that it performed close to Advacned LUBE method giving similar PICP values (close to 100, thus making the intervals wide and reducing sharpness) while being computationally very efficient and thus it can effectively replace systems using Traditionl LUBE, Advanced LUBE or other computationally heavy methods with a slight tradeoff for wider intervals (conservative performance).

Next, we developed our second Hybrid method using two different and popular Non-Parametric based methods namely Advanced LUBE and QR-based Methods. This method achieved the best results that we've gotten so far from any model. It achieved PICP values closer to the actual confidence levels and maintained the lowest PINAW, ACE and AWE values. It produced crisp and sharp intervals while barely overfitting or underfitting the data. This method can be used as an effective alternative to existing traditional probabilistic forecasting methods but it can be a bit computationally intensive.

Future Work may include deep diving into more hybrid methods by combining and fusing multiple existing methods with various different DL models to achieve better probabilistic forecasting results.