MACHINE LEARNING TIME SERIES FORECASTING: A COMPREHENSIVE SURVEY AND STOCK MARKET APPLICATION

by

TIMOTHY HALL

(Under the Direction of Khaled Rasheed)

ABSTRACT

This thesis presents two complementary studies that advance the understanding and application of machine learning techniques in time series forecasting, with a particular focus on financial markets. A comprehensive survey identifies top-performing techniques across tree-based models, deep learning architectures, and hybrid approaches. Building on these insights, the thesis applies a specialized forecasting framework to the domain of day trading. By leveraging a combination of LightGBM models with an extensive set of engineered features ranging from multi-timeframe technical indicators to contextual stock attributes, the model uses two years of second-by-second trade and quote data to estimate risk-reward ratios over multiple forward time horizons. Simulated results using realistic execution constraints demonstrate a pronounced performance advantage over human day traders, yielding daily returns several orders of magnitude higher.

INDEX WORDS: TIME SERIES FORECASTING, DAY TRADING, STOCK MARKET, FINANCIAL MARKETS, MACHINE LEARNING, TREE-BASED, LIGHTGBM

Machine Learning Time Series Forecasting: A Comprehensive Survey and Stock Market Application

by

Timothy Hall

B.S., University of Georgia, 2024

A Thesis Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Master of Science

Athens, Georgia

2025

©2025

Timothy Hall

All Rights Reserved

MACHINE LEARNING TIME SERIES FORECASTING: A COMPREHENSIVE SURVEY AND STOCK MARKET APPLICATION

by

TIMOTHY HALL

Major Professor: Khaled Rasheed

Committee: Frederick Maier
Jin Lu

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
May 2025

ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. Rasheed Khaled, for his insight and guidance throughout the writing of this thesis. I am also deeply grateful to my advisory committee, Frederick Maier and Jin Lu, for their support and thoughtful feedback on both coursework and thesis work. I would like to thank all my undergraduate and graduate instructors for equipping me with the knowledge and skills necessary to undertake this research. Lastly, I am forever thankful to my parents for their unwavering support and encouragement throughout my academic journey. This thesis would not have been possible without the mentorship, love, and support I have received along the way.

CONTENTS

Acknowledgements **iv**

List of Figures **vi**

List of Tables **vii**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Purpose and Scope . . . 2
1.3 Contributions of This Work . . . 2
1.4 Structure of the Thesis . . . 4

2 A Survey of Machine Learning Methods for Time Series Prediction **5**
2.1 Introduction . . . 6
2.2 Methodology . . . 7
2.3 Tree-Based Machine Learning Architectures . . . 9
2.4 Deep Learning Architectures . . . 12
2.5 Experimental Results and Discussion . . . 15
2.6 M5 and M6 Forecasting Competitions . . . 41
2.7 Conclusion . . . 45

3 Stock Market Application Paper **58**
3.1 Introduction . . . 59
3.2 Related Works . . . 61
3.3 Methodology . . . 64
3.4 Results . . . 71
3.5 Discussion . . . 81
3.6 Conclusion . . . 85

4 Conclusion **89**

LIST OF FIGURES

2.1 RF and GBDT Architecture 11
2.2 FFNN, CNN, and RNN Architecture 13
2.3 Overall Model Performance FPA and WRA scores 18
2.4 Sub-class Model Performance FPA and WRA scores 19
2.5 Individual Model Performance FPA scores 20
2.6 Individual Model Performance WRA scores 21
2.7 Time Series Prediction Task Categories 22
2.8 Task-Specific Model Performance FPA scores 24
2.9 Task-Specific Model Performance WRA scores 24
2.10 Dataset Size Model Performance FPA scores 25
2.11 Dataset Size Model Performance WRA scores 26
2.12 Time Interval Model Performance FPA scores 27
2.13 Time Interval Model Performance WRA scores 28
2.14 Research Focus Model Performance FPA scores 29
2.15 Research Focus Model Performance WRA scores 30
2.16 Error Metrics for Classification Models 33
2.17 Error Metrics for Regression Models 34
2.18 Hyperparameter Optimization Techniques 35
3.1 Model 1 Heatmap 72
3.2 Model 1 Cumulative Profit 73
3.3 Model 2 Heatmap 74
3.4 Model 2 Cumulative Profit 75
3.5 Model 3 Heatmap 76
3.6 Model 3 Cumulative Profit 77
3.7 Model 4 Heatmap 78
3.8 Model 4 Cumulative Profit 79
3.9 Model 5 Heatmap 80
3.10 Model 5 Cumulative Profit 80

LIST OF TABLES

2.1 Time Series Prediction Tasks 23
3.1 Model 1 Performance Metrics 74
3.2 Model 2 Performance Metrics 75
3.3 Model 3 Performance Metrics 77
3.4 Model 4 Performance Metrics 78
3.5 Model 5 Performance Metrics 81

CHAPTER 1

Introduction

1.1 Motivation

Time series prediction is a widely recognized and important application of machine learning (ML). Time series prediction usually relies on using historical patterns to accurately forecast future data points. Its significance exists across many domains, including finance, healthcare, energy management, and weather forecasting. In each of these domains, the ability to accurately model and forecast future values holds potential to optimize decision-making, improve performance, or mitigate risks. Among all time series prediction tasks, forecasting stock market behavior is one of the most difficult. This is due to the many complex factors including the countless individuals and institutions acting on the market at the same time. Within this domain, day trading, where positions are opened and closed within the same trading day poses an even greater challenge.

This difficulty, however, also presents an opportunity. While human traders are constrained by cognitive biases, limited processing capacity, and delayed reaction times, ML systems offer the potential to systematically uncover patterns that may be invisible to human traders. In this thesis, I explore the premise that advanced time series prediction ML methods, through their capability to process large amounts of data, identify subtle patterns, and execute rapid decision-making, offer compelling advantages over traditional human expertise in time series prediction, particularly in the domain of financial trading.

1.2 Purpose and Scope

My central aim of this study is to extensively explore state-of-the-art ML methodologies for time series prediction and apply the most promising methods to day trading forecasting in the stock market. This work is structured as a manuscript-style thesis comprising two major studies.

The first paper, *A Survey of Machine Learning Methods for Time Series Prediction*, investigates the landscape of ML techniques applied to time series forecasting. In it, I seek to identify which models offer the most promise across different forecasting formats and tasks. Unlike many existing surveys, this study narrows its focus to research that compares state-of-the-art models within the same experimental setup, allowing for more meaningful insights into their relative strengths, weaknesses, and situational advantages.

The second paper, *Can AI Beat Human Traders? Exploring Machine Learning in Day Trading*, applies the insights gained from the survey to a real-world, trading problem. In it, I implement and evaluate a ML model designed to replicate and surpass human day trading strategies by using second-by-second trade and quote data from all U.S. equities over a two-year period.

1.3 Contributions of This Work

This thesis contributes to ML literature in time series forecasting through both a comprehensive review and practical application. The survey of time series prediction addresses a notable gap in the existing literature. Previous surveys face limitations in comparative analysis because they analyze independent studies which each utilize a different implementation and dataset. This heterogeneity prevents apples to apples comparison. My initial research indicated the superior performance of two classes of ML models: tree-based machine learning (TBML) and deep learning (DL) methods. Consequently, this study only reviews research that compares at least one state-of-the-art TBML and DL methods within the same uniform experimental settings so that nuanced conclusions can be drawn. These conclusions include investigating in key factors that influence model performance, such as the type of time series task, dataset size, time

interval of historical data, potential biases in model development, and trade-offs between computational costs and performance.

Key findings of this survey show that specialized TBML architectures like LightGBM and recurrent neural network DL architectures like long short-term memory models consistently outperform other alternatives, with TBML methods showcasing a notable advantage in computational efficiency. Additionally, this study emphasizes that the quality of data is the most influential factor affecting model performance, overshadowing the incremental benefits of hyperparameter tuning. Thus, considerable emphasis is placed on feature selection and engineering in the second paper. Similarly, this study found that combining models and leveraging diverse sources of information further boosts forecasting performance, an insight that heavily informed the construction of the trading model in the second paper.

The second paper applies insights from the survey to develop and evaluate ML models specifically for a day trading application. Day trading refers to the short-term buying and selling of financial instruments within a single trading day. Day trading is a risky endeavor and necessitates rapid decision-making and high-dimensional data processing capabilities. Only the top performing day trading humans actually make a profit, and thus in this paper I create a model that mimics the strategies used by human day traders while leveraging the unique advantages of ML processing with the goal of surpassing the human trading benchmark.

Existing research in this area typically relies on models tailored to individual stocks, limited to longer time horizons, and often lacking in feature diversity. In contrast, human day traders face different constraints as they are prone to emotional decision-making, have slower processing speeds, and can only monitor a few stocks at a time. The ML approach presented in this study overcomes these limitations by simultaneously analyzing the entire universe of U.S. equities and can manage many positions at one time. This study advances the current literature by addressing key gaps in ML applications, leveraging a combination of models trained on second-by-second trade and quote data, and operating on a comprehensive, high-dimensional dataset.

In this paper, I conduct a series of experiments to test various regularization techniques, risk-reward horizon weighting strategies, and execution pricing methods. I evaluate models using both idealized close prices and realistic execution prices derived from the bid and ask prices at which investors are willing to transact, with both methods resulting in profitable models. The model that performed best under realistic trading environments incorporated a min-max regularized target feature based on a risk-reward ratio and applied an equal weighting scheme in this risk-reward calculation. The results of this model show an average profit of 20,000 bps per day with a Sharpe ratio of 15.78 across an average of 999 trades per day. This performance is driven by the model's ability to simultaneously manage a large number of positions at the same time.

The findings from this research show that the ML model I developed achieves performance far exceeding the profitability of top human day traders, demonstrating average daily returns more than 500 times higher. Additionally, the methodological contributions provide a comprehensive, practical blueprint for high-frequency data acquisition, preprocessing, and feature engineering, facilitating replication and extension of the study by future researchers.

1.4 Structure of the Thesis

The remainder of this thesis is organized as follows: Chapter 2 presents a detailed survey on ML methods for time series prediction, identifying best practices and critical factors that influence model effectiveness. Chapter 3 introduces the empirical research of applying these methodologies to the challenge of day trading, detailing data methodologies, model development, and performance evaluation. Finally, Chapter 4 concludes with a summary of the contributions and recommendations for continued research in this area.

CHAPTER 2

A Survey of Machine Learning Methods for Time Series Prediction[1]

[1] Timothy Hall. Submitted to *Applied Sciences, 4/15/25*

Abstract

This study provides a comprehensive survey of the top-performing research papers in the field of time series prediction, offering insights into the most effective machine learning techniques, including tree-based, deep learning, and hybrid methods. It explores key factors influencing model performance, such as the type of time series task, dataset size, and the time interval of historical data. Additionally, the study investigates potential biases in model development and weighs the trade-offs between computational costs and performance. A detailed analysis of the most used error metrics and hyperparameter tuning methods in the reviewed papers is included. Furthermore, the study evaluates the results from prominent forecasting competitions, such as M5 and M6, to enrich the analysis. The findings of this paper highlight that tree-based methods like LightGBM and deep learning methods like recurrent neural networks deliver the best performance in time series forecasting, with tree-based methods offering a significant advantage in terms of computational efficiency. The paper concludes with practical recommendations for approaching time series forecasting tasks, offering valuable insights and actionable strategies that can enhance the accuracy and reliability of predictions derived from time series data.

2.1 Introduction

Time Series Prediction is the process of predicting a future value based on historical sequential observations. Accurate predictions based on time series data plays a crucial role in a wide range of domains where forecasting future values is essential for strategic-planning, resource management, and decision-making. Applications of time series prediction span numerous fields, including electricity consumption forecasting, environmental quality assessments (e.g., air and water quality), meteorological predictions (e.g., rainfall, solar radiation, and wind patterns), medical diagnostics (e.g., forecasting COVID-19 case trends and pneumonia incidences), traffic flow prediction, and financial domains like sales forecasting and stock market analysis.

In recent years, models based on Machine Learning (ML) have demonstrated the most success in time series forecasting and are able to generalize well to unseen data, unlike models based solely on probability and statistics. Specifically, Tree-Based Machine Learning (TBML) and Deep Learning (DL) have emerged as the most prominent approaches, excelling in scenarios where complex, nonlinear dependencies exist within the data. Their ability to generalize to unseen data makes them highly applicable to real-world problems with diverse and dynamic characteristics.

While numerous studies have examined these techniques within specific domains, and several survey papers (Lim & Zohren, 2021; Z. Liu et al., 2021) have analyzed various approaches to time series prediction across domains, existing literature reviews face a significant limitation. Current survey papers struggle to draw meaningful comparisons between models because they analyze independent studies, each utilizing different implementations and datasets. This heterogeneity in experimental setups prevents direct model comparisons and obscures true performance differences. This paper addresses this gap by exclusively reviewing studies that compare both TBML methods and DL approaches within the same experimental framework. By focusing on research where both methodologies are implemented and evaluated by the same researchers using identical datasets, this survey enables more robust conclusions about the relative strengths and weaknesses of these modeling approaches.

The remainder of this paper is organized as follows: **Section 2.2** outlines the methodology employed in this survey. **Section 2.3** reviews TBML architectures, while **Section 2.4** examines DL architectures. **Section 2.5** presents experimental results and discusses findings. **Section 2.6** highlights recent time series prediction competitions, and **Section 2.7** concludes the paper.

2.2 Methodology

A rigorous and systematic methodology was employed to identify the most relevant research papers for this survey. Web of Science was selected as the primary database because it is a trusted publisher-independent source of data with comprehensive coverage of peer-reviewed scientific literature. Given the objective of

comparing TBML methods with DL approaches in time series prediction, this paper establishes specific inclusion criteria for article selection:

1. **Focus on Time Series Applications:** The research must address problems involving time series data.
2. **Utilization of Advanced TBML Methods:** Studies must implement advanced TBML architectures, particularly gradient-boosted decision trees or similar structures (e.g., XGBoost, LightGBM, or CatBoost).
3. **Utilization of Advanced Neural Network (NN) Architectures:** Papers must explore sophisticated NN architectures, including but not limited to recurrent neural networks (RNN), feedforward neural networks (FFNN), convolutional neural networks (CNN), long short-term memory networks (LSTM), gated recurrent units (GRU), or Transformers.
4. **Direct Comparisons Using Identical Datasets:** The research must present comparative evaluations of at least one TBML and one DL architecture under identical experimental setups, ensuring consistent datasets and conditions.

To identify relevant literature, a comprehensive search query was developed as follows:

(ALL=(XGBoost) OR ALL=(LightGBM) OR ALL=(CatBoost) OR ALL=("gradient boost*")) AND (ALL=("time series") OR ALL=("time sequence") OR ALL=("temporal series") OR ALL=("temporal sequence") OR ALL=("time forecast*")) AND (ALL=("LSTM") OR ALL=("ANN") OR ALL=("CNN") OR ALL=("RNN") OR ALL=("transformer") OR ALL=("GRU") OR ALL=("deep neur*") OR ALL=("deep lear*"))

This query yielded 589 papers published between 2017 and 2024. To ensure focus on the most influential research, papers were initially selected based on citation count. From the top 100 most-cited papers, 65 articles satisfied the inclusion criteria. To maintain contemporary relevance and capture recent

developments in the field, additional temporal criteria were implemented: a minimum of 10 papers per year from 2020 to 2024 must be included. Consequently, 4 highly cited papers from 2023 and the 10 most-cited articles from 2024 were incorporated. In total, this survey encompasses 79 influential papers investigating the comparative performance of TBML and DL approaches in time series analysis (see Appendix A for the complete list of included studies).

2.3 Tree-Based Machine Learning Architectures

This section will present an overview of the best performing TBML Architectures, which are widely utilized for both regression and classification tasks. These include Random Forests (RF), Gradient Boosted Decision Trees (GBDT), and three prominent implementations of GBDT: XGBoost, LightGBM, and CatBoost. Figure 2.1 provides a comparative visualization of the structural differences between RF and GBDT.

2.3.1 Random Forests

Random Forests (RF) is an ensemble learning method that constructs multiple decision trees and combines their outputs through averaging for regression tasks or majority voting for classification tasks. RF uses bootstrapping to train individual decision trees on a random subset of the data at each split. This randomization, coupled with its ensemble nature, enhances the robustness of RF compared to single decision trees, significantly reducing the risk of overfitting. The most widely used library for RF implementation in the studies reviewed in this paper is Scikit-learn.

2.3.2 Gradient Boosted Decision Trees

Gradient Boosted Decision Trees (GBDTs) is a machine learning algorithm that aggregates predictions from multiple weak learners, typically decision trees. GBDTs use “boosting” to build models iteratively where each subsequent models focuses on correcting mistakes made by previous models. The algorithm achieves this by optimizing a differentiable loss function using gradient descent. GBDTs are suitable for both classification tasks, using loss functions such as cross-entropy, and regression tasks, using loss

functions like mean squared error. Key hyperparameters to tune in GBDT include tree depth and learning rate, which are crucial for balancing model complexity and reducing overfitting. While Scikit-learn provides a general implementation of GBDT, the three most prominent and high-performing implementations discussed in this survey are XGBoost, LightGBM, and CatBoost, each offering distinct advantages.

2.3.2.1 XGBoost

XGBoost, eXtreme Gradient Boosting (Chen & Guestrin, 2016), introduced by Tianqi Chen in 2014, was designed to address key limitations of traditional GBDT, particularly computational efficiency and scalability. XGBoost gained immediate popularity due to its significant speed improvements, achieved through innovative approaches in decision tree construction. Unlike the greedy splitting methods used in standard GBDT, XGBoost employs a similarity score to evaluate potential splits. This score measures the homogeneity of observations within a node relative to the target variable, assessing the gain provided by a split. To further reduce overfitting, XGBoost incorporates several techniques, including pruning, where branches with a gain below a threshold (hyperparameter gamma, γ) are removed. Similarly, XGBoost has several regularization techniques that prevent overfitting by penalizing complex decision trees. XGBoost also supports advanced features such as built-in cross-validation and highly scalable parallel processing, making it ideal for large-scale datasets.

2.3.2.2 LightGBM

LightGBM (G. Ke et al., 2017), developed by Microsoft in 2017, shares many foundational principles with XGBoost with an even larger focus on computational efficiency. LightGBM achieves superior speed by employing histogram-based binning, which discretizes continuous features into bins, trading minor accuracy losses for dramatic speed gains. Additionally, LightGBM introduces Exclusive Feature Bundling, which is particularly effective for many real-world datasets with high-dimensional sparse features because it consolidates mutually exclusive features into a single representation. Another innovation is Gradient-

Based One-Side Sampling, which prioritizes instances with large gradients while randomly sampling smaller gradients, optimizing learning efficiency. LightGBM also utilizes a leaf-wise tree growth strategy, as opposed to the level-wise growth used in traditional GBDT and XGBoost. This approach selectively grows the leaf with the greatest potential to improve the model, enabling faster convergence and improved accuracy.

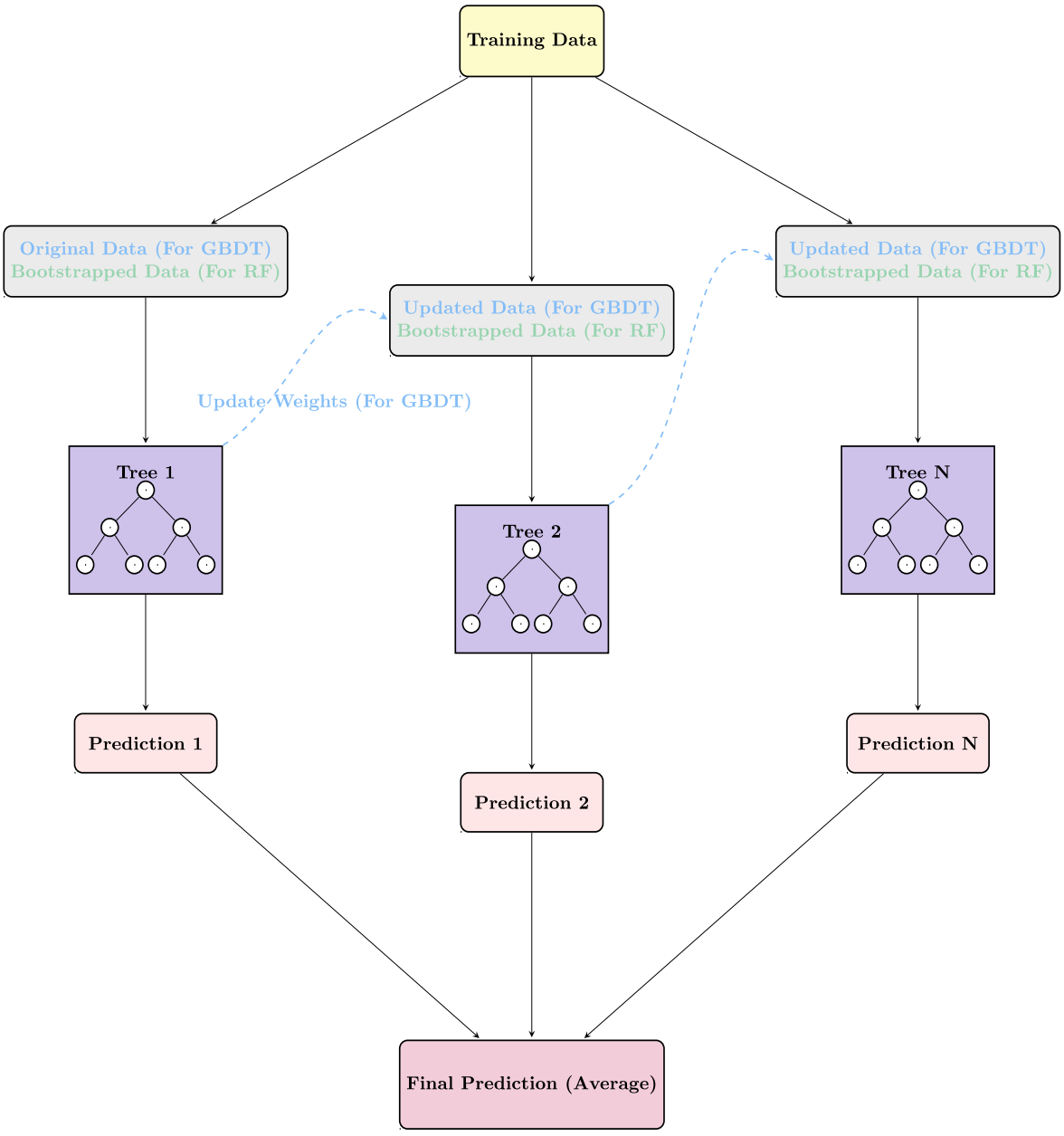

Figure 2.1: Illustrates the architectural differences between RF and GBDT, highlighting attributes unique to RF in green and those specific to GBDT in blue.

2.3.2.3 CatBoost

CatBoost (Prokhorenkova et al., 2018), developed by Yandex in 2017, was designed with a specific emphasis on handling datasets with categorical features. One of its key innovations is a unique implementation of target encoding which takes the concept of traditional target encoding, where categorical values are replaced with the mean of the target variable for each category, and instead constructs the encoding process using only previous data to avoid data leakage. Another distinguishing feature of CatBoost is its use of symmetric decision trees, where all leaves at the same depth use identical splitting criteria. This structure not only accelerates training but also significantly reduces inference time, an essential advantage in some time-sensitive applications of real-world forecasting.

2.4 Deep Learning Architectures

This section presents an overview of the most prominent Deep Learning (DL) architectures encountered in the surveyed literature. DL will be used in this paper to describe a subset of machine learning that utilizes neural networks to perform classification and regression tasks. The architectures are categorized into four primary groups: Feed-Forward Neural Networks (FFNN), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Attention-based architectures. Figure 2.2 presents a visual representation of a FFNN, highlighting the architectural modifications required to transform it into a RNN or CNN.

2.4.1 Feed-Forward Neural Networks

The term Feed-Forward Neural Network (FFNN) is often used interchangeably with other terminologies in the literature, including Artificial Neural Network (ANN), Neural Network (NN), Multilayer Perceptron (MLP), Back Propagation Neural Network (BPNN) (Nan et al., 2022; Pan et al., 2023), Deep Neural Network (DNN), and Deep Feed-Forward Neural Network (DFFNN). These networks are characterized by

a unidirectional flow of information from input to output nodes, often traversing one or more hidden layers. Unlike other architectures, FFNNs do not contain loops or cycles within their structure[2].

Although the various terms for FFNN are used interchangeably in the surveyed literature, there are small architectural differences that exist between these variants namely in their depth. General FFNN models do not require any hidden layers, whereas MLP models typically contain one or two hidden layers. DNN usually incorporate multiple hidden layers, allowing them to capture more complex patterns in the data. Despite their widespread use in time series analysis, these architectures require careful feature engineering to incorporate temporal information effectively and struggle to learn long-term patterns within the data. These limitations led to the development of more specialized architectures including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

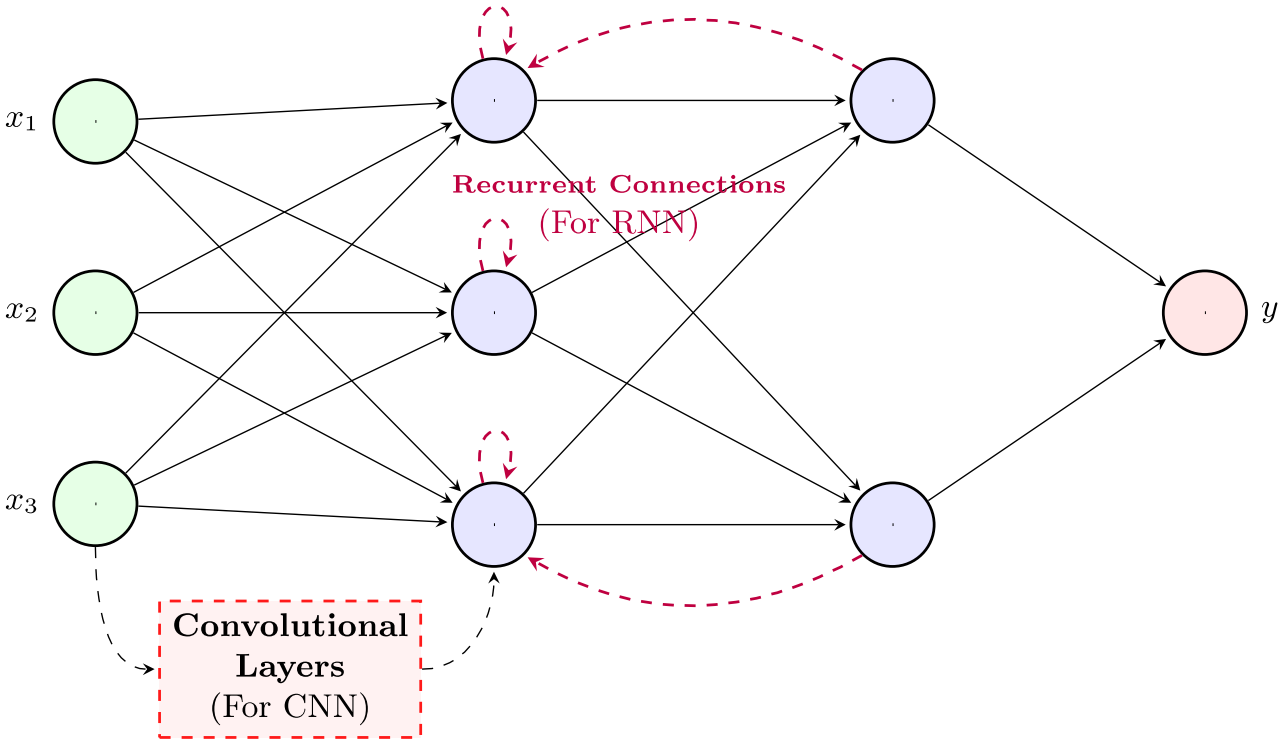

Figure 2.2: Illustrates the architecture of a FFNN, highlighting the modifications needed to transform it into a CNN (in orange) or a RNN (in red).

[2] For a comprehensive overview of the implementation details and mathematical foundation of FFNNs, see Svozil, Kvasnicka and Pospichal, 1997

2.4.1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a type of FFNN that distinguish themselves in their architectural design through convolutional layers. The biggest benefit of these convolutional layers is to systematically scan input data to extract various features[3] making them well suited candidates for handling image data as they can detect local features, combine them into complex patterns, and still maintain spatial relationships between features. Although this can be useful in certain datasets which contain a time series of images, CNN models lack the ability to remember information over longer periods of time, such as Recurrent Neural Network (RNN) models and Transformer models.

2.4.2 Recurrent Neural Networks

Recurrent Neural Networks (RNN) distinguish themselves through their unique feedback loop architecture, where outputs from previous timesteps are fed back into the network, creating an internal memory state that enables learning from past inputs. This recursive processing allows RNN models to incorporate historical information into current computations, making them particularly well-suited for time series analysis, where current predictions often depend heavily on historical patterns.

Among RNN variants, Long Short-Term Memory (LSTM) networks have emerged as the most popular in literature studied for this paper, largely due to their sophisticated gating mechanisms that help mitigate the vanishing gradient problem prevalent in general RNNs. LSTM models incorporate an input, forget, and output gate to regulate information flow and enable the network to capture both long-term dependencies and short-term patterns effectively (Sherstinsky, 2020). Variants such as Bidirectional LSTMs (Bi-LSTMs), which process sequences in both forward and backward directions, enhance the model's ability to leverage contextual information from both past and future timesteps. Another notable RNN variant is the Gated Recurrent Unit (GRU), which simplifies LSTM architecture by combining the input and forget gates into a

[3] For a comprehensive examination of CNN architectures, including detailed mathematical formulations and implementation considerations, see Gu, et al., 2018

single update gate and introducing a reset gate. GRUs are computationally more efficient than LSTMs while retaining comparable performance in capturing sequence dependencies.

2.4.3 Attention-Based Architectures

Attention-Based Architectures, particularly Transformer Models (Vaswani et al., 2017), are the least explored deep learning approach among the surveyed literature in time series applications. Unlike RNNs, which use recurrent layers to process data sequentially, Transformers use a self-attention mechanism to identify ways that distant data features in a series are related to each other. They operate by processing entire sequences in parallel, making them much faster to train than RNNs. The application of Transformers has mainly been used for Natural Language Processing (NLP) models and less so for time series applications. Among the surveyed literature, only five papers (Guan et al., 2024; López Santos et al., 2022; Torres, Martínez-Álvarez, & Troncoso, 2022; Zhao et al., 2024; Zrira et al., 2024) explore attention-based models in this domain. Early results suggest that Transformers can outperform traditional RNN-based models in tasks requiring long-term dependency modeling, though their suitability for specific time series tasks warrants further investigation.

2.5 Experimental Results and Discussion

This section outlines the methodology used to conduct the analysis presented in this survey. The process begins with a detailed discussion of data preprocessing in **Section 5.1**, where the inclusion and exclusion criteria for models are established. **Section 5.2** describes the evaluation metrics employed to systematically compare the performance of different models. The results of the analysis are presented in **Section 5.3**, followed by an in-depth discussion of these findings in **Section 5.4**.

2.5.1 Data Preprocessing

The primary objective of this survey is to evaluate and compare the performance of DL models against TBML methods for time series prediction. To maintain the integrity and relevance of the analysis, certain

models were excluded based on their consistent underperformance or lack of applicability. Linear regression (LR) models, for example, were omitted due to their poor performance across most surveyed studies, apart from a single instance of Ridge Regression that showed promising results (Yu et al., 2020). While this exception highlights potential avenues for future research, it should be noted that other Ridge Regression examples in the surveyed literature performed poorly (Chen, Guan, & Li, 2021). Decision Tree (DT) models were also excluded, as they consistently underperformed relative to other methods.

Autoregressive Integrated Moving Average (ARIMA) models, including variants like Seasonal ARIMA (SARIMA), have historically been popular for time series forecasting. However, all surveyed instances demonstrated that these models were outperformed by either DL or TBML methods. Consequently, ARIMA-based models were excluded from this comparative analysis (Hewamalage, Bergmeir, & Bandara, 2022; Ibañez et al., 2022; Khan et al., 2020; López Santos et al., 2022; Nan et al., 2022; Priyadarshi et al., 2019; Rafi et al., 2021; Ribeiro et al., 2022; Shi, He, & Liu, 2018; Yang et al., 2020; Zhu et al., 2023). Adaboost, a tree-based ensemble algorithm developed in 1995, was similarly excluded due to its inferior performance when compared to similar modern TBML methods such as XGBoost, LightGBM, and CatBoost. In all reviewed cases, Adaboost either underperformed or matched the performance of these more advanced methods (Comert et al., 2021; Farsi, 2021; G. Li et al., 2022; Pavlov-Kagadejev et al., 2024).

For this analysis, models were grouped into broader algorithmic categories to facilitate meaningful comparisons. Radial Basis Function Neural Network (RBFN) (Rafi et al., 2021), Neural Network Autoregression (NNAR) (Liang et al., 2021), and Broad Learning System (BLS) (G. Li et al., 2022) models were grouped with Feed-Forward Neural Networks (FFNNs), which also encompass MLP, ANN, and DNN architectures. While Convolutional Neural Networks (CNNs) are technically FFNNs, their unique operational characteristics justify separate analysis. Recurrent Neural Network (RNN) architectures, including GRU, LSTM, and Bi-LSTM, were grouped alongside DeepAR, an autoregressive recurrent neural network (Singh et al., 2024). The TBML category encompasses XGBoost, LightGBM, CatBoost, Bagging, GBDT, and RF models.

2.5.2 Evaluation Metrics

To systematically compare the performance of various models surveyed in the reviewed papers, two evaluation metrics were employed. These metrics were designed to capture different aspects of model performance. The first metric, First Place Aggregation (FPA), measures the frequency with which a model is identified as the highest-performing or tied for the highest-performing model in the surveyed studies. FPA is calculated as:

$$\mathrm{FPA} = \frac{N_{\mathrm{first}}}{N_{\mathrm{total}}} \times 100$$

Here, N_{first} represents the number of times the model achieved the top rank, while N_{total} is the total number of evaluations in which the model was included. FPA provides a straightforward measure of a model's dominance in performance comparisons. A key advantage of this metric is that it mitigates the influence of any poor performing models present within comparisons.

The second metric utilizes a weighted rank aggregation (WRA) approach to account for the relative performance of a model across all comparisons, not just first-place rankings. The WRA score is calculated as:

$$\mathrm{WRA} = 1 - \frac{(N_{\mathrm{rank}} - 1)}{(N_{\mathrm{total}} - 1)}$$

Here, N_{rank} denotes the model's rank in each comparison, with 1 representing the top rank. N_{total} is the total number of models in the comparison. This metric assigns a score of 1 to the top-performing model and 0 to the lowest-performing model, offering a nuanced perspective that accounts for varying numbers of models across comparisons. WRA is particularly valuable in scenarios where the relative performance of models varies significantly, enabling a more comprehensive evaluation of their effectiveness.

2.5.3 Results

2.5.3.1 Overall Model Performance

The first objective of this paper is to identify the models that exhibit the best overall performance across a variety of time series prediction tasks analyzed in the 79 surveyed studies. The models can broadly be categorized into two classes: Tree-Based Machine Learning (TBML) models and Deep Learning (DL) models. Figure 2.3 illustrates the comparative performance of these classes based on the First Place Aggregation (FPA) and Weighted Rank Aggregation (WRA) metrics.

TBML models marginally outperform DL models as a class, achieving the best performance (FPA) in 54.55% of tasks studied, with a WRA score of 0.6910. In comparison, DL models perform best (FPA) in 52.70% of tasks and achieve a WRA score of 0.6486. It is important to note that ties for the highest-performing model contribute to a combined percentage exceeding 100%. On average, TBML models outperform DL models by approximately 2.5%. However, this comparison at the class level provides only

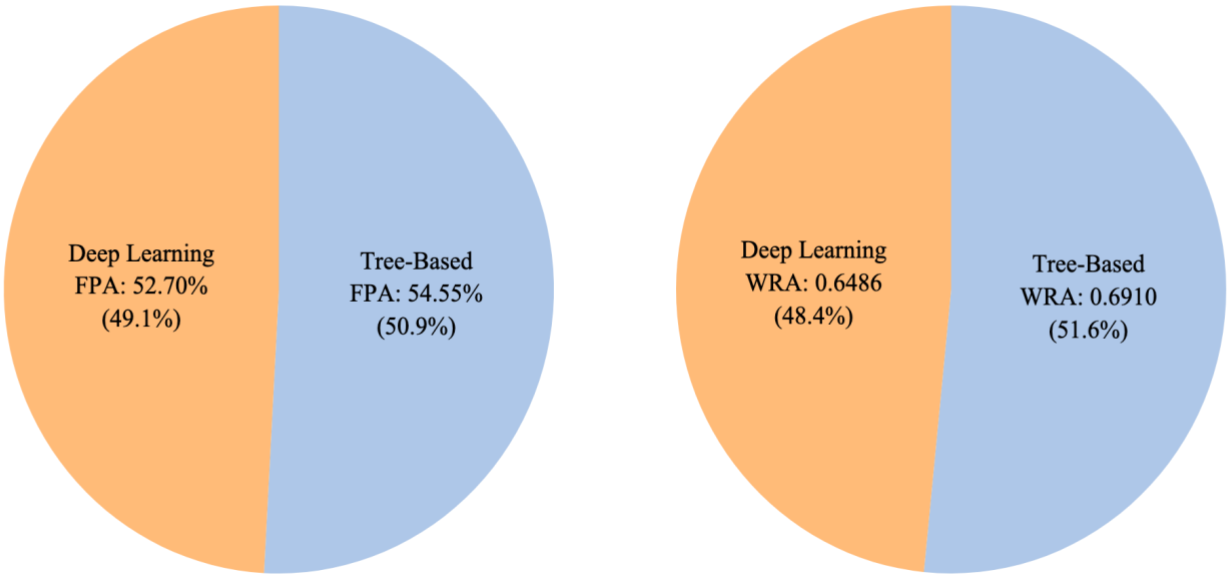

Figure 2.3: Illustrates the FPA and WRA score distribution for each class, along with the comparative performance (%) between the two classes.

a surface-level understanding of the performance landscape. A more granular analysis is needed to assess the performance of subclasses within each category.

Figure 2.4 breaks down the performance of model subclasses within the TBML and DL categories for FPA and WRA metrics. Notably, the “Attention” subclass, which exclusively comprises Transformer models, emerges as the best-performing model in three tasks but is only evaluated in five studies. While this result

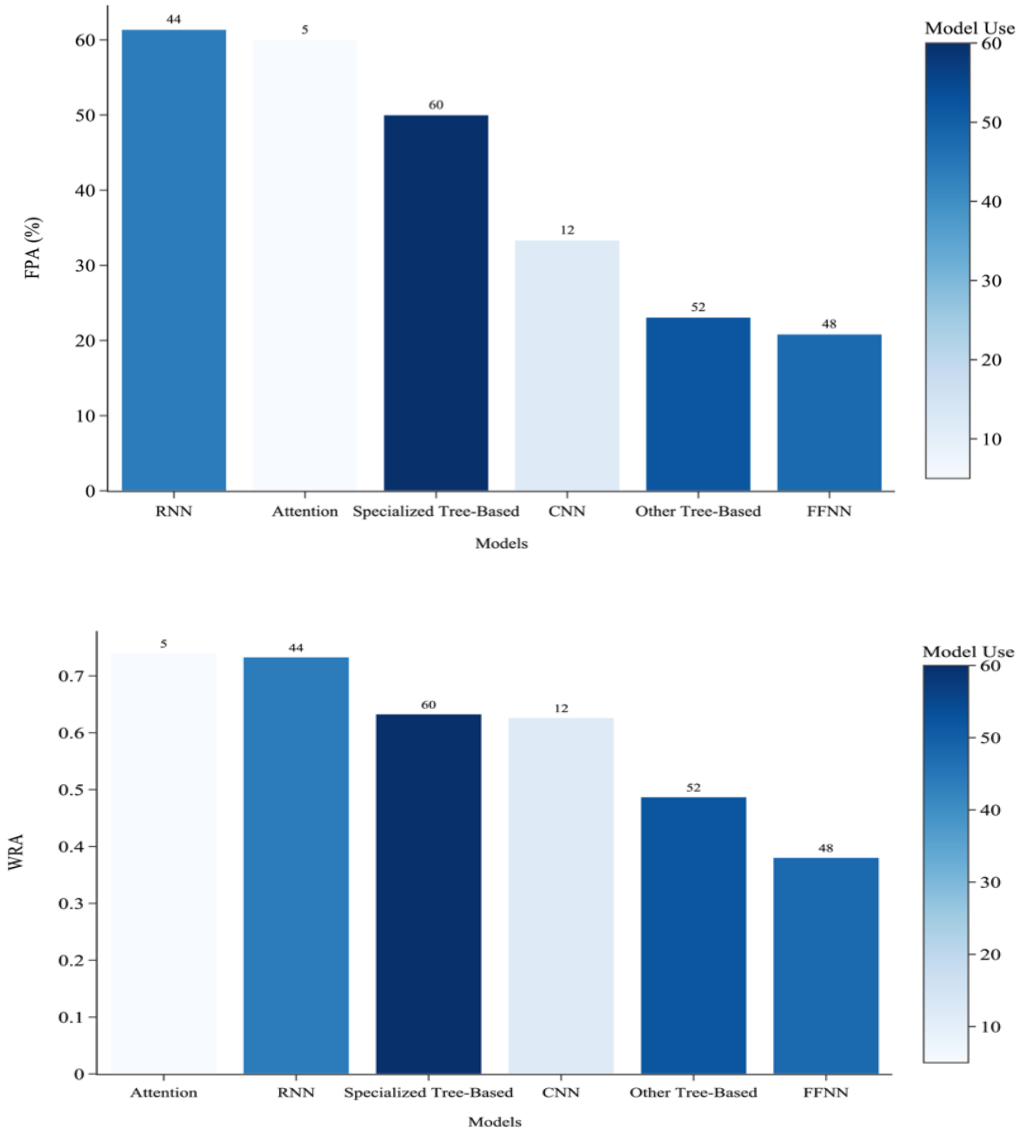

Figure 2.4: Depicts the FPA and WRA scores for each model. The "Model Use" legend indicates the total number of comparisons for each model, with specific counts labeled above the bars.

is intriguing and suggests a promising avenue for further research, the limited sample size warrants caution when interpreting these findings. Consequently, this paper does not place a primary focus on Attention-based models.

From the detailed subclass analysis, it is evident that the most effective algorithms for time series prediction are Recurrent Neural Network (RNN) models, primarily comprised of LSTM, GRU, and general RNN architectures, and the Specialized Tree-Based Models (SPTB) category, which includes XGBoost, LightGBM, and CatBoost. RNNs demonstrate superior performance, ranking as the best-performing models in 61.36% of studies and achieving a WRA score of 0.7330. They are closely followed by SPTB models, which perform best in 50% of studies and achieve a WRA score of 0.6328. These results underscore the complexity of the performance landscape and highlight that it is insufficient to simply state that TBML models outperform DL models in time series prediction tasks. Instead, a nuanced understanding of individual subclasses helps to draw more meaningful conclusions.

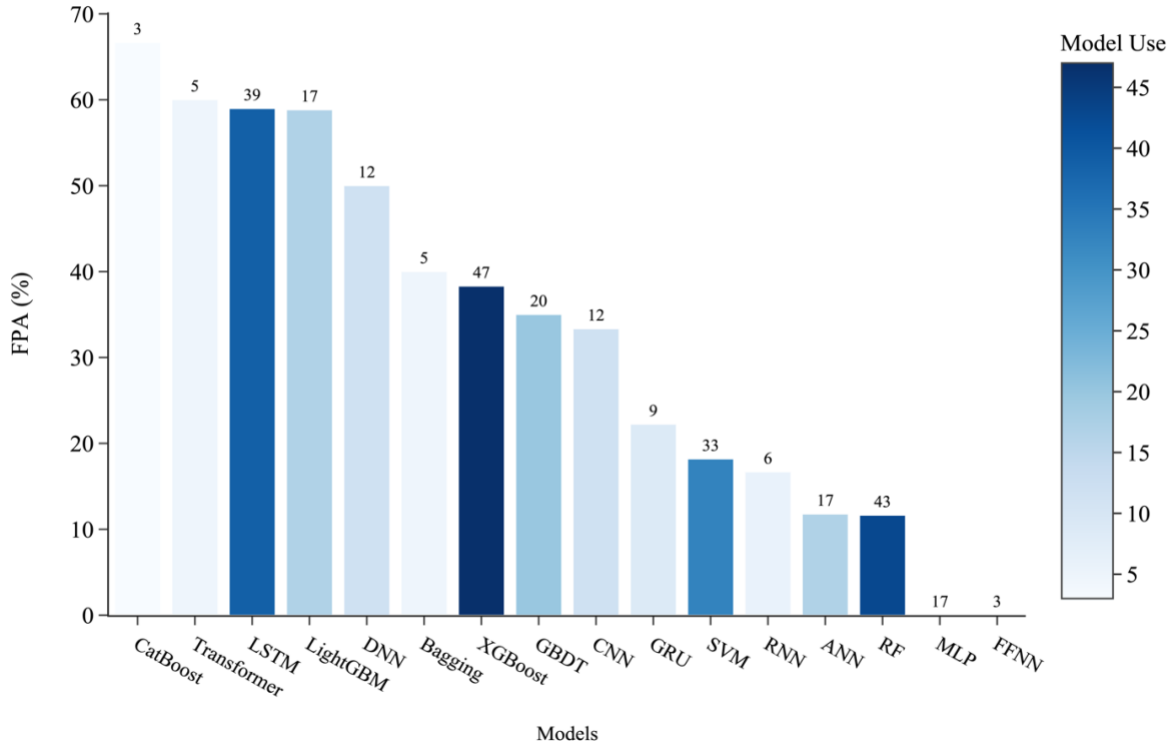

Figure 2.5: Depicts the FPA scores for each model. The "Model Use" legend indicates the total number of comparisons for each model, with specific counts labeled above the bars.

To further refine this understanding, the subclasses are broken down into individual models. Figures 2.5 and 2.6 present the performance of each model based on the FPA and WRA metrics, respectively. Among individual models, CatBoost emerges as the best performer across both metrics, followed by Transformers, LSTMs, and LightGBM. However, it is important to note that like Attention-Based Transformer models, the robustness of CatBoost's results may be limited, as it was the best-performing model in only two out of the three studies where it was evaluated. The LSTM model, on the other hand, demonstrates strong and consistent performance, achieving an FPA of 58.97% and a WRA score of 0.7222. With 39 instances of evaluation, its results are more statistically robust. LightGBM also performs well, with an FPA of 58.82% and a WRA score of 0.6608, based on 17 studies. In contrast, the two weakest-performing models were MLPs and general FFNNs. Neither model achieved the highest rank in any study, and their WRA scores, 0.2265 for MLP and 0.1667 for FFNN, were the lowest among all evaluated models.

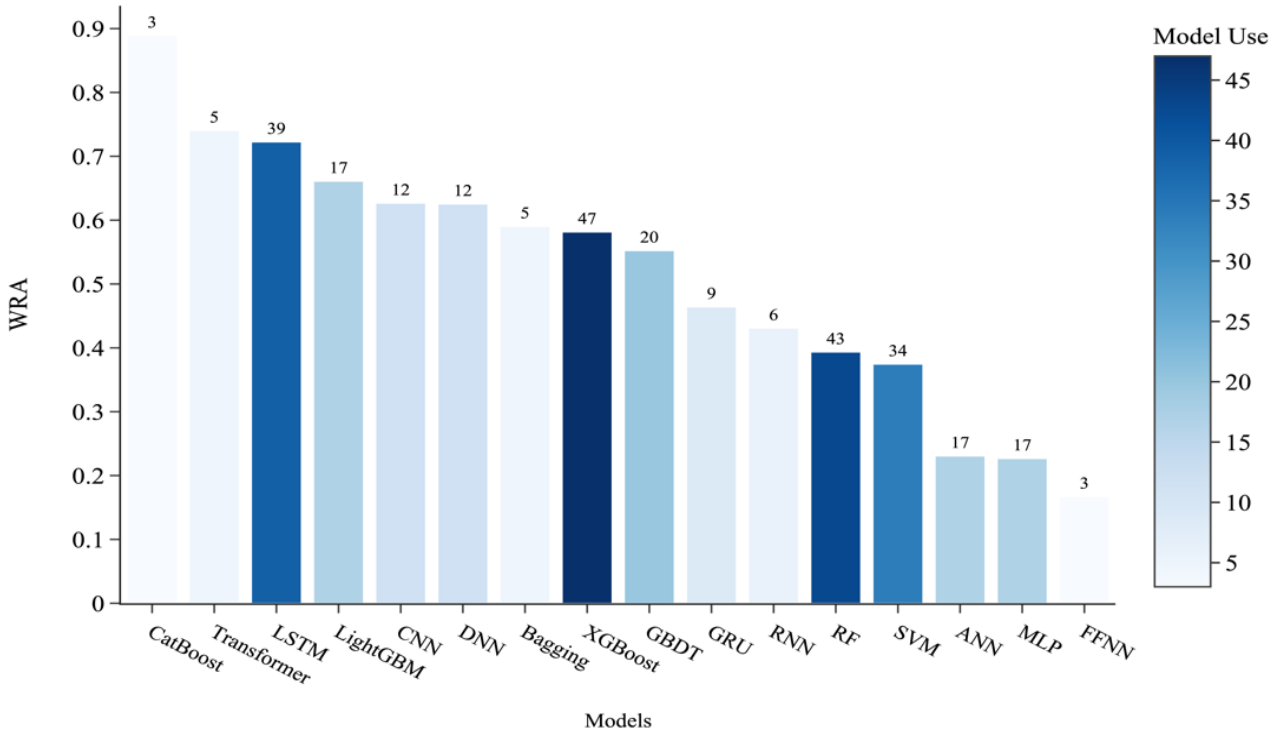

Figure 2.6: Depicts the WRA scores for each model. The "Model Use" legend indicates the total number of comparisons for each model, with specific counts labeled above the bars.

Additionally, it is notable that the most frequently compared models in the surveyed papers were XGBoost, RF, and LSTM, with 47, 43, and 39 instances, respectively.

2.5.3.2 Task-Specific Model Performance Analysis

Another objective of this paper is to evaluate whether certain models exhibit superior performance for specific types of time series prediction tasks. Table 2.1 provides a detailed list of the 46 unique tasks covered in the surveyed papers. To facilitate meaningful comparisons, these tasks were grouped into 10 broader categories based on shared characteristics and application domains (Figure 2.7).

For each task category, the performance of TBML models and DL models was compared, with additional focus on the best-performing subclasses within these categories: RNNs and SPTB models. Figures 2.8 and 2.9 illustrate the comparative performance of these model classes using the FPA and WRA metrics, respectively. The results highlight that TBML models consistently outperform DL models in Task Groups 1, 5, 9, and 10.

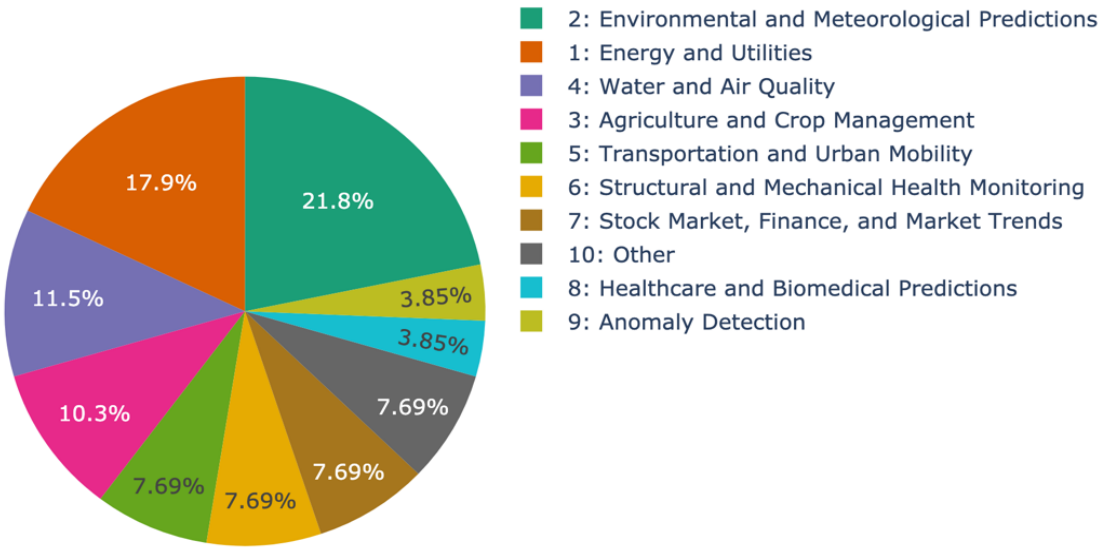

Figure 2.7: Illustrates the distribution of time series prediction tasks by category, including group numbers and the percentage representation of each category.

Table 2.1: Illustrates the 46 time series prediction tasks present in the surveyed papers. The frequency column shows the number of occurrences of the tasks, and the group column shows the assigned category number of each task

No.	Time Series Prediction Task	Frequency	Group
1	Total Electricity Consumption (Demand)	8	1
2	Load Forecasting	2	1
3	Electricity Theft Prediction	1	1
4	Heating Load Prediction	1	1
5	Return Temperature of District Heating System	1	1
6	Electricity Consumption of an Electric Bus	1	1
7	Solar Power Forecasting	3	2
8	Wind Power Forecasting	2	2
9	Rainfall Prediction (Including Rainfall Runoff)	2	2
10	Drought Prediction	2	2
11	River Inflow Prediction (Including Reclaimed Water Volumes)	2	2
12	Subsurface Temperature (Including Sea Surface Temperature)	2	2
13	Reservoir Water Level Prediction	1	2
14	Flood Frequency	1	2
15	Groundwater Availability	1	2
16	Indoor Daylight Illuminances Prediction	1	2
17	Crop Yield (Including Corn Biomass, Crop Height)	5	3
18	Crop Classification	3	3
19	Water Quality Prediction (Including Chlorophyll-a and Wastewater Treatment)	8	4
20	Air Quality	1	4
21	Passenger Demand (Includes Bike Sharing, Urban Rail Passenger Flow)	3	5
22	Travel Time Prediction	1	5
23	Future Traffic of Mobile Base Stations in Urban Areas	1	5
24	Traffic Queue Length	1	5
25	Tunnel Deformation Prediction	1	6
26	Dam Structural Health Prediction	1	6
27	Highway Tunnel Pavement Performance	1	6
28	Predict Temperature Trend of Wind Turbine Gearbox	1	6
29	Discharge Capacity Estimation for Li-Ion Batteries	1	6
30	Sintering Process Prediction	1	6
31	Stock Price (Including Crypto/Stock Trend)	3	7
32	Hedge Fund Return Prediction	1	7
33	Store Item Demand	1	7
34	Vegetables Demand	1	7
35	Post Stroke Pneumonia Prediction	1	8
36	Predict Peak Demand Days of Cardiovascular Admissions	1	8
37	COVID-19 New Cases Prediction	1	8
38	Anomaly Detection for Web Services	1	9
39	Leak Detection	1	9
40	Fall Detection	1	9
41	Global Models for Various Tasks (Simulated and Real World)	1	10
42	Predicting Emerging Research Topics	1	10
43	Lane Changing Risk	1	10
44	Predictive Process Monitoring	1	10
45	Oil Well Production	1	10
46	Crime Prediction	1	10

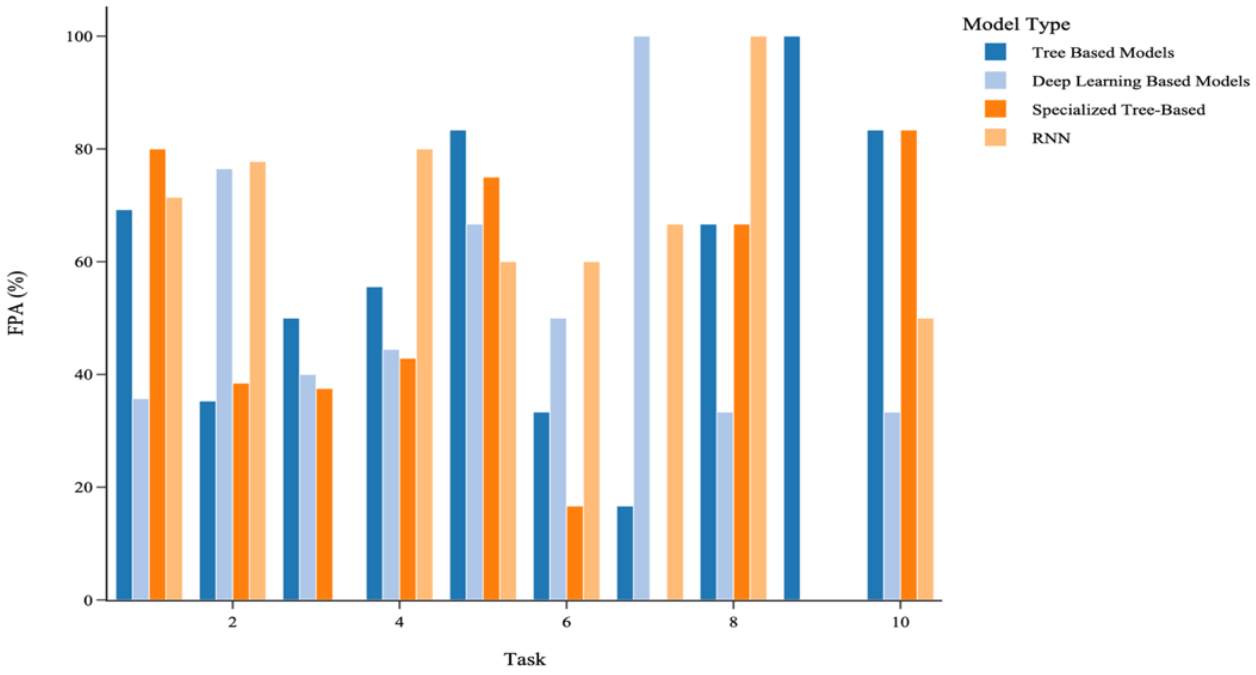

Figure 2.8: Illustrates the FPA scores for each model. Blue bars represent the overall model types, while orange bars highlight the best-performing model classes.

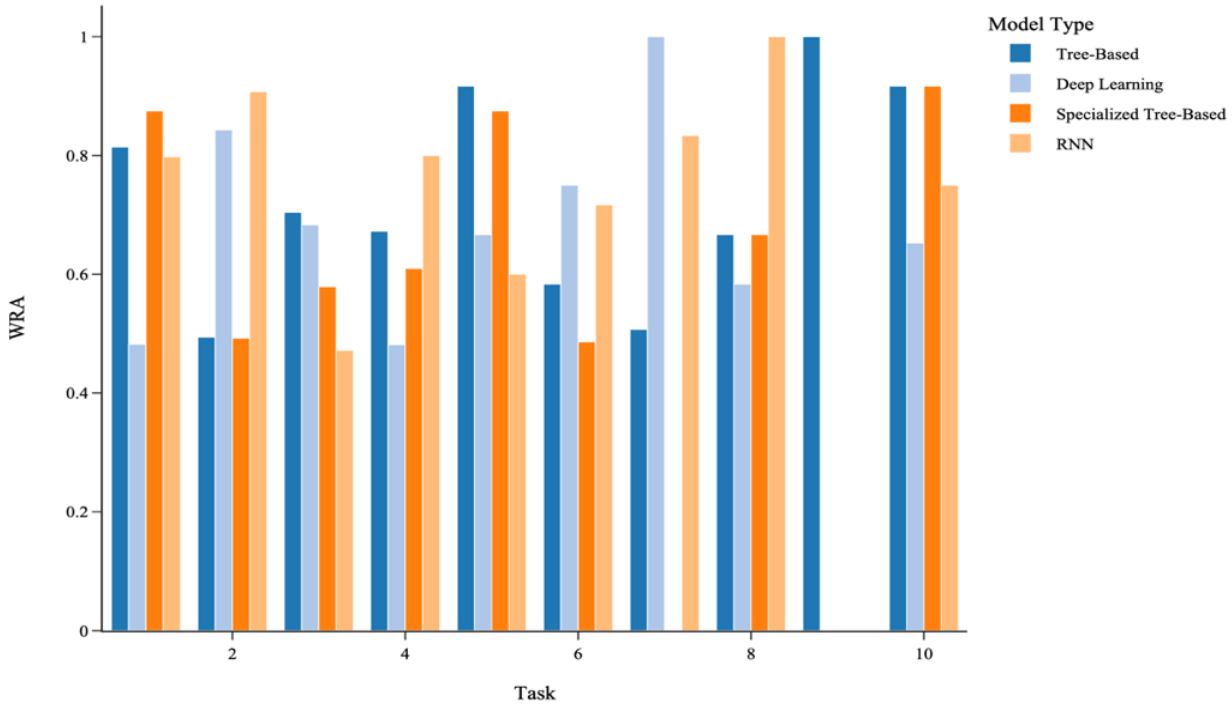

Figure 2.9: Illustrates the WRA scores for each model. Blue bars represent the overall model types, while orange bars highlight the best-performing model classes.

In these categories, TBML models demonstrate performance advantages ranging from 37.5% to 68.85% in WRA scores and 25% to 150% in FPA scores. Conversely, DL models show significant superiority in Task Groups 2, 6, and 7, with performance improvements of 28.57% to 97.26% in WRA and 50% to 500% in FPA

compared to TBML models. A more nuanced comparison between RNN and SPTB models reveals that SPTB models perform notably better than RNNs in Task Groups 5 and 10, with WRA improvements of 45.83% and 22.22% and FPA improvements of 25% and 66.67%, respectively compared to RNNs. In contrast, RNN models excel over SPTB models in Task Groups 2, 4, 6, 7, and 8, with performance gains compared to SPTB models ranging from 31.25% to 84.32% in WRA and 50% to 260% in FPA.

2.5.3.3 Impact of Dataset Size on Model Performance

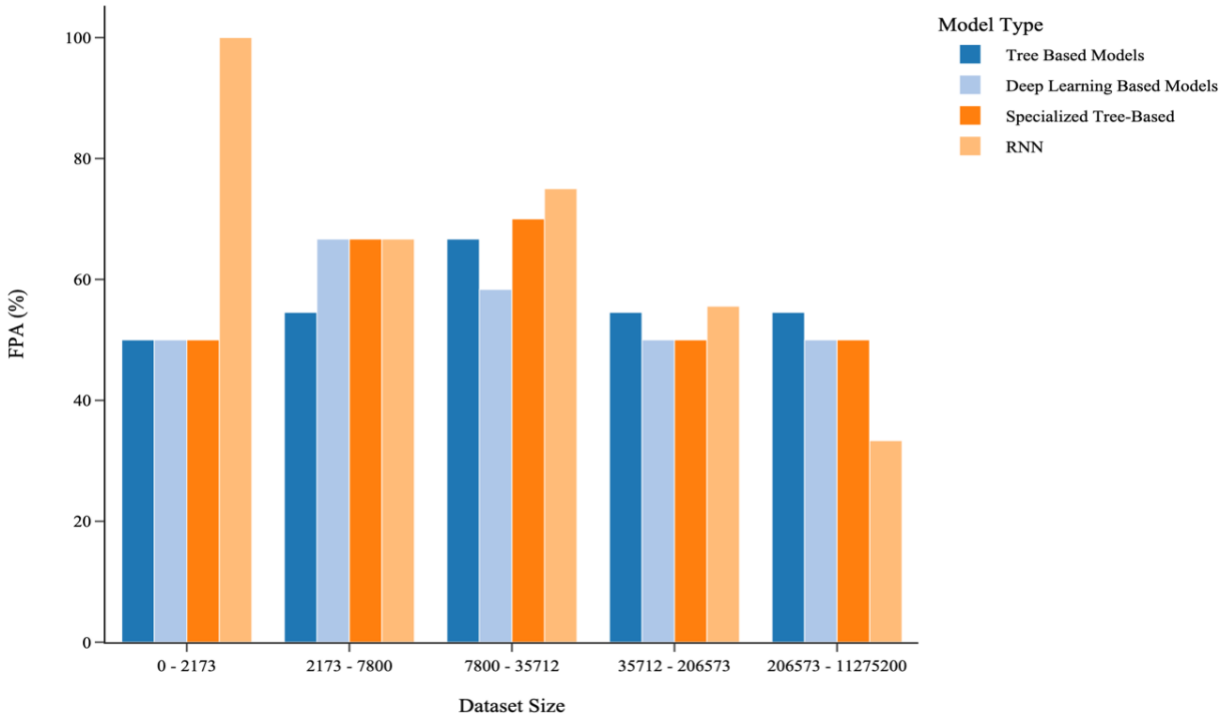

Figure 2.10: Illustrates the FPA percentage for each model based on dataset size. Blue bars represent the overall model types, while orange bars indicate the best-performing class of models. The dataset size range is exclusive on the left side and inclusive on the right side.

This study also examines how dataset size influences the relative performance of different machine learning models for time series prediction. To ensure balanced analysis, dataset ranges were selected to contain an equal number of model comparisons within each range. The analysis considers TBML models and DL models, as well as their top-performing subclasses: SPTB models and RNNs. Figures 2.10 and 2.11 present the performance of these model classes across the dataset ranges using FPA and WRA metrics, respectively.

In the smallest dataset range (0–2,173 samples), TBML models perform comparably to DL models overall, but RNN significantly outperform SPTB models, achieving an FPA advantage of 50% and a WRA advantage of 0.425. In the second range (2,173–7,800 samples), DL models demonstrate a slight edge over TBML models, with FPA and WRA gains of 12.12% and 0.0972, respectively. Within this range, RNN outperform SPTB models with a WRA gain of 0.1296, while the FPA scores for both subclasses are equal.

For mid-sized datasets (7,800–35,712 samples), TBML models begin to show a slight advantage over DL models, outperforming them by 8.33% in FPA and 0.0417 in WRA. However, RNN models continue

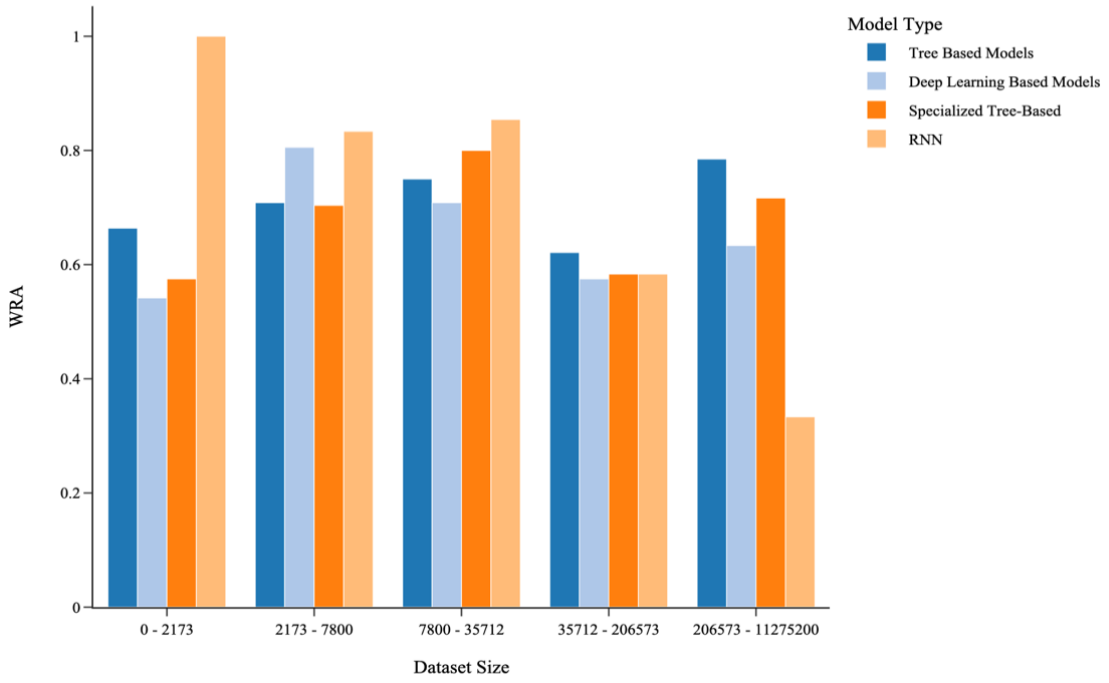

Figure 2.11: Illustrates the WRA percentage for each model based on dataset size. Blue bars represent the overall model types, while orange bars indicate the best-performing class of models. The dataset size range is exclusive on the left side and inclusive on the right side.

to outperform SPTB models, achieving an FPA gain of 5% and a WRA gain of 0.0542. In the second-largest dataset range (35,712–206,573 samples), TBML models maintain a modest advantage over DL models, with FPA and WRA improvements of 4.55% and 0.0462, respectively. Similarly, RNN outperform SPTB models by 0.0556 in FPA, while both achieve equivalent WRA scores. The largest dataset range (206,573–11,275,200 samples) reveals a more definitive trend. In this range, TBML models outperform DL models by 4.55% in FPA and 15.15 in WRA. Moreover, SPTB models achieve significant gains over RNN models, with an FPA advantage of 16.67% and a WRA advantage of 0.3833.

2.5.3.4 Impact of Data Time Interval on Model Performance

This study investigates whether the time interval of datasets influences the performance of machine learning models for time series prediction. As in previous analyses, the comparison includes TBML models and DL

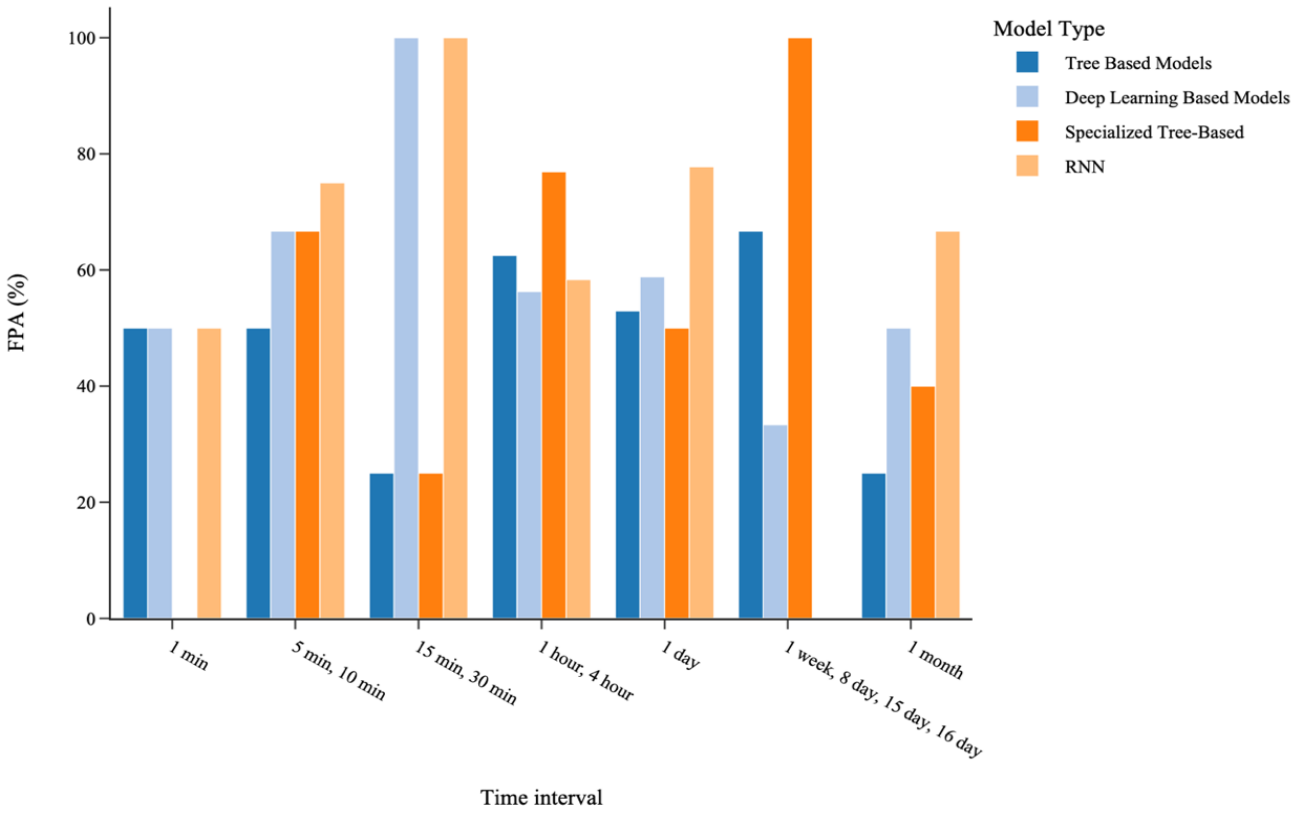

Figure 2.12: Illustrates the FPA score for each model based on time interval of data. Blue bars represent the overall model types, while orange bars indicate the best-performing class of models.

models, along with their top-performing subsets: SPTB models and RNNs. Figures 2.12 and 2.13 illustrate the FPA and WRA scores across different time intervals of the datasets.

DL models outperform TBML models in the 5/10-minute, 15/30-minute, and 1-month intervals. Notably, these intervals have a modest representation in the dataset, occurring 6, 4, and 8 times, respectively. Conversely, TBML models show slightly better performance in the 1/4-hour interval (16 occurrences) and the 7/8/15/16-day interval (3 occurrences[4]), with FPA and WRA advantages of 6.25% and 0.0469, and 33.33% and 0.0556, respectively.

A deeper comparison of subclasses reveals that RNN models outperform SPTB models in the 1-minute, 5/10-minute, 15/30-minute, 1-day, and 1-month time intervals. Among these, the 1-day interval (9 RNN occurrences vs. 12 SPTB occurrences) is particularly significant, where RNN models achieve a notable

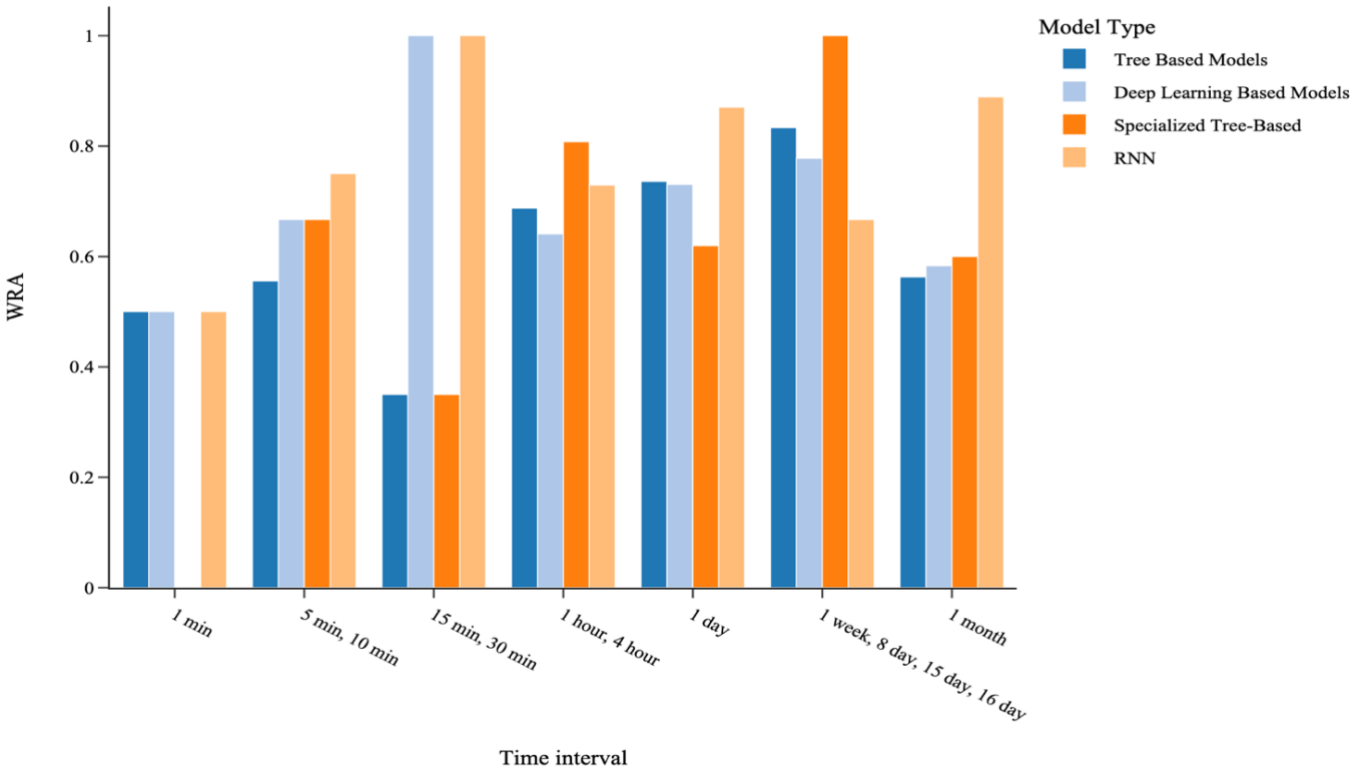

Figure 2.13: Illustrates the WRA score for each model based on time interval of data. Blue bars represent the overall model types, while orange bars indicate the best-performing class of models.

[4] The time intervals of 8 days and 16 days were tested using the same model, see Kang, et al., 2020

FPA advantage of 27.78% and a WRA advantage of 0.2509. On the other hand, SPTB models surpass RNN models in the 1/4-hour and 7/8/15/16-day intervals. The 1/4-hour interval (16 occurrences) is particularly noteworthy, with SPTB models outperforming RNN by 18.59% in FPA and 0.0785 in WRA.

2.5.3.5 Impact of Research Focus on Observed Model Performance

The papers surveyed in this study exhibit diverse objectives and emphases, ranging from developing hybrid deep neural networks to benchmarking specific advanced TBML models against DL models, or conducting balanced evaluations across a variety of ML and DL approaches. This subsection examines whether the primary focus of the research—categorized as TBML models, DL models, or a balanced approach—affects the observed performance outcomes of the evaluated models. Figures 2.14 and 2.15 present the FPA and WRA scores, respectively, for TBML and DL models under each research focus category. The results reveal a noticeable bias in performance outcomes depending on the primary focus of the papers:

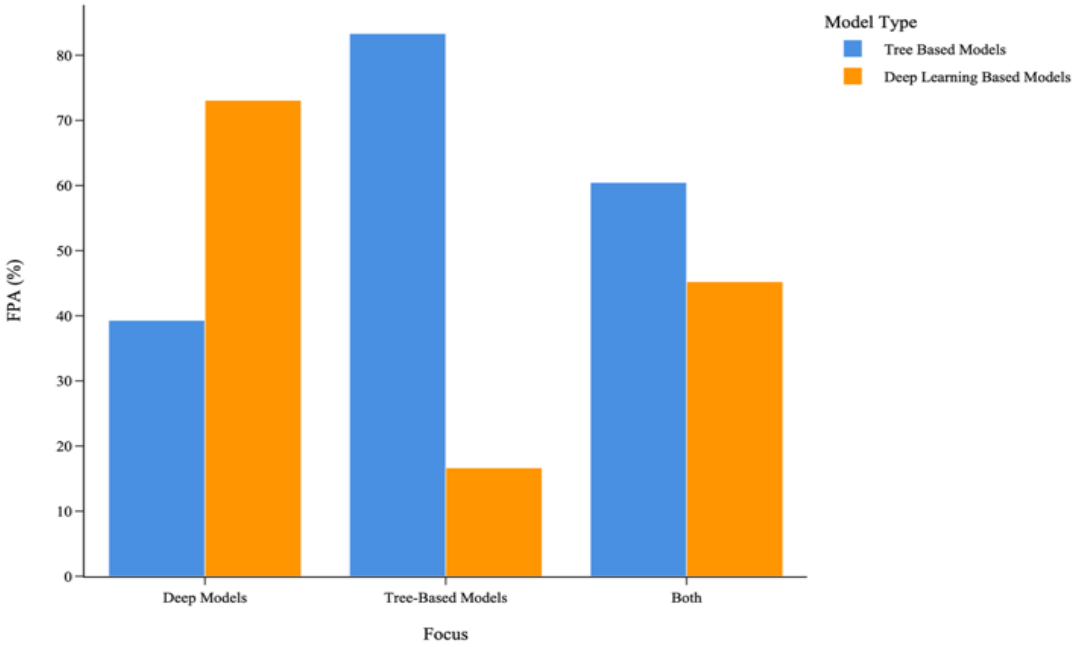

Figure 2.14: Illustrates the FPA score for each model class based on the focus of the paper.

1. **Deep Learning-Focused Papers:**

   When the primary focus of the paper is on deep learning models, DL models outperform TBML models significantly. The FPA score for DL models is 33.79% higher, and the WRA score is 0.2891 points higher than TBML models. This finding suggests that papers with a DL emphasis may introduce methodological, architectural, or experimental advantages tailored to highlight DL performance.

2. **Tree-Based Model-Focused Papers:**

   Conversely, when papers focus on TBML models, the observed performance skews in favor of TBML models. In this category, TBML models achieve a 66.67% higher FPA score and a 0.5694 higher WRA score compared to DL models. These results indicate that TBML focused research often optimizes conditions or design choices that favor these methods.

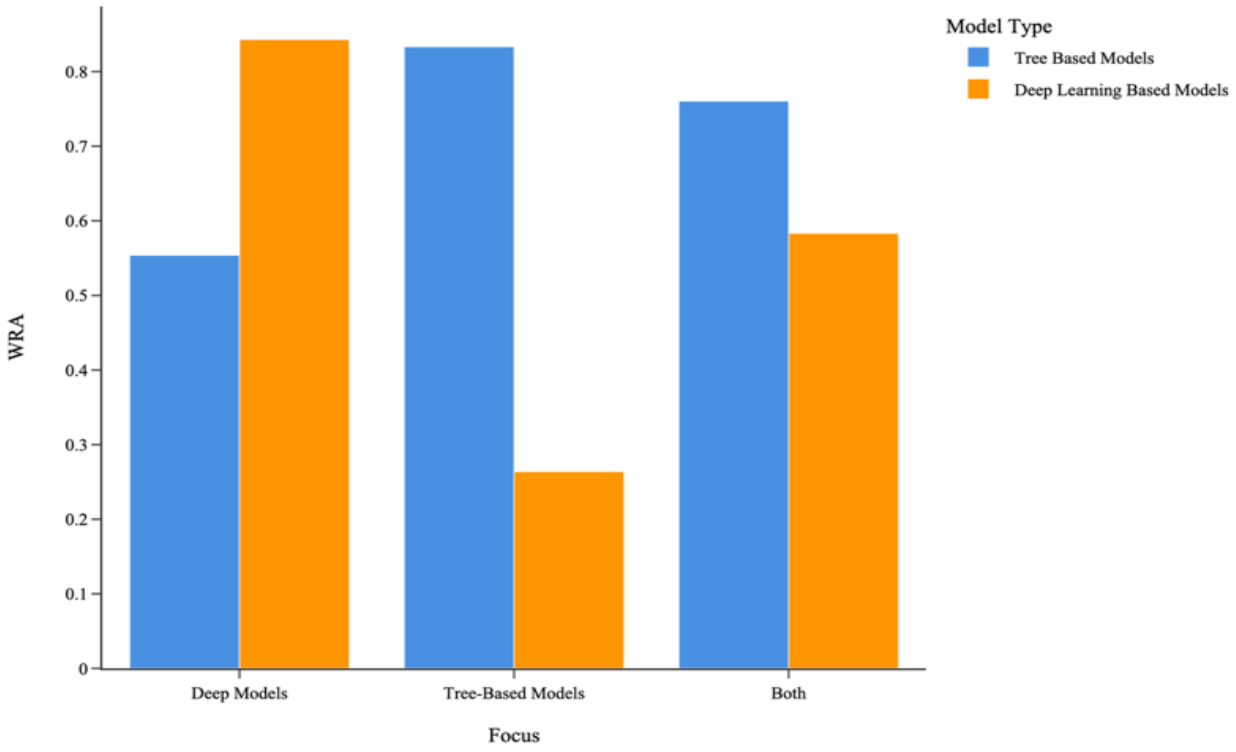

Figure 2.15: Illustrates the WRA score for each model class based on the focus of the paper.

3. **Balanced Focus Papers:**

   In papers with no specific emphasis on either model class, TBML models slightly outperform DL models. The FPA score for TBML models is 15.23% higher, and the WRA score is 0.1771 points higher than DL models. This finding suggests that when research is conducted without bias toward a specific model class, TBML models may have a slight advantage, potentially due to their relative simplicity and robustness in a range of scenarios.

2.5.3.6 Model Training Time Analysis

An essential factor to consider when evaluating the performance of ML models is their training time, particularly given the significant computational costs associated with some models. The training time directly impacts the feasibility of deploying these models in real-world applications where computational efficiency is often critical. Of the surveyed papers, ten studies (Comert et al., 2021; Ge et al., 2019; Hewamalage, Bergmeir, & Bandara, 2022; J. Ke et al., 2017; S. Liu et al., 2021; Ngarambe et al., 2020; Pan et al., 2023; Shi, He, & Liu, 2018; Wei et al., 2021; Zrira et al., 2024) provide training time comparisons for the models evaluated. In these papers, the training time of the best performing TBML model and DL model was analyzed.

The results demonstrate that TBML models significantly outperform DL models in terms of training efficiency. Notably, there is only one instance (Ge et al., 2019) where a DL model, an MLP, trained faster than its TBML counterpart (XGBoost). In this specific case, the MLP achieved a 22.55% reduction in training time compared to XGBoost. However, across all ten studies, TBML models demonstrated a marked advantage. On average, TBML models were 126,385.01% faster than DL models, with the median training time advantage being 5,603.43% faster. This stark difference underscores the efficiency of TBML models in scenarios where computational resources and time constraints are limiting factors.

2.5.3.7 Analysis of Error Metrics in Model Evaluation

Evaluating model performance requires the use of appropriate error metrics, which vary depending on whether the task involves classification or regression. This subsection provides a comprehensive analysis of the error metrics employed across the surveyed papers.

2.5.3.7.1 Error Metrics for Classification Models

The classification models reviewed in this study were evaluated using the following set of metrics:

- False Positive Rate (FPR)
- Kappa Coefficient (KC)
- Positive Predictive Value (PPV)
- Negative Predictive Value (NPV)
- Receiver-Operating Characteristic (ROC) Curve
- Matthews Correlation Coefficient (MCC)
- Area Under the ROC Curve (AUC)
- Sensitivity
- Specificity
- Recall
- Precision
- F1 Score
- Accuracy

The frequency of these metrics' usage is illustrated in Figure 2.16. The metric names are reported exactly as listed in the surveyed papers, even if some represent equivalent measures (e.g., recall and sensitivity). Among the most employed metrics, Recall was used in 35.29% of the studies, Precision in 41.18%, F1 Score in 47.06%, and Accuracy in 76.47% of the papers.

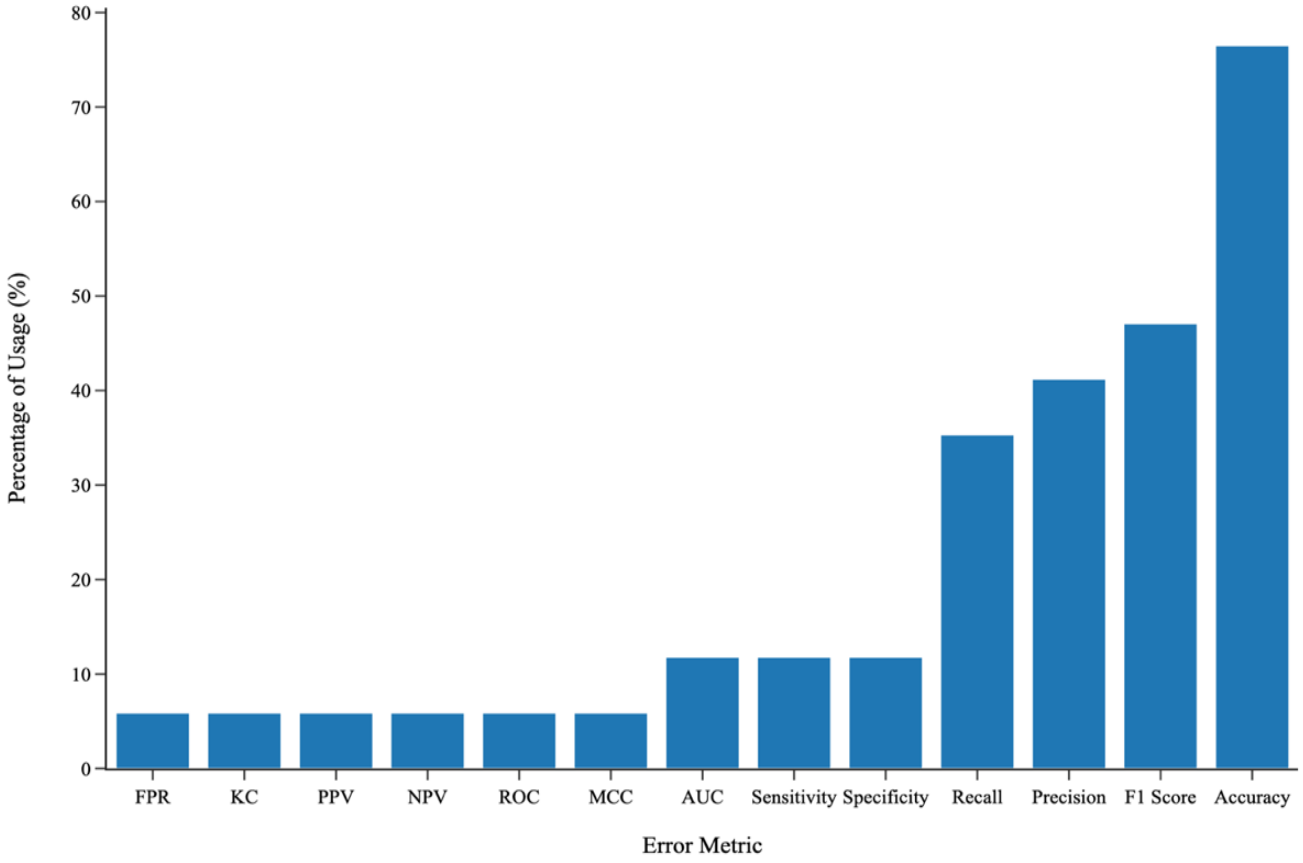

Figure 2.16: Illustrates the percent of papers that use each error metric for classification tasks.

2.5.3.7.2 Error Metrics for Regression Models

The regression models reviewed in this study were evaluated using the following set of metrics:

- Index of Agreement (IA)
- Normalized Mean Absolute Percentage Error (NMAPE)
- Prediction of Change in Direction (POCID)
- Mean Normalized Bias (MNB)
- Normalized Mean Bias Error (NMBE)
- Root Mean Squared Percentage Error (RMSPE)
- Root Squared Logarithmic Error (RMSLE)

- Mean
- Percent Bias (PBIAS)
- R
- Mean Absolute Scaled Error (MASE)
- Symmetric Mean Absolute Error (SMAPE)
- Coefficient of Variation of the Root Mean Square Error (CVRMSE)
- Nash-Sutcliffe Efficiency (NSE)
- Domain-Specific Error Metrics
- Mean Squared Error (MSE)
- Mean Absolute Percentage Error (MAPE)
- R^2

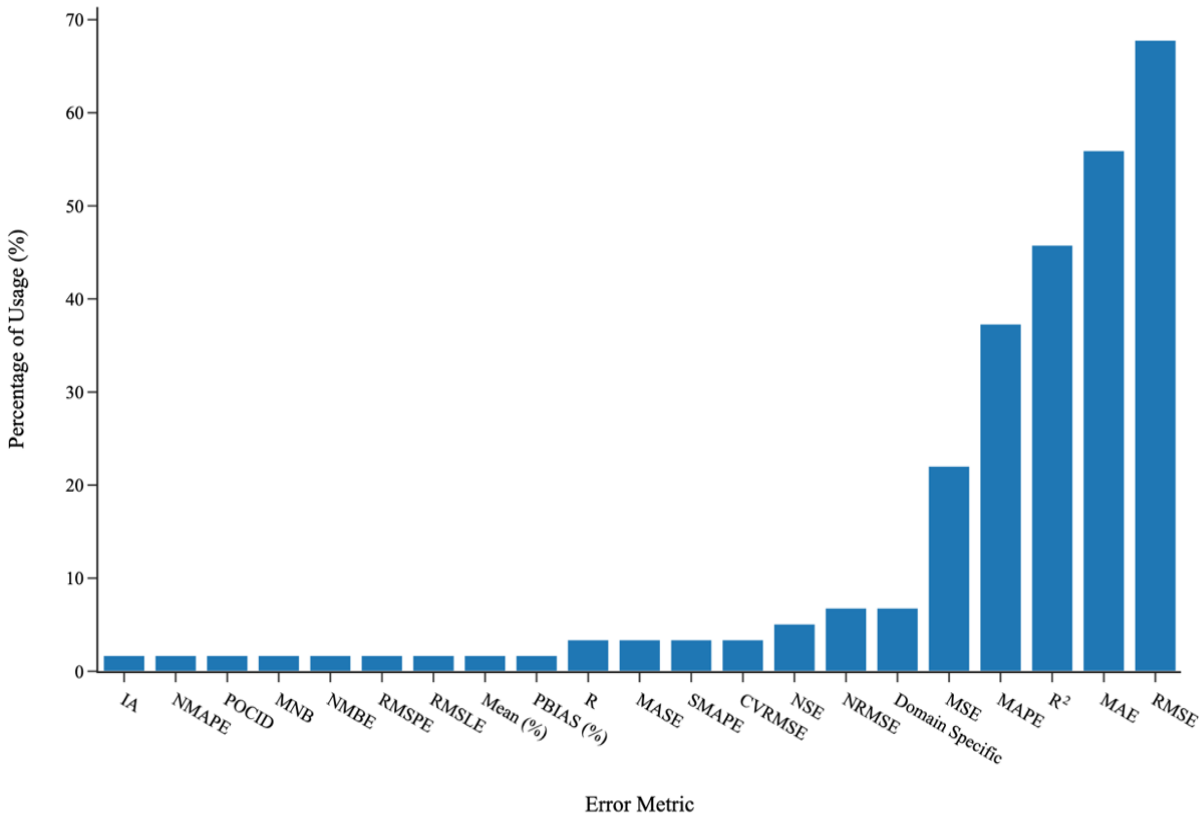

Figure 2.17: Illustrates the percent of papers that use each error metric for regression tasks.

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)

The prevalence of these metrics is depicted in Figure 2.17. Among the traditional metrics, MSE was used in 22.03% of the papers, MAPE in 37.29%, R^2 in 45.76%, MAE in 55.93%, and RMSE in 67.80%.

2.5.3.8 Hyperparameter Optimization Techniques

Hyperparameter tuning is a critical aspect of ML model development, as the choice of hyperparameters can significantly influence performance. Several of the surveyed articles specified the hyperparameter optimization techniques employed in their studies. Figure 2.18 illustrates the relative prevalence of these techniques across the surveyed papers. The most used hyperparameter optimization technique was Grid Search, followed by Bayesian Optimization (BO), Random Search, Manual Optimization, and OPTUNA

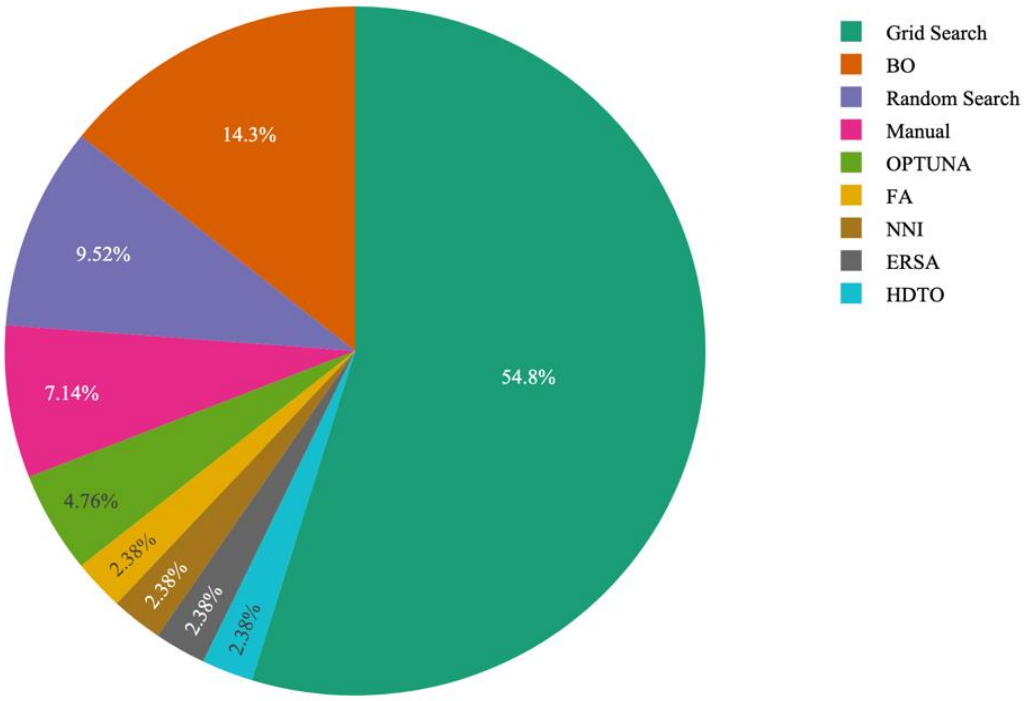

Figure 2.18: Illustrates the usage for each hyperparameter optimization technique.

Automatic Hyperparameter Tuning Software. Less frequently used methods included algorithms such as the Firefly Algorithm (FA) (Zahoor Ali et al., 2020), Neural Network Intelligence (NNI) (López Santos et al., 2022), Enhanced Reptile Search Algorithm (ERSA) (Pavlov-Kagadejev et al., 2024), and Hidden Dipper Throated Optimization (HDTO) (Singh et al., 2024), each of which was mentioned in only one paper. While many studies focused on a single optimization approach, one paper (Pan et al., 2023) provided a comparative evaluation of multiple hyperparameter optimization methods. This study found that Bayesian Optimization outperformed techniques such as Grid Search and Particle Swarm Optimization (PSO) in terms of search effectiveness. By leveraging probabilistic models to guide the search process, Bayesian Optimization was able to reduce computational costs while identifying optimal hyperparameter configurations more efficiently.

2.5.3.9 Comparative Analysis of Hybrid Models

Hybrid models, which integrate multiple ML and DL architectures, are a common approach to improving predictive performance in time series tasks. Across the surveyed literature, hybrid models often outperform individual stand-alone models, although there are notable exceptions. This section examines these exceptions as well as provides insights into the relative performance of different hybrid configurations.

2.5.3.9.1 Performance of Hybrid Models vs. Individual Models

While hybrid models generally exhibit superior performance, a few studies highlight cases where individual models outperform hybrids:

1. **Seydi, Amani and Ghorbanian (2022):** A 2D CNN, 3D CNN, and XGBoost model each outperformed a hybrid RNN-CNN model.
2. **Oyucu et al. (2024):** RF and XGBoost models surpassed multiple hybrid models, including CNN-LSTM, CNN-GRU, RNN-GRU, and RNN-LSTM configurations.

3. **Saravanan and Bhagavathiappan (2024):** A CatBoost model outperformed a spatio-temporal attention-based CNN and Bi-LSTM hybrid model.

2.5.3.9.2 Hybrid Models Compared to Other Hybrids

Several papers explored the relative performance of different hybrid configurations, providing valuable insights into design considerations:

1. **Pan et al. (2023):** a hybrid CNN-LSTM-Attention model outperformed a CNN-LSTM model, which in turn outperformed an LSTM-Attention model.
2. **Zhu et al. (2023):** CEEMDAN decomposition was applied to both an XGBoost and DL model. The hybrid XGBoost-CEEMDAN model performed better than its DL-based counterpart.
3. **Li et al. (2024):** A Bi-LSTM-LightGBM hybrid outperformed a Bi-LSTM-FFNN hybrid.
4. **Pavlov-Kagadejev et al. (2024):** LSTM models with decomposition techniques, Variational Mode Decomposition (VMD) and Empirical Mode Decomposition (EMD), were compared. The LSTM-VMD hybrid outperformed the LSTM-EMD hybrid.
5. **Zrira et al. (2024):** an Attention-based Bi-LSTM hybrid model performed better than an Attention-based Bi-GRU hybrid model.
6. **Oyucu et al. (2024):** Among four hybrid DL models, CNN-LSTM demonstrated the best performance, followed by CNN-GRU, RNN-GRU, and RNN-LSTM.
7. **Nan et al. (2022):** Four hybrid models were compared, with relative performance as follows: LSTM-XGBoost > FFNN-XGBoost > LSTM-MLR > FFNN-MLR.

2.5.4 Analysis

This section synthesizes the biggest takeaways from the experimental results, highlighting key insights, overarching trends, and limitations observed across the surveyed studies.

Comparative Performance of Tree-Based and Deep Learning Models: The experimental results indicate that TBML models and DL models exhibit comparable performance overall. Among their respective subcategories, SPTB models and RNNs emerge as the best performers, further underscoring why these models dominate much of the research focus in this paper. However, TBML models demonstrate a slight overall edge due to the stronger performance of their weakest algorithms. Notably, RF and GBDT outperform the weakest DL models, FFNN and its variants.

At the individual algorithm level, CatBoost, LSTM, and LightGBM stand out as the best-performing models. While CatBoost had limited representation in the surveyed papers, it is highly similar to LightGBM and XGBoost—both of which had much greater representation and demonstrated strong performance—further supporting the conclusion that CatBoost is a top-performing model. On the other hand, attention-based architectures (e.g., Transformers) also show strong potential, but unlike CatBoost, there are no directly comparable algorithms in this category with more extensive research in the surveyed papers. This lack of broader representation underscores the need for further validation of Transformers through more illustrative studies.

Domain-Specific Performance: Performance analysis across application domains reveals distinct advantages for specific model groups:

- **TBML Models**: Outperform in tasks related to energy and utilities, transportation and urban mobility, anomaly detection, and miscellaneous applications.
- **DL Models**: Outperform in tasks related to environmental and meteorological predictions, structural and mechanical health monitoring, and financial/market trend forecasting.
- **SPTB Models**: Outperform in tasks related to transportation and miscellaneous applications, while RNN models dominate in environmental, healthcare, and finance-related tasks.
- **RNN Models:** Outperform in tasks related to Environmental and Meteorological Predictions, Water and Air Quality, Structural and Mechanical Health Monitoring, Stock Market/Finance/Market Trends, Healthcare and Biomedical Predictions.

It is important to note that the dataset includes relatively small representations for some categories (ranging from 3.85% to 21.8% of the total papers). As a result, the findings presented here do not present definitive evidence that any one type of model is dominant within a given domain. Instead, they highlight interesting patterns and trends that merit further investigation and validation with larger datasets.

Dataset Size: Analysis of the influence of dataset size on the relative performance of machine learning models for time series prediction reveals some interesting trends. In ML tasks with the least amount of data, DL models perform comparably to TBML models overall. However, RNN models demonstrate a significant advantage over SPTB models. As dataset size increases, DL models begin to show a slight edge over TBML models in the lower mid-sized range (2173–7800 samples), although the gap narrows in the upper mid-sized range (7800–35712 samples). Across this range, RNN models continue to outperform SPTB models, though the margin diminishes, reflecting SPTB models' growing robustness with increasing data. For larger datasets, TBML models gain a clear advantage over DL models, with their performance gap widening at the largest dataset range (206573–11275200 samples). This trend demonstrates the scalability of TBML models, particularly SPTB models, in handling vast amounts of data.

Temporal Resolution: This study examines the impact of time interval lengths on the performance of ML models for time series prediction. However, no consistent pattern or trend emerges across the time intervals. DL models perform better in some intervals, while TBML performs better in others. As such, there is no common thread that unites these results across all temporal resolutions studied.

Influence of Research Focus: A notable finding is the inherent bias in performance outcomes depending on the research focus of the paper. DL models significantly outperform TBML models in papers focused on DL techniques, and vice versa for TBML-focused research. This suggests that researchers may devote disproportionate attention to optimizing models aligned with their research focus while investing less effort in models they are using for comparative efforts. Readers should exercise caution when interpreting comparative results in such studies. In studies with no specific focus, TBML models slightly edge out DL models providing an interesting case for their superior performance in "bias-free" papers.

Computational Efficiency: One of the most impactful findings is the disparity in training times between TBML and DL models. TBML models were found to be, on average, 2 to 4 orders of magnitude faster than DL models. This makes TBML models particularly appealing for real-world applications where computational cost is a critical consideration.

Error Metrics: Popular error metrics for evaluating classification and regression tasks may provide valuable benchmarks for future research. These include classification metrics such as Recall, Precision, F1 Score, and Accuracy, as well as Regression Metrics such as RMSE, MAE, MAPE, and MSE. Future research may benefit from aligning evaluation methods with these metrics to ensure comparability across studies.

Hyperparameter Optimization: Grid search is the most frequently used hyperparameter optimization method, but it is also the most computationally intensive. Bayesian Optimization offers a promising alternative, providing similar performance with much lower computational demands. Methods such as OPTUNA's automatic hyperparameter tuning software are also emerging as viable options for hyperparameter tuning.

Hybrid Models: This study found strong evidence that combining models generally enhances performance, as evidenced by numerous surveyed papers that implemented hybrid approaches. These hybrid models consistently outperformed individual models, with performance often improving as more models were integrated. There are notable exceptions to this trend, discussed in **Section 2.5.3.9**, where individual models showed superior performance compared to hybrids. These individual model examples exclusively involved SPTB models, further underscoring the robust performance of these algorithms. When comparing hybrid configurations, the results reveal that combinations of SPTB models with RNN models yield particularly strong outcomes. Additionally, hybrid models incorporating either SPTB or RNN architectures with attention-based or CNN models also demonstrated notable performance improvements.

Anecdotal Findings: Beyond the quantitative analysis across all surveyed papers, several anecdotal observations offer additional insights:

1. **Feature Sensitivity**: GBDT models are less affected by redundant or removed features, whereas ANN performance drops significantly when redundant features are added (Bagherzadeh et al., 2021).
2. **Feature Selection**: When all features are provided, XGBoost consistently delivers the best performance. However, when variables are selected using forward selection, other DL models begin to outperform it. Interestingly, the XGBoost model utilizing all features outperforms the XGBoost model that uses only the forward-selected features (Shin et al., 2020).
3. **Domain-Specific Findings**: LightGBM produces more accurate results for top research terms in emerging topics, even though it generally has higher errors than NN (Liang et al., 2021).
4. **Inference Time**: One study reported inference times for their models. They compared an XGBoost model (0.001 sec), with an LSTM model (0.311 sec), and a Bi-LSTM model (1.45 secs), finding XGBoost to be 311 times faster than Bi-LSTM and 1450 times faster than LSTM. This drastically faster inference time emphasizes its practicality in time-sensitive applications (Zrira et al., 2024).
5. **Simulated vs. Real-World Data**: LightGBM matches neural network performance on simulated data but outperforms on real-world datasets (Hewamalage, Bergmeir, & Bandara, 2022).
6. **Time-Series Image Data**: CNN models excel in prediction tasks involving time-series image data (Seydi, Amani, & Ghorbanian, 2022; Su et al., 2021; Zhong, Hu, & Zhou, 2019).

2.6 M5 and M6 Forecasting Competitions

Besides looking at research papers that focus on using ML models for real world applications in time series prediction, competitions that challenge teams to create models for a common dataset provide valuable insights into the comparative performance of machine learning methods for time series prediction. These contests enable direct comparisons under the same controlled conditions. Among such competitions, the M Forecasting Competitions stand out as the most prominent, well-structured, and well-funded. This section focuses on the two most recent iterations: the M5 and M6 Forecasting Competitions.

2.6.1 M5 Forecasting Competition

The M5 Forecasting Competition, held in 2020, focused on predicting retail sales using real-world data comprising 42,840 time series of Walmart unit sales. The competition had two components: the Accuracy Competition and the Uncertainty Competition, each with a prize pool of $50,000. These incentives attracted thousands of participants, creating a large dataset for analysis.

2.6.1.1 M5 Accuracy Competition

The M5 Accuracy Competition (Makridakis, Spiliotis, & Assimakopoulos, 2022) tasked participants with providing the most accurate point forecasts, evaluated using a Weighted Root Mean Squared Scaled Error (WRMSSE) metric. A total of 5,507 teams from 101 countries participated, with LightGBM emerging as the dominant model among the top 50 best performing teams. Brief insights from the top 5 models include:

1. **First Place:** Combined recursive and non-recursive LightGBM models to create 220 models, where the average of 6 models was used to forecast the series, each exploiting a different learning approach and training set.
2. **Second Place**: Created 50 LightGBM models, 5 for each of the 10 stores, utilizing a DL neural network to adjust multipliers based on historical sales data for each store.
3. **Third Place**: Employed 43 recursive neural networks (LSTMs) incorporating over 100 features
4. **Fourth Place:** Created 40 non-recursive LightGBM models
5. **Fifth Place**: Utilized 7 recursive LightGBM models.

Nearly all top 50 submissions applied the last four 28-day windows of data for cross-validation to fine-tune hyperparameters. Many top performing teams, including 1st and 3rd place, used exogenous features like special days and zero-sales periods in their models. The researchers of this challenge concluded that the competition reinforced the value of model combination, cross-learning, and cross-validation. New findings included the superior performance of the LightGBM model compared to all others, as well as the importance of exogenous/explanatory variables used for forecasting.

2.6.1.2 M5 Uncertainty Competition

The Uncertainty competition (Makridakis et al., 2022) tasked participants with forecasting the distribution of realized values, requiring predictions of nine quantiles (0.005, 0.025, 0.165, 0.250, 0.500, 0.750, 0.835, 0.975, and 0.995). The Weighted Scaled Pinball Loss function was used to evaluate performance. Although this competition was less popular, attracting 892 teams, the reliance on LightGBM models remained consistent, with four of the top five submissions incorporating it in their frameworks. Brief insights from the top 5 models include:

1. **First Place**: Utilized 126 LightGBM models, one for each quantile and aggregation level.
2. **Second Place**: Combined recursive LightGBM models, statistical methods, and simple time series forecasting techniques.
3. **Third Place**: Employed a hybrid approach integrating LightGBM and neural networks.
4. **Fourth Place**: Used two LSTM-based neural networks.
5. **Fifth Place**: Implemented 280 LightGBM models in a comprehensive ensemble.

A Monte Carlo simulation, used by the sixth-place team, was the only top-50 method not involving LightGBM, XGBoost, or neural networks. The findings from the Uncertainty competition mirrored those of the Accuracy competition, reaffirming the dominance of LightGBM and the importance of model combination. A notable observation was the stark contrast in participant expertise: while the Accuracy competition was won by an undergraduate student with limited knowledge in the domain of retail sales, the Uncertainty competition was dominated by Kaggle masters and grandmasters with strong statistical backgrounds.

2.6.2 M6 Forecasting Competition

The M6 forecasting competition from 2022 to 2023 (Makridakis et al., 2024) marked a significant evolution in the M series. With a prize pool of $300,000, it attracted 226 teams to participate. The challenge revolved

around creating investment portfolios using real-time data from 50 U.S. stocks and 50 exchange-traded funds (ETFs). Unlike prior iterations, the M6 competition emphasized both forecasting and investment decision-making, awarding prizes for forecasting accuracy, investment performance, and overall performance. Participants were given flexibility in their data sources and methodologies. While organizers provided an optional dataset of adjusted closing values, teams could choose their own data, frequency, and supporting information, such as economic indicators or news. The competition spanned 12 months, with teams submitting monthly forecasts and investment strategies for the subsequent 20 trading days.

The winners in forecasting, investment, and the combined "duathlon" category all used distinct methods. The top forecasting model used a Bayesian dynamic factor model. The best-performing investment model relied on AutoTS, an open-source Python library for probabilistic time series forecasting. The duathlon champion employed a meta-learning model using NNs, which ranked fourth in forecasting. Notably, an XGBoost-based approach secured second place in both forecasting and the duathlon. Interestingly, the team that won the investment challenge placed 92nd in forecasting, and the second-place investment team, which used a type of exponential smoothing model (ATA), ranked 110th in forecasting. In fact, the researchers found zero correlation between forecasting and investment performance emphasizing that a team's ability to accurately predict a stock's future price was not the driving factor in creating the most profitable investment portfolio.

The researchers were not surprised that the best performing methods included both conventional econometric time series methods as well as sophisticated machine learning methods because of the unique challenges present in financial forecasting including external factors, seasonality, stochastic trends, etc. The key takeaway from the M6 competition is that the choice of data and its usage is as critical as the forecasting techniques themselves in achieving superior results.

2.6.3 Takeaways from M5 and M6 Forecasting Competitions

One of the most notable takeaways from the M5 competition was the dominance of LightGBM. Its ease of use allowed even relatively inexperienced participants to excel, as evidenced by the first-place finisher in

the M5 Accuracy Competition, an undergraduate student. Most top-performing teams relied on ensembles of LightGBM models, leveraging its efficiency in handling large datasets. The results of the M5 competition also emphasized the importance of combining multiple models. Most top teams utilized ensembles, often containing hundreds of individual models, to enhance the accuracy of their predictions. Across both the M5 and M6 competitions, cross-validation (CV) and hyperparameter tuning emerged as indispensable components of successful forecasting methodologies. All top-performing teams employed thorough CV to optimize their models.

One critical takeaway from the M6 competition was the importance of data quality, feature engineering, and the inclusion of exogenous variables. Unlike the M5 competition, where all participants worked with a standardized dataset, M6 participants were responsible for sourcing their own data. The complexity of this competition, including the unique challenges in the financial domain, also resulted in the number of participating teams to decrease to less than 4% of the M5 competition's turnout (despite a substantial increase in incentives). The results of the competition showed no clear consensus on the best-performing methodology and suggest that the importance of data quality and strong feature engineering, including the exogenous variables chosen, often outweigh the importance in the choice of the prediction model itself.

2.7 Conclusion

This survey of machine learning methods for time series prediction illuminates several key strategies for approaching time series forecasting tasks. Based on these findings, the following approach is recommended for researchers and practitioners tackling time series prediction problems.

First and foremost, the domain and data play a pivotal role in determining the success of a forecasting model. Careful domain analysis to identify the most impactful data sources, coupled with meticulous feature engineering to extract and construct relevant features, should be the initial focus. The importance of high-quality data and strong exogenous features cannot be overstated, as evidenced by both the findings of this study and the results of recent forecasting competitions. Investing significant effort in this phase is likely

to yield the greatest dividends in predictive performance. Once the data is prepared, starting with a Tree-Based Machine Learning (TBML) model like LightGBM—or CatBoost for datasets with a high proportion of categorical features—is a logical choice. These gradient boosting methods offer several advantages: they are computationally efficient, require less feature selection, and have demonstrated competitive or superior performance compared to deep learning approaches in various settings. Such models serve as a low-cost baseline for experimentation and iterative improvement.

Model evaluation should be guided by task-specific metrics. For regression problems, metrics such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) are appropriate, while classification tasks should use Recall, Precision, F1 Score, and Accuracy. Hyperparameter tuning is another critical component of the modeling process. If prior knowledge exists about potential hyperparameter values, a grid search can be employed for systematic exploration. Otherwise, techniques like Bayesian Optimization or libraries such as OPTUNA are recommended for their ability to efficiently explore large parameter spaces and identify optimal configurations. If the initial model fails to meet desired performance standards or further enhancements are required, Recurrent Neural Networks (RNNs), such as LSTMs or GRUs, are a logical next step. When building a deep learning model, it is crucial to carefully select and preprocess features to optimize the model's performance and ensure it delivers the best results. As with earlier stages, hyperparameter tuning should be employed to maximize performance. Finally, combining models—such as the initial TBML model with a deep learning model—can often yield superior results. Ensemble methods leverage the strengths of multiple methodologies, capturing diverse patterns and mitigating weaknesses inherent to any single approach, as demonstrated in this study.

In conclusion, effective time series prediction is a combination of domain knowledge, data quality, rigorous preprocessing, and the strategic application of the best machine learning methodologies. Starting with scalable and interpretable TBML models, fine-tuning their parameters, and iteratively incorporating advanced techniques like DL provides an efficient framework for tackling time series forecasting challenges. By following this approach, researchers and practitioners can maximize predictive accuracy while optimizing computational costs in a variety of domains.

Appendix A: Surveyed Papers

The following 79 papers were selected based on citation count and adherence to inclusion criteria:

Bagherzadeh et al., 2021; Barrera-Animas et al., 2022; Chakraborty & Elzarka, 2019; Chen, Guan, & Li, 2021; Comert et al., 2021; Cui et al., 2021; Farsi, 2021; Galicia et al., 2019; Ge et al., 2019; Geng et al., 2021; Ghimire et al., 2023; Gong et al., 2020; Hewamalage, Bergmeir, & Bandara, 2022; Hussein et al., 2020; Ibañez et al., 2022; Jing et al., 2021; Joseph et al., 2022; Ju et al., 2019; Kang et al., 2020; J. Ke et al., 2017; Khan et al., 2020; Kumar et al., 2023; Kwon, Kim, & Han, 2019; G. Li et al., 2022; Li et al., 2020; Y. Li et al., 2022; Li, Yang, & Sun, 2022; Liang et al., 2021; S. Liu et al., 2021; López Santos et al., 2022; Luo et al., 2021; Mazzia, Khaliq, & Chiaberge, 2020; Nan et al., 2022; Ngarambe et al., 2020; Pan et al., 2023; Paudel et al., 2023; Pham Hoang et al., 2022; Priyadarshi et al., 2019; Prodhan et al., 2021; Qiu et al., 2020; Rafi et al., 2021; Ravichandran et al., 2021; Ribeiro et al., 2022; Safat, Asghar, & Gillani, 2021; Seydi, Amani, & Ghorbanian, 2022; Shangguan et al., 2022; Shen et al., 2022; Shi, He, & Liu, 2018; Shin et al., 2020; Srivastava & Eachempati, 2021; Su et al., 2021; Sundararajan et al., 2021; Teinemaa et al., 2018; Ting et al., 2020; Torres, Martínez-Álvarez, & Troncoso, 2022; Wang et al., 2021; Wei et al., 2021; Wu et al., 2021; Yang et al., 2020; Yu et al., 2020; Zahoor Ali et al., 2020; Zhang et al., 2022; Zheng et al., 2020; Zhong, Hu, & Zhou, 2019; Zhu et al., 2023, Abdikan et al., 2023; Guan et al., 2024; Phan et al., 2023; Wang et al., 2023, Inbar & Avisar, 2024; Li et al., 2024; Mangukiya & Sharma, 2024; Oyucu et al., 2024; Pavlov-Kagadejev et al., 2024; Saravanan & Bhagavathiappan, 2024; Singh et al., 2024; Zhang et al., 2024; Zhao et al., 2024; Zrira et al., 2024

References

Abdikan, S., Sekertekin, A., Narin, O. G., Delen, A., & Balik Sanli, F. (2023). A comparative analysis of SLR, MLR, ANN, XGBoost and CNN for crop height estimation of sunflower using Sentinel-1 and Sentinel-2. *Advances in Space Research*, *71*(7), 3045-3059.

Bagherzadeh, F., Mehrani, M.-J., Basirifard, M., & Roostaei, J. (2021). Comparative study on total nitrogen prediction in wastewater treatment plant and effect of various feature selection methods on machine learning algorithms performance. *Journal of Water Process Engineering*, *41*, 102033.

Barrera-Animas, A. Y., Oyedele, L. O., Bilal, M., Akinosho, T. D., Delgado, J. M. D., & Akanbi, L. A. (2022). Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, *7*, 100204.

Chakraborty, D., & Elzarka, H. (2019). Advanced machine learning techniques for building performance simulation: a comparative analysis. *Journal of Building Performance Simulation*, *12*(2), 193-207.

Chen, H., Guan, M., & Li, H. (2021). Air Quality Prediction Based on Integrated Dual LSTM Model. *IEEE Access*, *9*, 93285-93297.

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA.

Comert, G., Khan, Z., Rahman, M., & Chowdhury, M. (2021). Grey models for short-term queue length predictions for adaptive traffic signal control. *Expert Systems with Applications*, *185*, 115618.

Cui, Z., Qing, X., Chai, H., Yang, S., Zhu, Y., & Wang, F. (2021). Real-time rainfall-runoff prediction using light gradient boosting machine coupled with singular spectrum analysis. *Journal of Hydrology*, *603*, 127124.

Farsi, M. (2021). Application of ensemble RNN deep neural network to the fall detection through IoT environment. *Alexandria Engineering Journal*, *60*(1), 199-211.

Galicia, A., Talavera-Llames, R., Troncoso, A., Koprinska, I., & Martínez-Álvarez, F. (2019). Multi-step forecasting for big data time series based on ensemble learning. *Knowledge-Based Systems*, *163*, 830-841.

Ge, Y., Wang, Q., Wang, L., Wu, H., Peng, C., Wang, J., Xu, Y., Xiong, G., Zhang, Y., & Yi, Y. (2019). Predicting post-stroke pneumonia using deep neural network approaches. *International Journal of Medical Informatics*, *132*, 103986.

Geng, L., Che, T., Ma, M., Tan, J., & Wang, H. (2021). Corn Biomass Estimation by Integrating Remote Sensing and Long-Term Observation Data Based on Machine Learning Techniques. *Remote Sensing*, *13*(12).

Ghimire, S., Nguyen-Huy, T., Al-Musaylh, M. S., Deo, R. C., Casillas-Pérez, D., & Salcedo-Sanz, S. (2023). A novel approach based on integration of convolutional neural networks and echo state network for daily electricity demand prediction. *Energy*, *275*, 127430.

Gong, M., Bai, Y., Qin, J., Wang, J., Yang, P., & Wang, S. (2020). Gradient boosting machine for predicting return temperature of district heating system: A case study for residential buildings in Tianjin. *Journal of Building Engineering*, *27*, 100950.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354-377.

Guan, S., Wang, Y., Liu, L., Gao, J., Xu, Z., & Kan, S. (2024). Ultra-short-term wind power prediction method based on FTI-VACA-XGB model. *Expert Systems with Applications*, *235*, 121185.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2022). Global models for time series forecasting: A Simulation study. *Pattern Recognition*, *124*, 108441.

Hussein, E. A., Thron, C., Ghaziasgar, M., Bagula, A., & Vaccari, M. (2020). Groundwater Prediction Using Machine-Learning Tools. *Algorithms*, *13*(11).

Ibañez, S. C., Dajac, C. V. G., Liponhay, M. P., Legara, E. F. T., Esteban, J. M. H., & Monterola, C. P. (2022). Forecasting Reservoir Water Levels Using Deep Neural Networks: A Case Study of Angat Dam in the Philippines. *Water*, *14*(1).

Inbar, O., & Avisar, D. (2024). Enhancing wastewater treatment through artificial intelligence: A comprehensive study on nutrient removal and effluent quality prediction. *Journal of Water Process Engineering*, *61*, 105212.

Jing, Y., Hu, H., Guo, S., Wang, X., & Chen, F. (2021). Short-Term Prediction of Urban Rail Transit Passenger Flow in External Passenger Transport Hub Based on LSTM-LGB-DRS. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7), 4611-4621.

Joseph, R. V., Mohanty, A., Tyagi, S., Mishra, S., Satapathy, S. K., & Mohanty, S. N. (2022). A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting. *Computers and Electrical Engineering*, *103*, 108358.

Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., & Rehman, M. U. (2019). A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting. *IEEE Access*, *7*, 28309-28318.

Kang, Y., Ozdogan, M., Zhu, X., Ye, Z., Hain, C., & Anderson, M. (2020). Comparative assessment of environmental variables and machine learning algorithms for maize yield prediction in the US Midwest. *Environmental Research Letters*, *15*(6), 064005.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). *LightGBM: a highly efficient gradient boosting decision tree* Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA.

Ke, J., Zheng, H., Yang, H., & Chen, X. (2017). Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, *85*, 591-608.

Khan, P. W., Byun, Y. C., Park, N., Waqas Khan, P., Byun, Y.-C., Lee, S.-J., & Park, N. (2020). Machine Learning Based Hybrid System for Imputation and Efficient Energy Demand Forecasting. *Energies.*, *13*(11), 2681.

Kumar, V., Kedam, N., Sharma, K. V., Mehta, D. J., & Caloiero, T. (2023). Advanced Machine Learning Techniques to Improve Hydrological Prediction: A Comparative Analysis of Streamflow Prediction Models. *Water*, *15*(14).

Kwon, D. H., Kim, J. B., & Han, Y. H. (2019). Time Series Classification of Cryptocurrency Price Trend Based on a Recurrent LSTM Neural Network. *JOURNAL OF INFORMATION PROCESSING SYSTEMS*, *15*(3), 694-706.

Li, G., Zhang, A., Zhang, Q., Wu, D., & Zhan, C. (2022). Pearson Correlation Coefficient-Based Performance Enhancement of Broad Learning System for Stock Price Prediction. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *69*(5), 2413-2417.

Li, L., Dai, S., Cao, Z., Hong, J., Jiang, S., & Yang, K. (2020). Using improved gradient-boosted decision tree algorithm based on Kalman filter (GBDT-KF) in time series prediction. *The Journal of Supercomputing*, *76*(9), 6887-6900.

Li, Y., Bao, T., Gao, Z., Shu, X., Zhang, K., Xie, L., & Zhang, Z. (2022). A new dam structural response estimation paradigm powered by deep learning and transfer learning techniques. *Structural Health Monitoring*, *21*(3), 770-787.

Li, Y., Yang, C., & Sun, Y. (2022). Dynamic Time Features Expanding and Extracting Method for Prediction Model of Sintering Process Quality Index. *IEEE Transactions on Industrial Informatics*, *18*(3), 1737-1745.

Li, Z., Ma, E., Lai, J., & Su, X. (2024). Tunnel deformation prediction during construction: An explainable hybrid model considering temporal and static factors. *Computers & Structures*, *294*, 107276.

Liang, Z., Mao, J., Lu, K., Ba, Z., & Li, G. (2021). Combining deep neural network and bibliometric indicator for emerging research topic prediction. *Information Processing & Management, 58*(5), 102611.

Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379*(2194), 20200209.

Liu, S., Zeng, A., Lau, K., Ren, C., Chan, P.-w., & Ng, E. (2021). Predicting long-term monthly electricity demand under future climatic and socioeconomic changes using data-driven methods: A case study of Hong Kong. *Sustainable Cities and Society, 70*, 102936.

Liu, Z., Zhu, Z., Gao, J., & Xu, C. (2021). Forecast Methods for Time Series Data: A Survey. *IEEE Access, 9*, 91896-91912.

López Santos, M., García-Santiago, X., Echevarría Camarero, F., Blázquez Gil, G., & Carrasco Ortega, P. (2022). Application of Temporal Fusion Transformer for Day-Ahead PV Power Forecasting. *Energies, 15*(14).

Luo, J., Zhang, Z., Fu, Y., & Rao, F. (2021). Time series prediction of COVID-19 transmission in America using LSTM and XGBoost algorithms. *Results in Physics, 27*, 104462.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting, 38*(4), 1346-1364.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., Chen, Z., Gaba, A., Tsetlin, I., & Winkler, R. L. (2022). The M5 uncertainty competition: Results, findings and conclusions. *International Journal of Forecasting, 38*(4), 1365-1385.

Makridakis, S., Spiliotis, E., Hollyman, R., Petropoulos, F., Swanson, N., & Gaba, A. (2024). The M6 forecasting competition: Bridging the gap between forecasting and investment decisions. *International Journal of Forecasting*.

Mangukiya, N. K., & Sharma, A. (2024). Alternate pathway for regional flood frequency analysis in data-sparse region. *Journal of Hydrology, 629*, 130635.

Mazzia, V., Khaliq, A., & Chiaberge, M. (2020). Improvement in Land Cover and Crop Classification based on Temporal Features Learning from Sentinel-2 Data Using Recurrent-Convolutional Neural Network (R-CNN). *Applied Sciences*, *10*(1).

Nan, S., Tu, R., Li, T., Sun, J., & Chen, H. (2022). From driving behavior to energy consumption: A novel method to predict the energy consumption of electric bus. *Energy*, *261*, 125188.

Ngarambe, J., Irakoze, A., Yun, G. Y., & Kim, G. (2020). Comparative Performance of Machine Learning Algorithms in the Prediction of Indoor Daylight Illuminances. *Sustainability*, *12*(11).

Oyucu, S., Dümen, S., Duru, İ., Aksöz, A., & Biçer, E. (2024). Discharge Capacity Estimation for Li-Ion Batteries: A Comparative Study. *Symmetry*, *16*(4).

Pan, S., Yang, B., Wang, S., Guo, Z., Wang, L., Liu, J., & Wu, S. (2023). Oil well production prediction based on CNN-LSTM model with self-attention mechanism. *Energy*, *284*, 128701.

Paudel, D., de Wit, A., Boogaard, H., Marcos, D., Osinga, S., & Athanasiadis, I. N. (2023). Interpretability of deep learning models for crop yield forecasting. *Computers and Electronics in Agriculture*, *206*, 107663.

Pavlov-Kagadejev, M., Jovanovic, L., Bacanin, N., Deveci, M., Zivkovic, M., Tuba, M., Strumberger, I., & Pedrycz, W. (2024). Optimizing long-short-term memory models via metaheuristics for decomposition aided wind energy generation forecasting. *Artificial Intelligence Review*, *57*(3), 45.

Pham Hoang, V., Trinh Tan, D., Tieu Khoi, M., Hoang Vuong, P., Tan Dat, T., Khoi Mai, T., Hoang Uyen, P., & The Bao, P. (2022). Stock-Price Forecasting Based on XGBoost and LSTM. *Computer systems science and engineering.*, *40*(1), 237-246.

Phan, Q. T., Wu, Y. K., Phan, Q. D., & Lo, H. Y. (2023). A Novel Forecasting Model for Solar Power Generation by a Deep Learning Framework With Data Preprocessing and Postprocessing. *IEEE Transactions on Industry Applications*, *59*(1), 220-231.

Priyadarshi, R., Panigrahi, A., Routroy, S., & Garg, G. K. (2019). Demand forecasting at retail stage for selected vegetables: a performance analysis. *Journal of Modelling in Management*, *14*(4), 1042-1063.

Prodhan, F. A., Zhang, J., Yao, F., Shi, L., Pangali Sharma, T. P., Zhang, D., Cao, D., Zheng, M., Ahmed, N., & Mohana, H. P. (2021). Deep Learning for Monitoring Agricultural Drought in South Asia Using Remote Sensing Data. *Remote Sensing*, *13*(9).

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). *CatBoost: unbiased boosting with categorical features* Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, Canada.

Qiu, H., Luo, L., Su, Z., Zhou, L., Wang, L., & Chen, Y. (2020). Machine learning approaches to predict peak demand days of cardiovascular admissions considering environmental exposure. *BMC Medical Informatics and Decision Making*, *20*(1), 83.

Rafi, S. H., Nahid Al, M., Deeba, S. R., & Hossain, E. (2021). A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network. *IEEE Access*, *9*, 32436-32448.

Ravichandran, T., Gavahi, K., Ponnambalam, K., Burtea, V., & Mousavi, S. J. (2021). Ensemble-based machine learning approach for improved leak detection in water mains. *Journal of Hydroinformatics*, *23*(2), 307-323.

Ribeiro, A. M. N. C., do Carmo, P. R. X., Endo, P. T., Rosati, P., & Lynn, T. (2022). Short- and Very Short-Term Firm-Level Load Forecasting for Warehouses: A Comparison of Machine Learning and Deep Learning Models. *Energies*, *15*(3).

Safat, W., Asghar, S., & Gillani, S. A. (2021). Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques. *IEEE Access*, *9*, 70080-70094.

Saravanan, K. S., & Bhagavathiappan, V. (2024). Prediction of crop yield in India using machine learning and hybrid deep learning models. *Acta Geophysica*, *72*(6), 4613-4632.

Seydi, S. T., Amani, M., & Ghorbanian, A. (2022). A Dual Attention Convolutional Neural Network for Crop Classification Using Time-Series Sentinel-2 Imagery. *Remote Sensing*, *14*(3).

Shangguan, Q., Fu, T., Wang, J., Fang, S. e., & Fu, L. (2022). A proactive lane-changing risk prediction framework considering driving intention recognition and different lane-changing patterns. *Accident Analysis & Prevention*, *164*, 106500.

Shen, M., Luo, J., Cao, Z., Xue, K., Qi, T., Ma, J., Liu, D., Song, K., Feng, L., & Duan, H. (2022). Random forest: An optimal chlorophyll-a algorithm for optically complex inland water suffering atmospheric correction uncertainties. *Journal of Hydrology*, *615*, 128685.

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

Shi, J., He, G., & Liu, X. (2018, 22-24 Aug. 2018). Anomaly Detection for Key Performance Indicators Through Machine Learning. 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC),

Shin, Y., Kim, T., Hong, S., Lee, S., Lee, E., Hong, S., Lee, C., Kim, T., Park, M. S., Park, J., & Heo, T.-Y. (2020). Prediction of Chlorophyll-a Concentrations in the Nakdong River Using Machine Learning Methods. *Water*, *12*(6).

Singh, R. B., Patra, K. C., Samantra, A., Singh, R. B., Patra, K. C., Pradhan, B., & Samantra, A. (2024). HDTO-DeepAR: A novel hybrid approach to forecast surface water quality indicators. *Journal of environmental management.*, *352*, 120091.

Srivastava, P. R., & Eachempati, P. (2021, 2021/09//). Deep Neural Network and Time Series Approach for Finance Systems: Predicting the Movement of the Indian Stock Market. *Journal of Organizational and End User Computing*, *33*(5), NA.

Su, H., Wang, A., Zhang, T., Qin, T., Du, X., & Yan, X.-H. (2021). Super-resolution of subsurface temperature field from remote sensing observations based on machine learning. *International Journal of Applied Earth Observation and Geoinformation*, *102*, 102440.

Sundararajan, K., Garg, L., Srinivasan, K., Ali Kashif, B., Jayakumar, K., Ganapathy, G. P., Senthil Kumaran, S., & Meena, T. (2021). A Contemporary Review on Drought Modeling Using Machine Learning Approaches. *Computer Modeling in Engineering & Sciences*, *128*(2), 447-487.

Svozil, D., Kvasnicka, V., & Pospichal, J. í. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, *39*(1), 43-62.

Teinemaa, I., Dumas, M., Leontjeva, A., & Maggi, F. M. (2018). Temporal stability in predictive process monitoring. *Data Mining and Knowledge Discovery*, *32*(5), 1306-1338.

Ting, P. Y., Wada, T., Chiu, Y. L., Sun, M. T., Sakai, K., Ku, W. S., Jeng, A. A. K., & Hwu, J. S. (2020). Freeway Travel Time Prediction Using Deep Hybrid Model – Taking Sun Yat-Sen Freeway as an Example. *IEEE Transactions on Vehicular Technology*, *69*(8), 8257-8266.

Torres, J. F., Martínez-Álvarez, F., & Troncoso, A. (2022). A deep LSTM network for the Spanish electricity consumption forecasting. *Neural Computing and Applications*, *34*(13), 10533-10545.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need* Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA.

Wang, X., Li, Y., Qiao, Q., Tavares, A., & Liang, Y. (2023). Water Quality Prediction Based on Machine Learning and Comprehensive Weighting Methods. *Entropy*, *25*(8).

Wang, Z., Hong, T., Li, H., & Ann Piette, M. (2021). Predicting city-scale daily electricity consumption using data-driven models. *Advances in Applied Energy*, *2*, 100025.

Wei, Z., Zhang, T., Yue, B., Ding, Y., Xiao, R., Wang, R., & Zhai, X. (2021). Prediction of residential district heating load based on machine learning: A case study. *Energy*, *231*, 120950.

Wu, W., Chen, J., Yang, Z., & Tindall, M. L. (2021). A Cross-Sectional Machine Learning Approach for Hedge Fund Return Prediction and Selection. *Management Science*, *67*(7), 4577-4601.

Yang, Y., Heppenstall, A., Turner, A., & Comber, A. (2020). Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems. *Computers, Environment and Urban Systems*, *83*, 101521.

Yu, G., Zhang, S., Hu, M., & Wang, Y. K. (2020). Prediction of Highway Tunnel Pavement Performance Based on Digital Twin and Multiple Time Series Stacking. *Advances in Civil Engineering*, *2020*(1), 8824135.

Zahoor Ali, K., Muhammad, A., Javaid, N., Malik, N. S., Shafiq, M., & Choi, J.-G. (2020). Electricity Theft Detection Using Supervised Learning Techniques on Smart Meter Data. *Sustainability*, *12*(19), 8023.

Zhang, L., Wang, C., Hu, W., Wang, X., Wang, H., Sun, X., Ren, W., & Feng, Y. (2024). Dynamic real-time forecasting technique for reclaimed water volumes in urban river environmental management. *Environmental Research*, *248*, 118267.

Zhang, Y., Li, C., Jiang, Y., Sun, L., Zhao, R., Yan, K., & Wang, W. (2022). Accurate prediction of water quality in urban drainage network with integrated EMD-LSTM model. *Journal of Cleaner Production*, *354*, 131724.

Zhao, Z.-h., Wang, Q., Shao, C.-s., Chen, N., Liu, X.-y., & Wang, G.-b. (2024). A state detection method of offshore wind turbines' gearbox bearing based on the transformer and GRU. *Measurement Science and Technology*, *35*(2), 025903.

Zheng, J., Zhang, H., Dai, Y., Wang, B., Zheng, T., Liao, Q., Liang, Y., Zhang, F., & Song, X. (2020). Time series prediction for output of multi-region solar power plants. *Applied Energy*, *257*, 114001.

Zhong, L., Hu, L., & Zhou, H. (2019). Deep learning based multi-temporal crop classification. *Remote Sensing of Environment*, *221*, 430-443.

Zhu, X., Guo, H., Huang, J. J., Tian, S., & Zhang, Z. (2023). A hybrid decomposition and Machine learning model for forecasting Chlorophyll-a and total nitrogen concentration in coastal waters. *Journal of Hydrology*, *619*, 129207.

Zrira, N., Kamal-Idrissi, A., Farssi, R., & Khan, H. A. (2024). Time series prediction of sea surface temperature based on BiLSTM model with attention mechanism. *Journal of Sea Research*, *198*, 102472.

CHAPTER 3

Can AI Beat Human Traders? Exploring Machine Learning in Day Trading[5]

[5] Timothy Hall. To be submitted to a peer-reviewed journal

Abstract

Day trading is a high-risk trading strategy where individuals attempt to capitalize on short-term price movements in financial markets. This paper investigates whether modern machine learning (ML) techniques can outperform human day traders by mimicking technical analysis tools used by humans. This study uses a dataset of all U.S. equities from over two years of data between 2023 and 2025 to develop a combination of LightGBM models to processes second-by-second trade and quote data. The model utilizes a wide array of engineered features including multi-timeframe technical indicators, contextual stock attributes, and fundamental data to predict risk-reward ratios over multiple forward time horizons. Performance is benchmarked against the best human traders with simulations incorporating realistic bid/ask execution scenarios. Results show that the best-performing model achieves average daily returns more than 500 times higher than top performing human day traders, suggesting that ML based systems offer a compelling alternative to human day traders in the domain of day trading.

3.1 Introduction

Day trading refers to the practice of buying and selling financial instruments such as equities, cryptocurrencies, or forex within a single trading day. These trades aim to capitalize on short-term price volatility and are often held for only seconds or minutes. Day traders typically focus on highly liquid assets to facilitate rapid execution and to ensure that there exists sufficient market depth for timely exits.

Unlike longer-term investment strategies that base decisions on fundamental analysis, day trading predominantly relies on technical analysis. Technical Analysis (TA) involves studying price charts, trading volume, and a variety of other statistical indicators to identify patterns that suggest future price movement. TA assumes that historical price and volume is all the information needed to predict future price movements. Fundamental Analysis (FA), on the other hand, is more commonly used by long-term investors. It focuses on evaluating a financial asset's intrinsic value by analyzing economic and financial factors such as a company's earnings, revenue, and growth prospects. FA makes trading decisions based on the asset

being undervalued or overvalued relative to its current market price. The primary objective of day traders is not to invest in the underlying fundamentals of a company, but rather to capture temporary market opportunities, often triggered by some catalyst such as news events, shifts in market sentiment, or surges in volume identified by TA.

Day trading is frequently confused with high-frequency trading (HFT); however, these trading strategies are not the same. The biggest difference is the execution speed required to execute trades. HFT is 100% automated and executes trades in microseconds or milliseconds, much faster than a humans reaction time, targeting minuscule arbitrage opportunities across exchanges for incremental but frequent profits. In contrast, day trading relies on human judgment and predictive insights, with positions held longer than one second; in this study, trades last on average around 30 minutes. This necessitates accurate short-term price forecasting and risk management, rather than purely speed-based arbitrage where the fastest player is guaranteed to make a profit. Nonetheless, both HFT and day trading enhance market liquidity and efficiency by narrowing bid-ask spreads, and thereby facilitating more accurate asset pricing. This crucial value allows investors to buy and sell assets quickly and at desirable prices.

Despite its potential profitability, day trading is considered highly risky, and most day traders end up losing money. Most of these traders incur losses due to psychological and cognitive limitations. Emotional biases such as fear, greed, and confirmation bias frequently impair human decision-making, leading to suboptimal trading outcomes where traders essentially end up just gambling their money like a casino in Las Vegas. Additionally, human traders face inherent constraints in information processing capacity, resulting in analysis paralysis when attempting to interpret excessive data simultaneously. These challenges are not unique to day trading but exist in all trading activities. This motivation has led to the increased interest and use of algorithmic trading as well as the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques to algorithmic trading. While algorithmic and ML methods have been extensively explored for medium and long-term trading horizons, their use in replicating or outperforming human day traders remains underexplored.

This paper aims to bridge this research gap by employing state-of-the-art time series machine learning methodologies outlined by Hall (2025) to replicate and evaluate day trading strategies. Specifically, this study investigates whether advanced ML algorithms can consistently outperform human day traders, offering insights into the capabilities and limitations of AI-driven approaches within this unique trading environment.

The remainder of this paper is structured as follows: **Section 2** reviews related literature, **Section 3** outlines the research methodology, **Section 4** presents the results of experimentation, **Section 5** provides a discussion of findings, and **Section 6** concludes the study and provides insight into future directions.

3.2 Related Works

Day trading has historically faced scrutiny regarding its profitability compared to long-term investment strategies. Many financial advisors advocate for long-term investing strategies and emphasize the importance of backing trading decisions with fundamentals over short-term speculation. Prominent investors like Warren Buffett advocate for investing based on company fundamentals and maintaining positions over an extended period. This philosophy has historically outperformed more active trading approaches. Nevertheless, the question remains: can state-of-the-art ML techniques overcome the limitations that often lead human day traders to underperform, or can they match or exceed the consistently profitable subset of day traders?

To contextualize the risks and outcomes of day trading, Barber et al. (2014) conducted a comprehensive 14-year study of individual traders in Taiwan. Their findings revealed that fewer than 1% of day traders consistently achieved positive net returns when accounting for transaction fees. Notably, the top 500 traders earned an average gross return of 61.3 basis points (0.613%) per day, which fell to 28.1 basis points (0.281%) after fees. Similarly, Jordan and and Diltz (2003) suggest that about twice as many day traders lose money as make money and about one trader in five is more than margin-ally profitable.

Given the thin profit margins and significant risks, successful day trading demands precise decision-making and a variety of analytical tools. Human day traders typically employ a range of technical strategies rather than relying on fundamental analysis. While fundamental indicators such as earnings per share (EPS), return on equity (ROE), and operating margins are useful for long-term investing as shown by Khanpuri, Darapaneni and Paduri (2024), they are less relevant for intraday trading.

Instead, day trading decisions are made through technical indicators. This paper incorporates several widely used tools, including the Moving Average Convergence Divergence (MACD), with both the standard 12-26-9 configuration (Eric, Andjelic, & Redzepagic, 2009) and the 3-10-16 variation popularized by Linda Bradford Raschke (Rosenbloom, 2010). Other popular indicators used include the Relative Strength Index (Rodríguez-González et al., 2010), Bollinger Bands (Day et al., 2023), and Exponentially Weighted Moving Average Indicators (Deng & Sakurai, 2014).

Beyond indicators, Wang and Wang (2019) discuss the feasibility of candlestick pattern strategies, while Chan et al. (2022) emphasize the relevance of support and resistance levels in algorithmic trading. This paper also incorporates Volume Weighted Average Price (VWAP), as suggested by Zhou, Kalev and Frino (2020), and Volume Profile techniques, including the Point of Control (POC) as detailed by Rizvanov et al. (2024). Trade types, such as odd lot trades explored by Johnson (2014) and intermarket sweep trades examined by McInish, Upson and Wood (2014), are shown to impact market dynamics by indicating potential volatility and thus are considered in this study.

The increasing complexity and speed of computers have led many investors to transition from manual strategies to algorithmic trading where computer programs and algorithms execute trades in the financial marketplace. By 2009, algorithmic systems accounted for approximately 73% of U.S. stock market volume, primarily driven by institutional traders and managers of hedge funds, pension funds, and mutual funds as documented by Hendershott, Jones and Menkveld (2011). More recently, ML and deep learning (DL) techniques have become dominant within algorithmic trading systems. Henrique, Sobreiro and Kimura (2019) survey 57 studies employing ML models to forecast various market variables including price direction, returns, risk, and volatility, minimums, maximums, or a combination of these variables. These

studies focus primarily on daily or longer timeframes. One example of this is Patel et al. (2015) who compare the predictive performance of several models (ANN, SVM, RF, Naïve Bayes) on two stocks using 10 technical indicators over a decade of daily data. Kumbure et al. (2022) also conduct a large survey in this field as they provide analysis of 138 papers, finding that most focus on daily predictions, with only limited attention paid to intraday trading.

These limited papers that focus on ML applications at intraday intervals include Sun et al. (2019) where experiments are carried out on intraday movements of S&P 500 data and Borovkova and Tsiamas (2019) who examine 22 different stocks on 5 minute data intervals aggregated from high frequency trade data. Huddleston, Liu and Stentoft (2020) demonstrate that tree-based models can effectively predict 5-minute equity returns even after accounting for transaction costs. Taroon et al. (2020) push the horizon further, modeling next-minute price movements in the SPDR S&P 500 ETF, reinforcing that a model can be used to successfully predict price movements in stocks as granular as a one-minute time interval. Labiad, Berrado and Benabbou (2016) take this one step further by using tick-by-tick data and classifying 10-minute forward price movements as binary outcomes (up/down) on individual stocks from the Maroc Telecom stock exchange. They employ Random Forests and Gradient Boosting Machines, an approach similar to the one taken in this paper.

Among deep learning methods, Long Short-Term Memory (LSTM) networks have emerged as particularly effective. Li and Bastos (2020) identify LSTM as the most widely adopted DL technique in stock prediction, corroborated by Fischer and Krauss (2018), where LSTM outperforms traditional models like RF and DNN when models are only given past price sequences. This suggests that LSTMs are especially adept at modeling time dependencies without requiring extensive feature engineering. More recently, large language models (LLMs) have been applied to financial sentiment analysis. Fatouros et al. (2023), for example, use ChatGPT to enhance sentiment-based trading signals which showcases a novel frontier in the integration of AI and the stock market.

While promising in certain applications, the existing research suffers from some critical limitations. As shown, many previous models are stock-specific, lack sufficient feature diversity (particularly technical

indicators), or are constrained to longer timeframes making them unsuitable for day trading. Simultaneously, human day traders remain limited by cognitive biases, slower execution speeds, and inability to process large data volumes in real time. This paper addresses these gaps by leveraging state-of-the-art ML models on a high-frequency, feature-rich dataset to emulate and assess day trading strategies with second-by-second decision-making precision.

3.3 Methodology

This study employs a comprehensive methodological framework encompassing data acquisition, feature engineering, model development, and performance evaluation. The goal of this study is to replicate, and potentially enhance, the decision-making process of an experienced day trader through the application of ML algorithms trained on a feature-dense financial dataset. The methodology is structured to mimic the typical workflow of technical day trading, while leveraging the scalability and pattern recognition capabilities of modern ML systems.

3.3.1 Data Collection and Preprocessing

To build a dataset representative of real-world day trading conditions, all U.S. equities were queried for the period spanning January 1, 2023 to January 17, 2025 using the Alpaca Markets API. Initially, hourly OHLCV (Open, High, Low, Close, Volume) data were retrieved, including after-hours trading, and subsequently aggregated into daily OHLC candles. To focus on high-volatility trading opportunities that are consistent with the preferences of technical day traders, only date-symbol pairs satisfying the following three conditions were retained:

1. **Price Range Filter**: Restricts attention to stocks whose opening price fell between $0.60 and $30.00. This range targets lower-cap, more volatile equities, which are frequently favored by day traders due to their potential for substantial intraday movement. By setting a lower bound above $0.60, the filter

also helps exclude most penny stocks which often suffer from extreme illiquidity, wide bid-ask spreads, and heightened susceptibility to manipulation.

2. **Liquidity Filter**: Retains only those stocks trading at least 200,000 shares on the given day. This helps eliminate stocks where bid-ask spreads and slippage would otherwise distort the practical feasibility of short-term strategies.
3. **Volatility Filter**: Only include days where the intraday price range exceeded 15%. This ensures that only days exhibiting substantial price fluctuation, and thus potential profit windows, are considered.

After applying these criteria, the dataset was narrowed from the full U.S. equity universe to a specific set of 82,883 symbol-date pairs. To prevent bias introduced by structurally distinct instruments, the dataset was narrowed to exclude all non-common stock equities including ETFs and preferred shares resulting in 81,322 valid pairs.

3.3.1.1 Granular Data Construction

To construct a robust dataset for modeling high-frequency trading behavior, high-resolution intraday data was collected from the Securities Information Processor (SIP) feed, accessed via the Alpaca Markets API, which consolidates trade and quote data across all U.S. exchanges. For each stock under consideration, two consecutive trading days were selected: one characterized by heightened volatility and the immediately preceding day. For both days, the complete set of raw SIP-reported quotes and trades was retrieved, capturing every market event timestamped to the millisecond. This tick-level data was aggregated into 1-second bins, each containing OHLC values, along with a comprehensive set of trade and quote statistics necessary for technical indicator computation. Adjustments were applied in cases of stock splits between the two days to ensure continuity in price and volume metrics.

Further filtering was applied to maintain trading realism and avoid introducing sampling bias due to the initial inclusion criteria of a stock. Specifically, a sample was discarded if:

- Total daily volume for the specific stock was not yet at 200,000 shares,

- Intraday price fluctuation had not yet crossed 15%
- Liquidity thresholds were not met. These included:
    - At least 10 individual trades of 5,000+ shares in the last 1 minute, or
    - At least 15 trades of 10,000+ shares in the last 5 minutes and at least 5 trades in the last 1 minute

These constraints help simulate realistic trading conditions under which a human trader could feasibly engage with the market.

3.3.2 Feature Engineering

The feature set consists of 506 features, designed to mimic how human traders assess opportunities using only price action and historical data. Many retail day traders do not incorporate news-based data in real time, instead depending on price action, momentum, volume dynamics, and key technical indicators. Therefore, no sentiment analysis or natural language inputs were utilized in model construction.

3.3.2.1 Fundamental Features

Although not used for real-time decision-making, fundamental variables are incorporated to provide context and assist the model in differentiating between different categories and types of stocks. These features, sourced from EOD Historical Data (EODHD), include:

- **Structural attributes**: shares outstanding, sector, industry, country of headquarters, time since IPO, and time since last stock split
- **Financial**: earnings per share (EPS), revenue, net income, free cash flow, debt-to-equity ratio, gross margin, operating margin, return on equity (ROE), EPS growth rate, price-to-earnings (PE), price-to-book (PB), and price-to-earnings growth (PEG) ratio.

All features reflect only the financial data available in balance sheets, income statements, and cash flows prior to the evaluated trading day, maintaining temporal consistency.

3.3.2.2 Market Context and Time Features

To estimate the macro-market environment, intraday SPY (S&P 500 ETF) movements were included, along with daily exponential moving averages (EMAs) of SPY. These indicators serve as proxies for market sentiment and help gauge the directional bias of the broader indices. Intraday time and day-of-week features were used to capture seasonal trading behavior, as certain hours and weekdays are statistically more volatile or liquid than others. Notably, day trading activity is notoriously concentrated at the beginning of the trading day as stocks tend to exhibit significantly higher volume and volatility during these hours. This surge in early-session activity reflects rapid price discovery and heightened trader participation, making it crucial for the model to account for temporal patterns like this when learning market dynamics.

3.3.2.3 Technical Features

To mimic multi-timeframe candlestick chart analysis, features were derived across multiple of the most popular day trading windows: 10-second, 1-minute, 2-minute, 5-minute, 15-minute, and 30-minute intervals.

Feature groups include:

- **Candlestick chart metrics**: size and direction of last two candles, including relative size comparison, and wick and body ratio
- **Technical indicators**: Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Ichimoku Cloud Indicator, Stochastic Oscillator, Bollinger Bands, Williams Alligator Indicator, Aroon Indicator, Average Directional Index (ADX), On-Balance Volume (OBV), Klinger Volume Oscillator, Rate of Change (ROC), Accumulation/Distribution Line (ADL),

Volume Weighted Average Price (VWAP), Exponential Moving Average (EMAs) including EMA slopes, and volume profile

- **Quote and trade statistics**: spread width, quote imbalance, bid/ask momentum, odd-lot vs round-lot trade behavior, intermarket sweep order activity
- **Pattern recognition features**: recent pullbacks, breakouts, and directional streaks
- **Support and resistance levels**: recent pivot points and duration since last price interaction with those levels
- **Psychological barriers**: proximity to half-dollar and whole-dollar levels was used to reflect psychological support/resistance zones
- **Daily context features**: longer-term technical statistics (52-week high/low, 3-month trends, 1-month trends, 1-week trends, previous-day performance, relative volume, and average true range) were included to provide broader trend context for each individual stock.

Most numerical indicators were normalized using min-max scaling to maintain comparability across assets, calculated as:

$$x'_t = \frac{x_t - cummin(x)_t}{cummax(x)_t - cummin(x)_t}$$

Where:

- x_t is the raw feature value at time *t*
- $cummin(x)_t$ is the minimum of *x* from time 0 to *t*
- $cummax(x)_t$ is the maximum of *x* from time 0 to *t*
- x'_t is the normalized feature value at time *t*

3.3.2.4 Target Variable Construction

In order to replicate the decision framework used by day traders who seek trades with favorable risk-reward profiles, this study does not predict a single price or direction. Instead, it forecasts the maximum upward and downward price movements over short horizons: 30 seconds, 1 minute, 2 minutes, 3 minutes, 4 minutes, and 5 minutes after each second-level data point. Each of these 12 prediction targets (6 upswings and 6 drawdowns) is trained using two normalization strategies:

Percentage Change, calculated as:

$$Return_t = \frac{P_{t+\Delta} - P_t}{P_t} \times 100$$

Where:

- P_t is the price at time *t*
- $P_{t+\Delta}$ is the future price at the forecast horizon Δ seconds ahead
- $Return_t$ *is th*e Percentage return between the current and future price

Min-Max Scaling, calculated as:

$$Return_t = \frac{P_{t+\Delta} - P_t}{cummax(P)_t - cummin(P)_t}$$

Where:

- P_t is the price at time *t*
- $P_{t+\Delta}$ is the future price at the forecast horizon Δ seconds ahead
- $cummin(P)_t$ is the minimum price from time 0 to *t*
- $cummax(P)_t$ is the maximum price from time 0 to *t*
- $Return_t$ scales the movement relative to the observed price range up to time *t*

This produces a total of 24 target variables (2 metrics × 6 horizons × 2 transformations), and these outputs are used to compute a risk-reward ratio for every potential trade opportunity:

$$RR_t = \frac{\sum_{i=1}^{6} w_i \cdot MaxUpswing_i}{\sum_{i=1}^{6} w_i \cdot MaxDrawdown_i}$$

Where:

- RR_t is the risk-reward ratio at time *t*
- w_i is the weight assigned to each horizon where weighted w_i (WRR) decreases for longer time horizons, placing greater emphasis on near-term movements and equal weighted w_i (EQRR) assigns each horizon the same weight (w_i = [21, 19, 18, 16, 14, 12] or [16.67, 16.67, 16.67, 16.67, 16.67, 16.67])
- $MaxUpswing_i$ is the maximum upward price movement over horizon i
- $MaxDrawdown_i$ is the maximum downward price movement over horizon i

3.3.2.5 Model Architecture and Training

All predictive models were implemented using LightGBM (Ke et al., 2017), a gradient boosting framework optimized for speed and memory efficiency. LightGBM was selected based on empirical findings from Hall (2025) that demonstrate its performance superiority to deep learning methods on large-scale time-series data. Furthermore, its ability to handle high-dimensional feature spaces without explicit feature selection is beneficial for this application. Moreover, the M5 (Makridakis, Spiliotis, & Assimakopoulos, 2022) and M6 (Makridakis et al., 2024) forecasting competitions top performing submissions demonstrate that a combination LightGBM models consistently outperforms other types of ML or DL models.

Each of the 24 target variables is modeled independently using a separate LightGBM regressor and trained on 80% of the available data. Hyperparameter tuning was conducted using the Optuna framework, leveraging Bayesian optimization to balance predictive accuracy with computational cost. Training was

executed on an Amazon EC2 r7i.16xlarge instance, equipped with 512 GB of RAM and 64 vCPUs. This configuration enabled the entire dataset to be held in memory, which in turn allowed LightGBM to exploit its highly parallelized CPU-optimized algorithms for efficient training.

3.3.2.6 Evaluation and Testing Strategy

Model profitability was evaluated on a 20% holdout set. The first 10% of this set was used to calibrate trade entry/exit thresholds based on predicted risk-reward ratios, and the final 10% served as a pure test set to assess model performance in real-time conditions. Instead of selecting the threshold combination that yields the absolute highest profit, this study selects the threshold combination that is most centralized within a broader region of stable profitability. This approach promotes generalizability by reducing the risk of overfitting to a narrowly optimal point. Various experiments were conducted to find the most optimal model using the following metrics:

- Profitability under different risk-reward thresholds
- Sharpe Ratio
- Max Drawdown
- Profit Factor (The ratio of stocks that were positively traded per day to the ratio of negatively traded stocks per day)
- Profitability comparison to human day traders

3.4 Results

To evaluate the efficacy of machine learning in day trading, a series of simulation models was constructed and tested on unseen data from the final 10% of the dataset and optimized using a separate 10% validation partition. The performance of these models is benchmarked against the findings of Barber et al. (2014), one of the most comprehensive studies on human day traders to date. According to this study, top-performing human traders average 28.1 basis points (bps) of profit per day after fees. This figure serves as a baseline

for evaluating the profitability of the following simulations. Additionally, Chague, De-Losso and Giovannetti (2019) conducted a study on day traders in the Brazilian equity index futures market, reporting that only 3% of traders were profitable after fees. The most successful trader in their sample of 1,551 individuals earned an average of $310 per day. While this study does not express profitability in bps and is therefore not used as a direct benchmark, it reinforces that the findings in Barber et al. (2014) have not changed significantly in recent years. This further supports the validity of the benchmark used in the study.

Model 1: Idealized Execution Test

This initial experiment is designed as a baseline sanity check to determine whether the model can accurately identify profitable trading opportunities under idealized conditions. Entry and exit prices are defined using the close price of the current second as the entry point and the close price at the model-determined exit time for the exit point. This represents the price of the last transaction in the stock market. While this approach

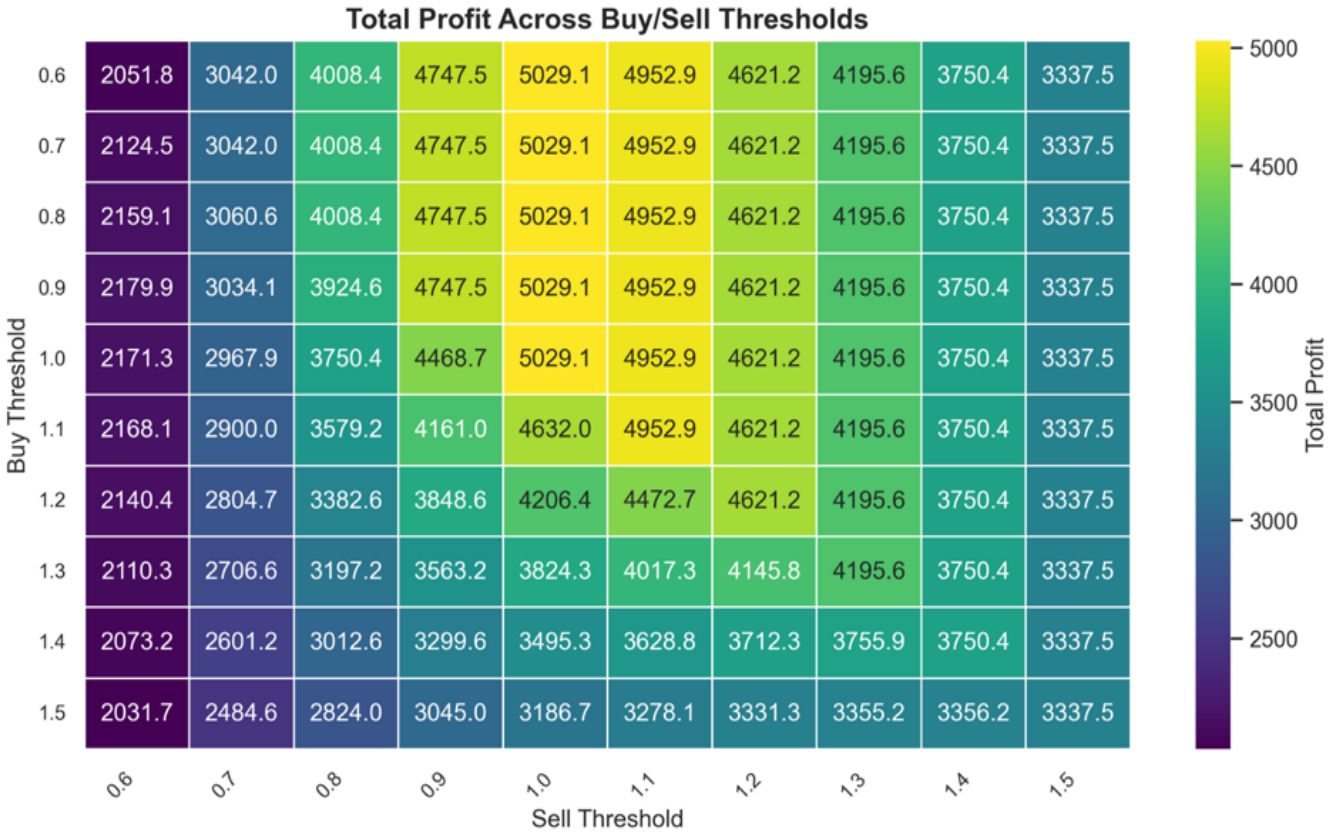

Figure 3.1: Heatmap of buy and sell values used to optimize total profit for Model 1

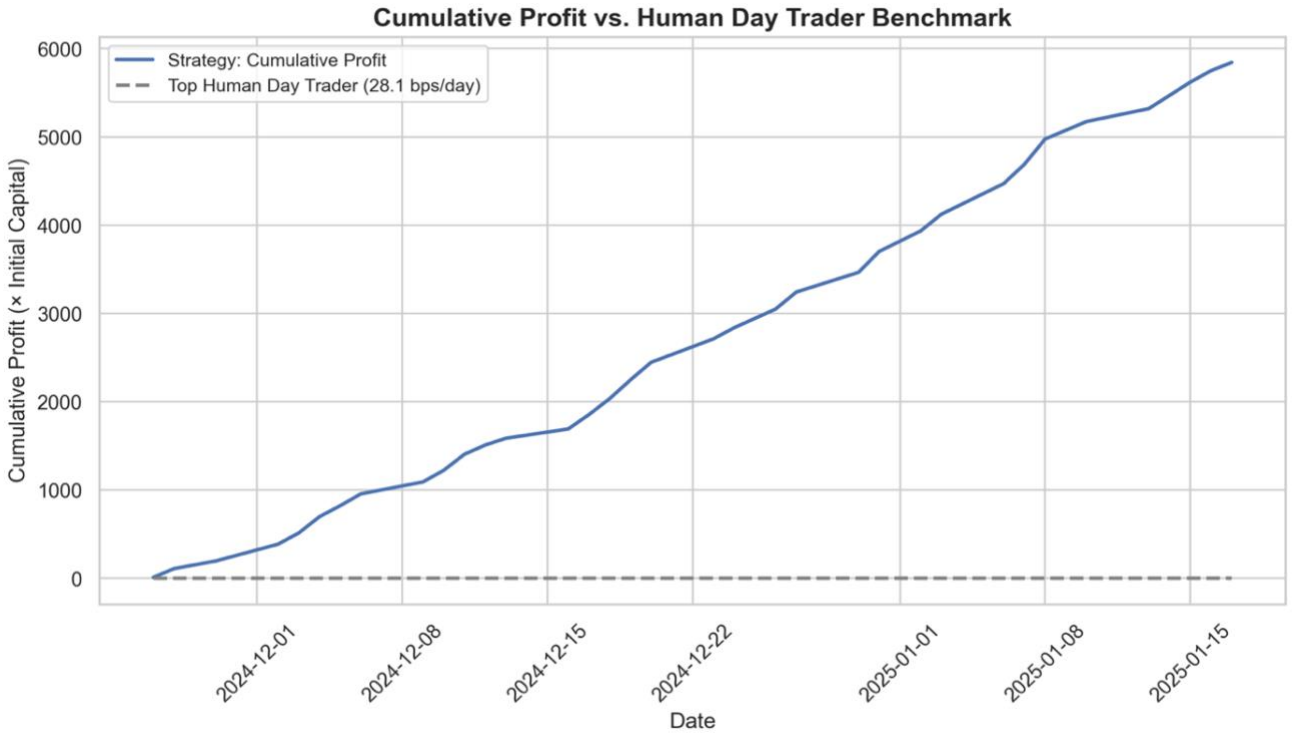

Figure 3.2: Cumulative profit and comparison to human day traders for Model 1

is unrealistic in a live trading environment, it provides a reference point for the upper bound of model potential. Model 1 employs the use of min-max normalized target variables and a WRR ratio. As shown in Figure 3.1, the heatmap of total profit[6] across different buy and sell thresholds identified 1.0 as the optimal value for both. Using these parameters, Model 1 is evaluated on the final 10% of the holdout data.

When evaluated on the test set, Model 1 produced an implausibly high 16,693% average profit per day (Figure 3.2) due to perfect market timing assumptions. This means that, assuming a $100 stake per trade, the model would generate an average daily profit of $16,693. The model also achieved this figure through an unrealistic volume of 31,351 trades daily with zero days of drawdown (Table 3.1) across the entire period. This corresponds to an average return of 1,669,300 bps per day, which exceeds the human benchmark by several orders of magnitude. Although informative as a proof of concept, these results are

[6] The values displayed within the heatmap figures represent the total profit accumulated over the 35-day validation period.

not feasible under real market conditions due to slippage and bid/ask spreads, highlighting the need for a more realistic execution assumption.

Table 3.1: Performance Metrics for Model 1

Metric	Buy Threshold	Sell Threshold	Average Profit Per Day	Max Drawdown	Profit Factor	Sharpe Ratio	Average Trades Per Day
Value	1.0	1.0	166.93	0.0	1.716	40.28	31,351

Model 2: Realistic Execution Test

To address the limitations of Model 1, Model 2 employs a more realistic trading mechanism using quote data. The entry price of a stock position is defined as the last ask price of the current time period. This reflects the price at which a seller is willing to transact, ensuring a trade will occur if the model places a

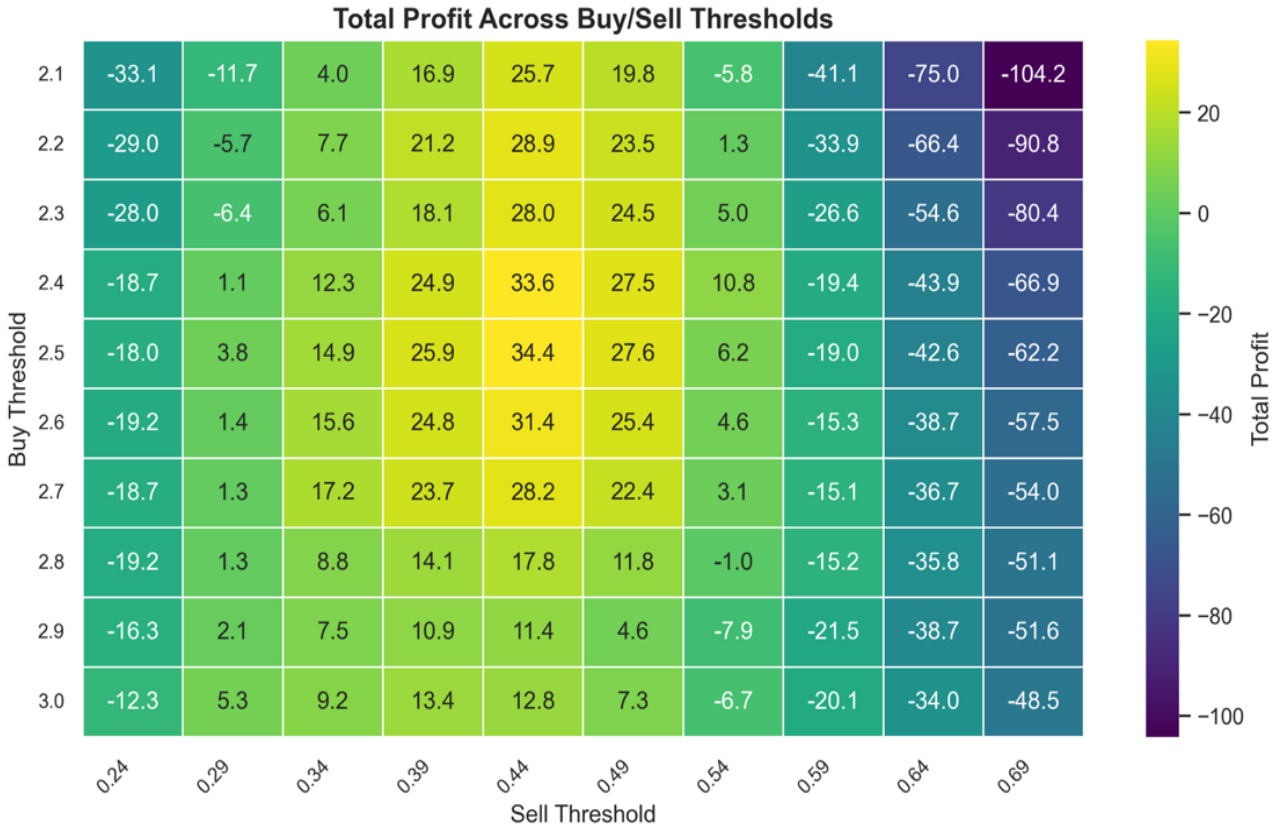

Figure 3.3: Heatmap of buy and sell values used to optimize total profit for Model 2

buy order matching this price. Similarly, the exit price is determined by the last bid price of the exit period, representing a buyer's standing order. A trade is guaranteed if the model places a sell order at this bid price. The model again uses min-max normalization and the WRR ratio calculation. As illustrated in Figure 3.3, optimal buy and sell thresholds were determined to be 2.5 and 0.44, respectively. With these settings, Model 2 achieved an average daily return of 1.42 (or 14,200 bps) as illustrated in Table 3.2. This still vastly outperforms human traders by a factor of approximately 500x (Figure 3.4).

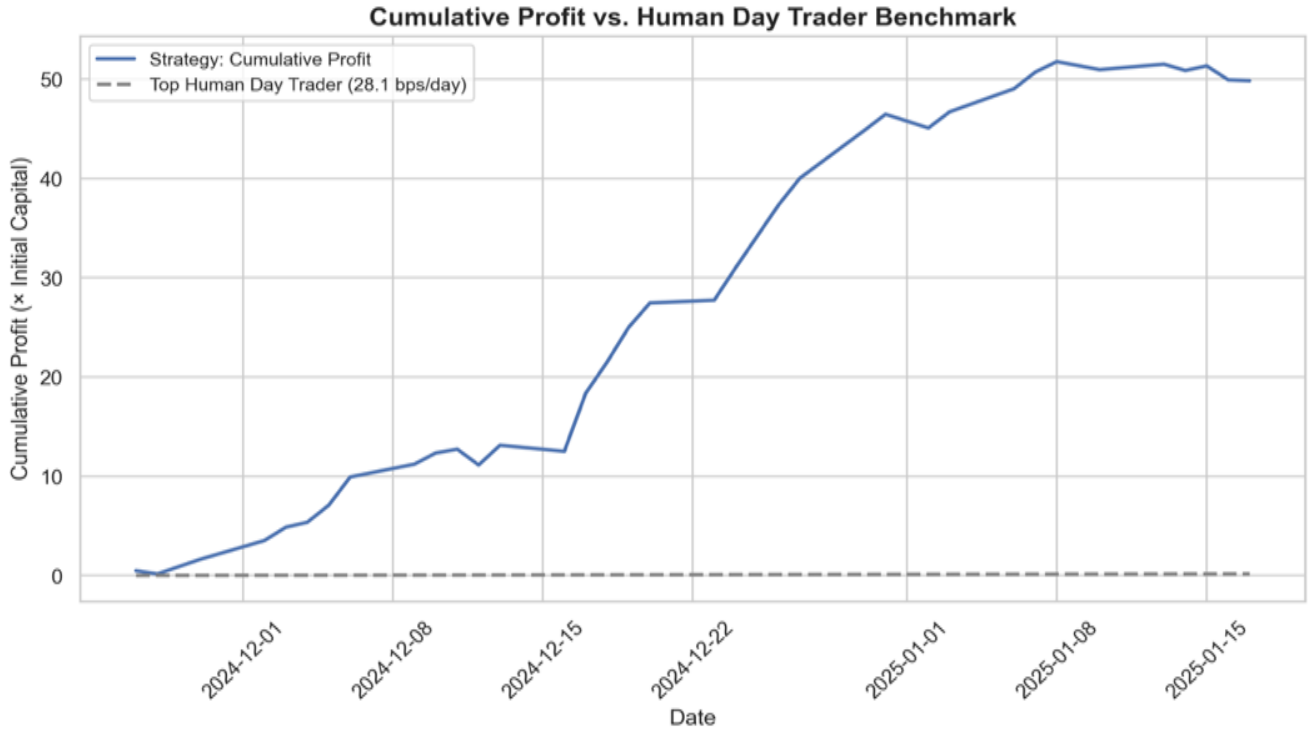

Figure 3.4: Cumulative profit and comparison to human day traders for Model 2

Table 3.2: Performance Metrics for Model 2

Metric	Buy Threshold	Sell Threshold	Average Profit Per Day	Max Drawdown	Profit Factor	Sharpe Ratio	Average Trades Per Day
Value	2.5	0.44	1.42	-1.928	2.884	11.95	1,069

Model 3: Percentage Change Normalization

Model 3 tested an alternative approach to target normalization by applying percentage change rather than min-max normalization. This model continued to use bid/ask quotes for entry and exit execution and WRR ratio for risk/reward combination. As illustrated in Figure 3.5, optimal buy and sell thresholds were determined to be 2.1 and 0.44, respectively. Testing shows that although this model remained profitable, with an average daily profit of 0.53 (5,300 bps) as illustrated in Figure 3.6, performance declined relative to Model 2, particularly in terms of drawdown, which increased to -5.34 (Table 3.3). These findings suggest that min-max normalization yields superior signal quality for the learning algorithm both in terms of profitability and volatility.

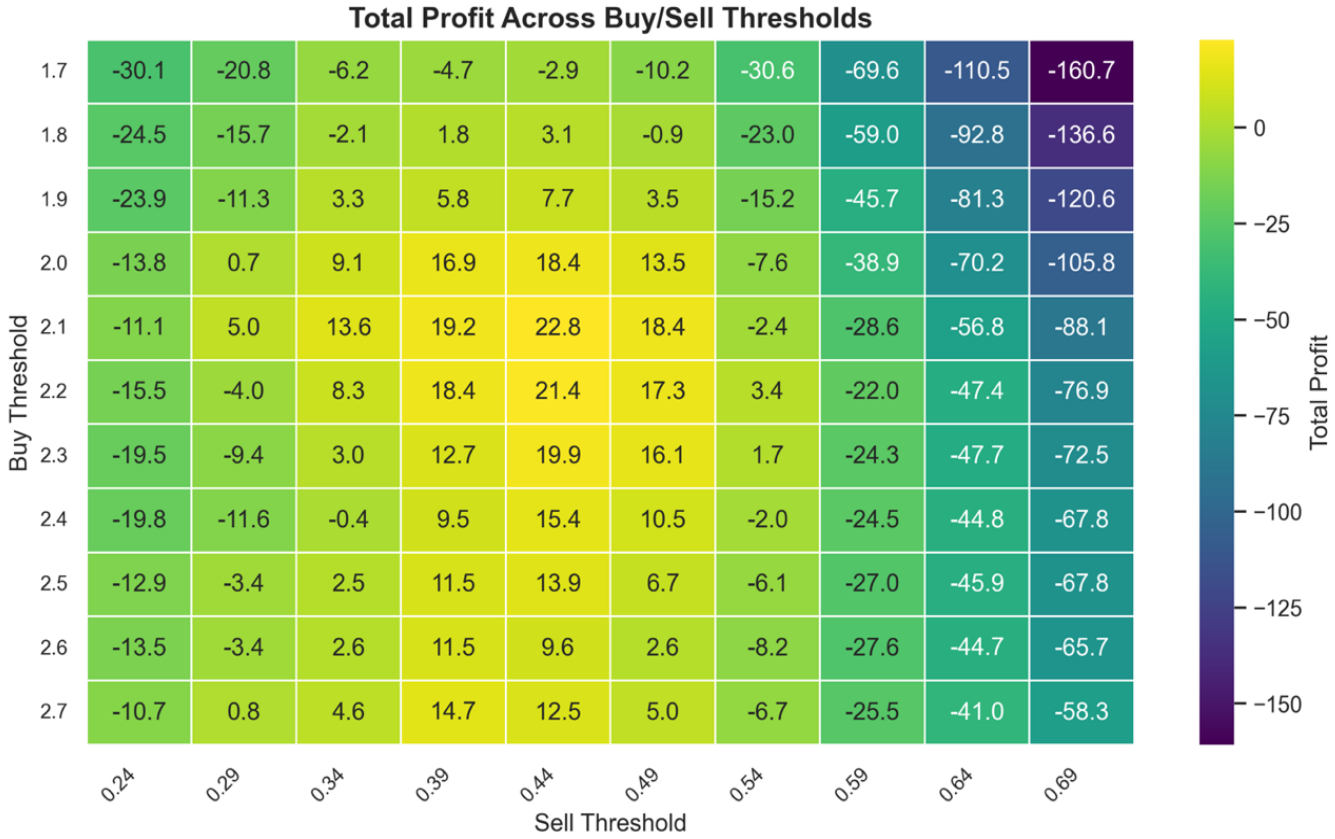

Figure 3.5: Heatmap of buy and sell values used to optimize total profit for Model 3

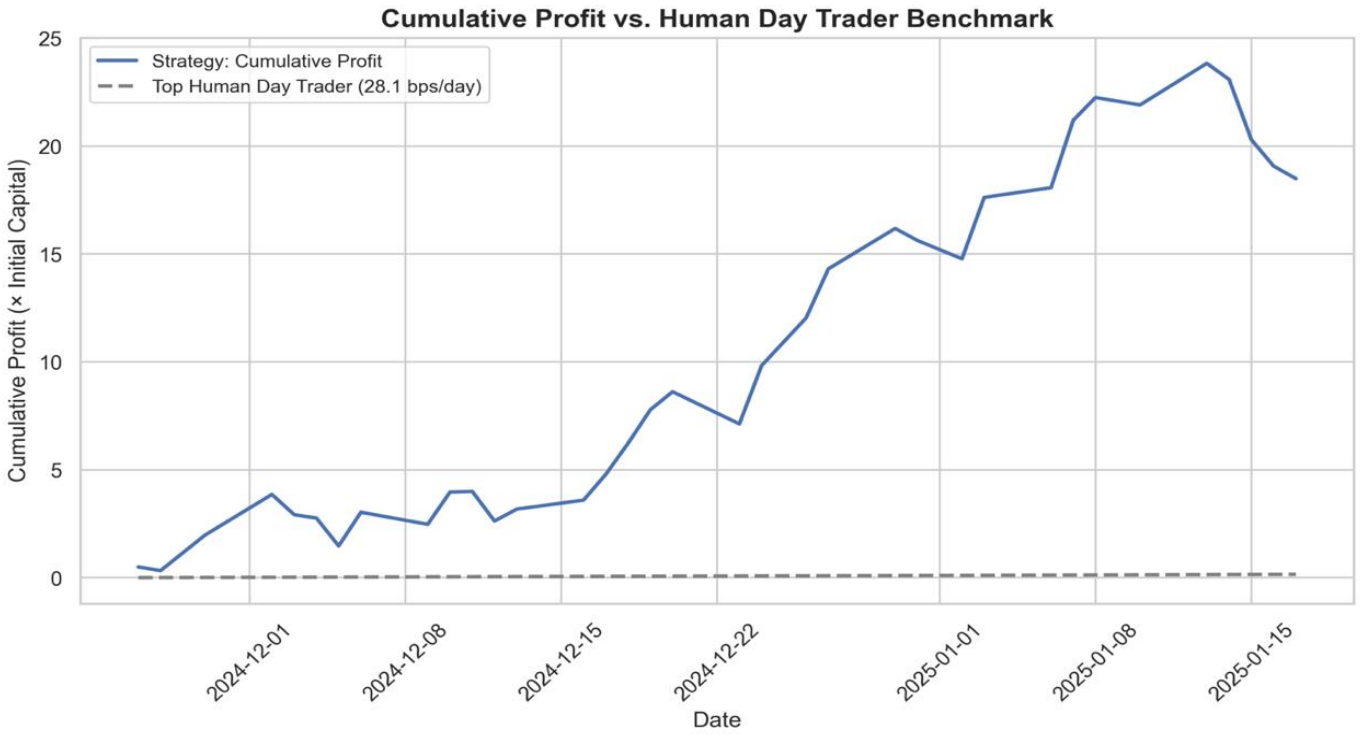

Figure 3.6: Cumulative profit and comparison to human day traders for Model 3

Table 3.3: Performance Metrics for Model 3

Metric	Buy Threshold	Sell Threshold	Average Profit Per Day	Max Drawdown	Profit Factor	Sharpe Ratio	Average Trades Per Day
Value	2.1	0.44	0.53	-5.34	2.67	5.82	883

Model 4: Hybrid Normalization

To explore whether combining min-max and percentage change normalization might offer complementary benefits, Model 4 averaged the RR ratios derived from both methods. Model 4 also uses the same model combination weighting (WRR) techniques and bid/ask prices as previous models, and the resulting heatmap of this is shown in Figure 3.7. Despite this hybrid strategy, the results underperformed compared to Model 2, with average daily profits dropping to 0.43 (visualization of this cumulative profit shown in Figure 3.8) and drawdown dropping to -4.77 (Table 3.4). This indicates that the addition of percentage-based features

may dilute model performance, and that min-max regularization is the best target normalization technique as a standalone technique.

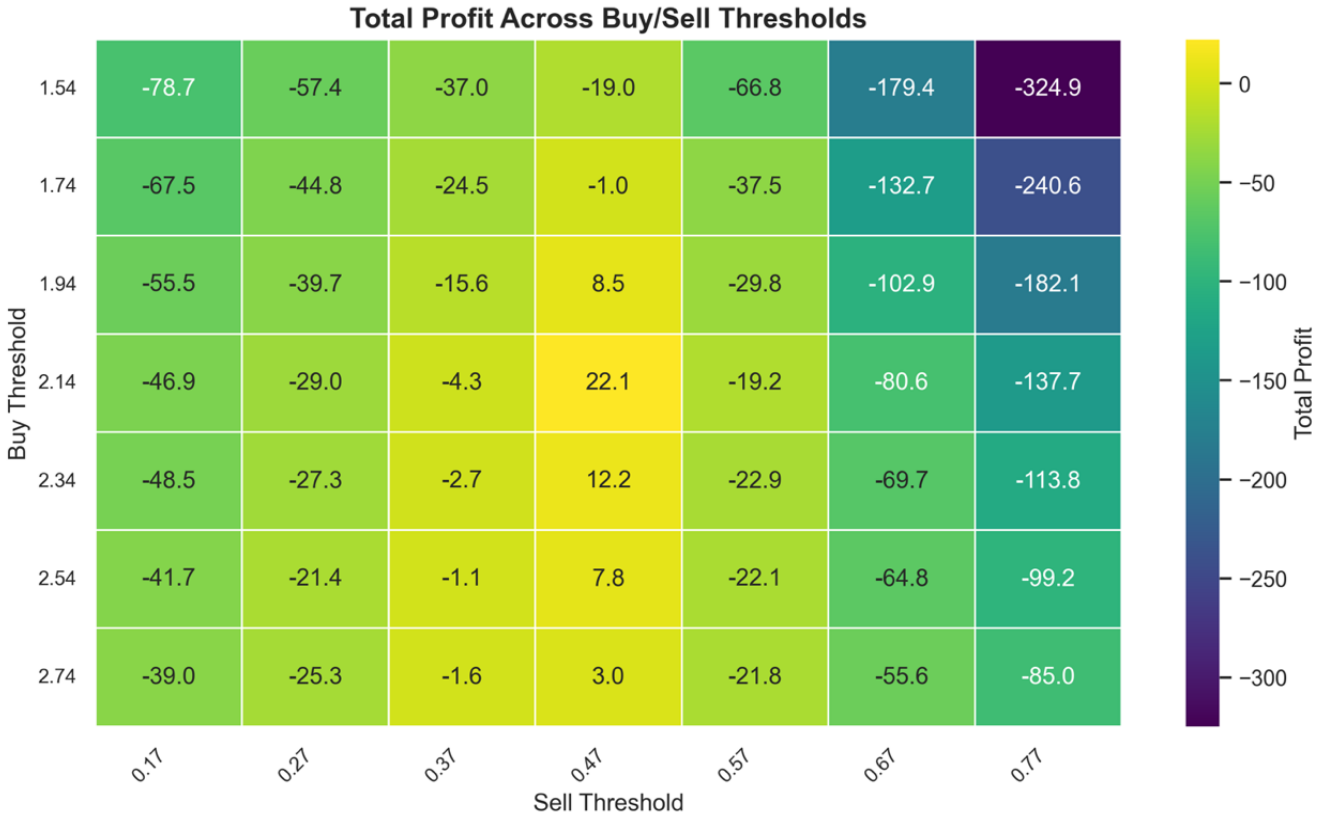

Figure 3.7: Heatmap of buy and sell values used to optimize total profit for Model 4

Table 3.4: Performance Metrics for Model 4

Metric	Buy Threshold	Sell Threshold	Average Profit Per Day	Max Drawdown	Profit Factor	Sharpe Ratio	Average Trades Per Day
Value	2.14	0.472	0.43	-4.77	2.61	4.08	1,213.17

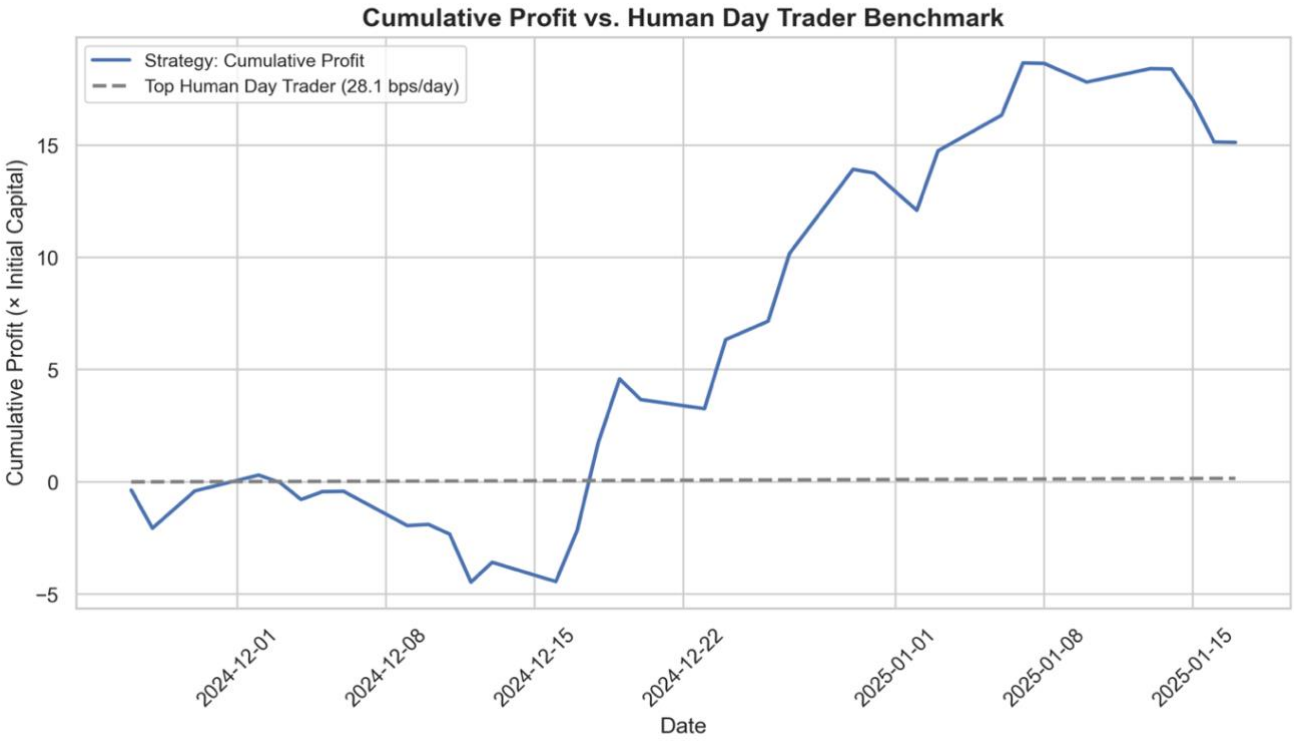

Figure 3.8: Cumulative profit and comparison to human day traders for Model 4

Model 5: Equal Weighting of RR Components

Finally, Model 5 investigates the effect of the decaying weighting scheme (WRR) used in the computation of the RR ratio in the preceding four models, where more recent periods are weighed more heavily than longer periods, by implementing the EQRR ratio as a point of comparison. The model uses min-max normalization and bid/ask prices for execution, and the resulting heatmap is shown in Figure 3.9. The performance statistics are displayed in Table 3.5 and the cumulative profit graph is shown in Figure 3.10. The outcome was the best performance among all five models, with an average daily profit of 2.00 (20,000 bps), moderate drawdown, and the highest Sharpe ratio (15.78) observed in the study. Since this model demonstrated the best performance, an additional statistic was calculated to provide further context: the average time per trade, which was 38 minutes and 29 seconds. Overall, this suggests that an equal weighting approach to RR calculation may provide a more stable and effective signal for day trading in this model.

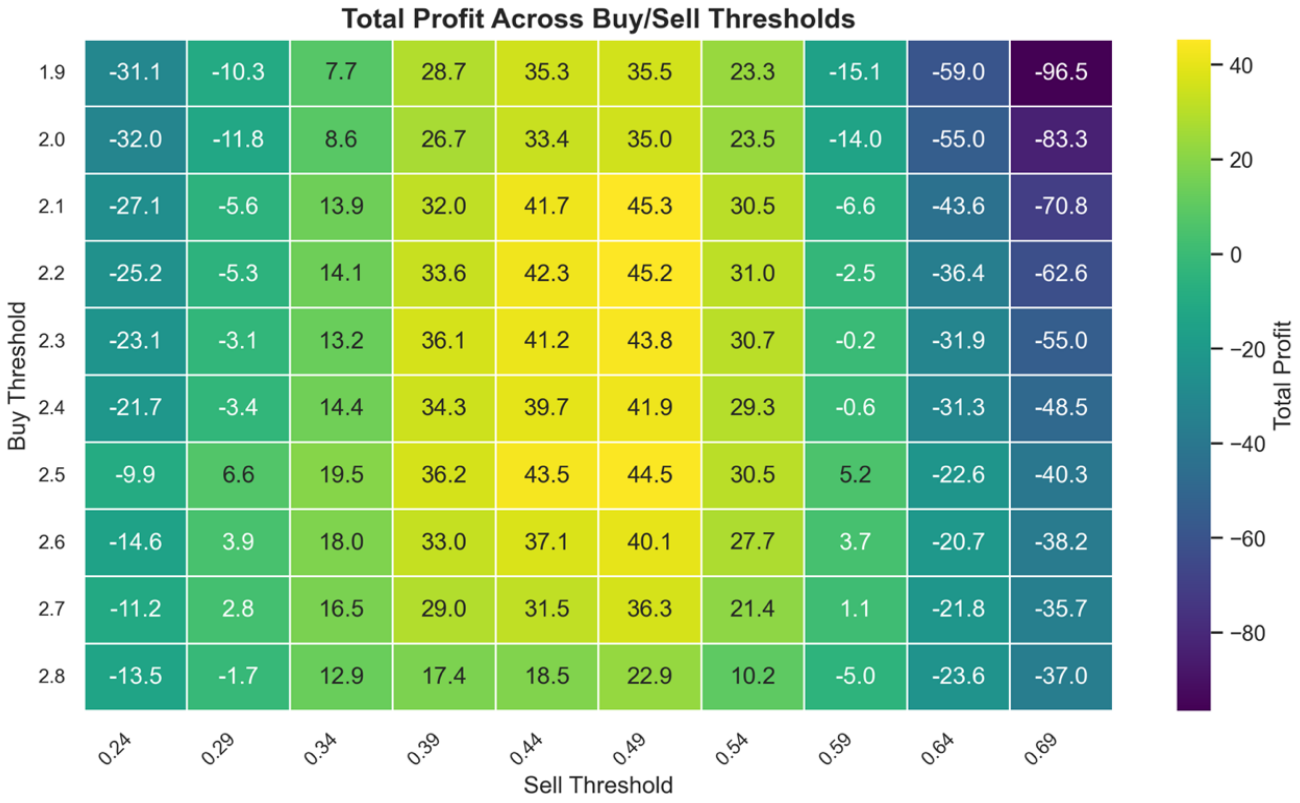

Figure 3.9: Heatmap of buy and sell values used to optimize total profit for Model 5

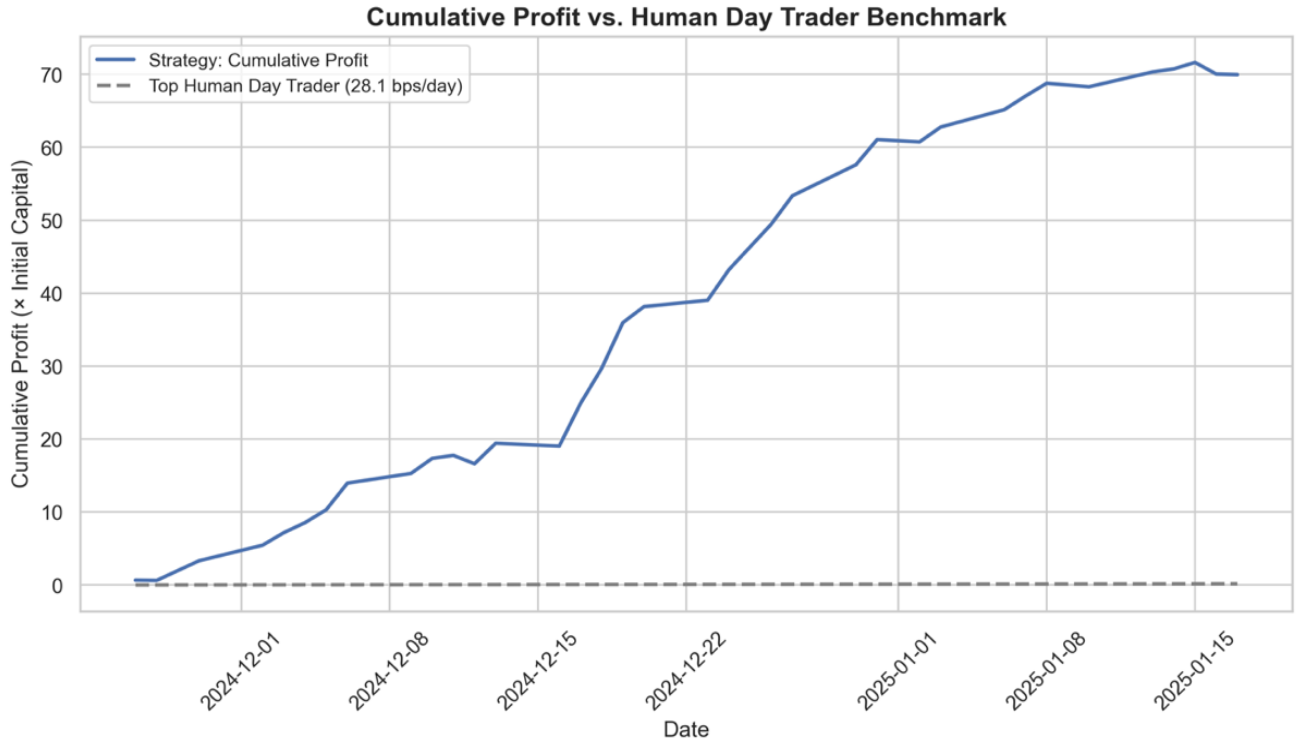

Figure 3.10: Cumulative profit and comparison to human day traders for Model 5

Table 3.5: Performance Metrics for Model 5

Metric	Buy Threshold	Sell Threshold	Average Profit Per Day	Max Drawdown	Profit Factor	Sharpe Ratio	Average Trades Per Day
Value	2.3	0.44	2.00	-1.67	2.8717	15.78	999

3.5 Discussion

This section interprets the results of the ML simulations conducted in this study, comparing them to human benchmarks and evaluating the impact of various design decisions. The experiments demonstrate that a well-engineered ML system, when applied to high-frequency market data, can significantly outperform human traders in intraday trading. However, these results also expose several limitations that must be addressed before these models are considered viable for live trading scenarios.

3.5.1 Comparing AI Performance to Human Traders

The most striking result from this study is the clear performance advantage that ML-based trading strategies exhibit over human day traders. The best-performing model (Model 5) achieved an average daily return of 2.00 (20,000 bps/day), compared to the 28.1 bps/day benchmark reported by Barber et al. (2014) for elite human traders. Additionally, this outperformance occurred during a period when the S&P 500 declined by 0.4% (40 bps), suggesting that the model's success was largely independent of broader market trends. This performance gap stems from several fundamental differences.

One of these differences is the ML models created here operate on a per-second granularity across the entire trading day, evaluating thousands of opportunities across hundreds of tickers simultaneously. In contrast, human traders are constrained by both attention span and reaction time, often focusing on a handful of stocks per day and often focusing only on the hours immediately following market open. Barber et al. (2014) note that the top-performing human traders typically focus on a few familiar stocks, suggesting that this specialization accounts for their limited capacity, an issue that does not constrain ML models.

Another difference is machines are immune to emotional and cognitive biases that often impair human decision-making. While human traders might hesitate to re-enter the market after experiencing a significant drawdown, the model continues executing trades as designed, recovering losses without hesitation, as evidenced in the simulations. One other difference is ML models can detect microstructure patterns and subtle statistical signals invisible to human perception. These can include rapid shifts in spread dynamics, temporary inefficiencies, or high-probability entry conditions that occur only for a few seconds.

3.5.2 Impact of Target Normalization Techniques

The study found that using min-max normalization on the target variable led to superior model performance and lower variance in returns when compared to percentage change normalization. This is particularly noteworthy because the model's input features were also min-max normalized, which likely led to more coherent signals during training and improved generalization. This may indicate that when both are scaled within the same range, the model can learn patterns more effectively because the relationship between inputs and outputs remains proportionate. This reinforces the importance of keeping the preprocessing techniques of the model aligned and provides insight that opens avenues for further experimentation with alternative normalization schemes. For example, applying a percentage-change normalization to both features and target variables may better capture the relative movements in asset prices and could help improve the predictive capability of the percentage change target normalization technique demonstrated in Model 3.

3.5.3 Impact of Risk-Reward Ratio Weighting

One hypothesis tested in this research was that weighting predictions based on risk-reward ratios, giving higher importance to near-term forecasts, would enhance performance. However, results indicate that this added complexity does not translate into improved outcomes. Model 5, which uses equal weighting, outperforms Model 2, which applies a decaying weight to emphasize nearer-term forecasts. This suggests that the weighting scheme introduces unnecessary noise into the model, signifying that simplicity in

weighting is preferable for short-term trading systems. Equal weighting bases its trading decisions on a more holistic view of future price behavior and thus may allow the model to smooth out erratic signals.

3.5.4 Feature Engineering

The quality and diversity of the feature set, as well as extensive feature engineering, likely serve as the primary driver of the model's success. The following are some interesting insights into the model's most important features.

Among the 24 trained LightGBM models, the industry classification of a stock emerged as the most important feature in every model, with country of origin also consistently appearing in the top 10. This implies that contextual awareness, knowing what type of stock is being evaluated, significantly enhances the model's ability to generalize across assets and allows the model to group stocks by similar behavioral patterns.

Other consistently high-ranking features include:

- Time of day
- Volume Profile metrics, such as the Point of Control (POC) and Value Area, indicating where trading activity is concentrated.
- Volume Oscillator

These findings highlight the critical role of contextual categorization in replicating how human traders identify high-probability setups. They also show that volume-based indicators play a central role in identifying profitable trading setups, especially in intraday markets where liquidity can shift rapidly.

3.5.5 Limitations of Current Model

While the results are promising, the study acknowledges several limitations that impact the real-world applicability of the proposed models. One major constraint is the assumption of idealized execution. In Model 1, trades were executed at closing prices, a clearly unrealistic assumption. Model 2 introduced more

grounded assumptions using bid/ask prices; however, even this approach omits critical market dynamics such as order latency and slippage. These factors can significantly degrade performance in live environments.

Additionally, the model assumes full order fulfillment at the bid or ask price, regardless of the order book depth. In reality, the number of shares available at a given price may be limited and attempting to trade large volumes can move the market against the trader. Even though this was in mind and somewhat accounted for by imposing minimum volume and volatility requirements on the stocks being traded, some of the smaller cap, lower liquidity stocks may still produce unrealistic trade results and may not be feasible in practice. An extension of this work would be to incorporate full historical order book (Level II) data to more accurately model fill probability and slippage.

Furthermore, transaction costs are not accounted for in this study mainly due to the rise of commission-free trading brokers such as Alpaca, which also serve as the source for most of the market data collected for this research. While commission-free brokers may seem attractive from a cost perspective, their execution quality can be inferior due to payment-for-order-flow arrangements, which route trades through less favorable venues. Conversely, commission-based brokers often provide better order execution and prices, which may offset the explicit trading fees. Future studies could incorporate broker-specific execution models to assess how these tradeoffs impact algorithmic performance.

There are also several directions for enhancing the model's capabilities. For instance, incorporating short selling would allow the model to trade in both bullish and bearish marketplaces. In addition, the model currently explores only a limited feature set, and there are virtually infinite combinations of technical indicators, price patterns, and alternative data sources like news data that could be engineered to improve predictive performance and robustness.

3.6 Conclusion

This study set out to investigate whether the state-of-the-art ML techniques for time series analysis laid out by Hall (2025) can be applied in the domain of intraday trading to outperform human traders. To accomplish this task, the study constructs a highly granular, feature-rich dataset that mimics the decision-making process of technical day traders. The dataset is tested through various LightGBM models across different execution assumptions, target normalization strategies, and risk-reward formulations. The findings indicate that a properly engineered model can vastly outperform even the most successful human traders documented in the literature. The best-performing model operates at second-level resolution and can scan hundreds of stocks in parallel. This approach allows the ML model to hold multiple positions at the same time, resulting in nearly 1,000 trades executed per day with an average trade duration of approximately 38 minutes from entry to exit. Results from preliminary experimentation forecast average daily returns of over 500 times greater than the benchmark set by Barber et al. (2014).

Among the various design choices tested, min-max normalization of target variables, equal weighting of predictive horizons, and the inclusion of contextual features like industry, country, and time of day, emerged as critical drivers of performance. In addition to its modeling contributions, this study also provides a detailed, end-to-end methodology for data acquisition, preprocessing, and feature engineering, a component often underexplored in similar research. By outlining the process of sourcing high-frequency trade and quote data, filtering that data, and engineering both technical and contextual features, this paper serves as a practical guide for researchers looking to extend this work.

This study acknowledges the limitations in evaluating the profitability of the models created. While bid/ask-based execution improves realism, it does not fully capture market dynamics such as order book depth, latency, slippage, and partial fills. Future work should incorporate Level II order book data, paper trading simulated environments, and eventually real-world deployment environments to further validate the findings.

References

Barber, B. M., Lee, Y.-T., Liu, Y.-J., & Odean, T. (2014). The cross-section of speculator skill: Evidence from day trading. *Journal of Financial Markets*, *18*, 1-24.

Borovkova, S., & Tsiamas, I. (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. *Journal of Forecasting*, *38*(6), 600-619.

Chague, F., De-Losso, R., & Giovannetti, B. (2019). Day Trading for a Living?

Chan, J. Y., Phoong, S. W., Cheng, W. K., & Chen, Y.-L. (2022). Support Resistance Levels towards Profitability in Intelligent Algorithmic Trading Models. *Mathematics*, *10*(20).

Day, M.-Y., Yirung, C., Paoyu, H., & and Ni, Y. (2023). The profitability of Bollinger Bands trading bitcoin futures. *Applied Economics Letters*, *30*(11), 1437-1443.

Deng, S., & Sakurai, A. (2014). Integrated model of multiple kernel learning and differential evolution for EUR/USD trading. *ScientificWorldJournal*, *2014*, 914641.

Eric, D., Andjelic, G., & Redzepagic, S. (2009). Application of MACD and RVI indicators as functions of investment strategy optimization on the financial market*. *Zbornik Radova Ekonomski Fakultet u Rijeka*, *27*(1), 171-195.

Fatouros, G., Soldatos, J., Kouroumali, K., Makridis, G., & Kyriazis, D. (2023). Transforming sentiment analysis in the financial domain with ChatGPT. *Machine Learning with Applications*, *14*, 100508.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654-669.

Hall, T. (2025). *A Survey of Machine Learning Methods for Time Series Prediction*. University of Georgia.

Hendershott, T., Jones, C. M., & Menkveld, A. J. (2011). Does Algorithmic Trading Improve Liquidity? *The Journal of Finance*, *66*(1), 1-33.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226-251.

Huddleston, D., Liu, F., & Stentoft, L. (2020). *Intraday market predictability: A machine learning approach*.

Johnson, H. (2014). Odd Lot Trades: The Behavior, Characteristics, and Information Content, Over Time. *Financial Review*, *49*(4), 669-684.

Jordan, D. J., & and Diltz, J. D. (2003). The Profitability of Day Traders. *Financial Analysts Journal*, *59*(6), 85-94.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). *LightGBM: a highly efficient gradient boosting decision tree* Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA.

Khanpuri, A., Darapaneni, N., & Paduri, A. R. (2024). Utilizing Fundamental Analysis to Predict Stock Prices. *EAI Endorsed Transactions on AI and Robotics*, *3*.

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, *197*, 116659.

Labiad, B., Berrado, A., & Benabbou, L. (2016, 19-20 Oct. 2016). Machine learning techniques for short term stock movements classification for Moroccan stock exchange. 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA),

Li, A. W., & Bastos, G. S. (2020). Stock Market Forecasting Using Deep Learning and Technical Analysis: A Systematic Review. *IEEE Access*, *8*, 185232-185242.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*(4), 1346-1364.

Makridakis, S., Spiliotis, E., Hollyman, R., Petropoulos, F., Swanson, N., & Gaba, A. (2024). The M6 forecasting competition: Bridging the gap between forecasting and investment decisions. *International Journal of Forecasting*.

McInish, T., Upson, J., & Wood, R. A. (2014). The Flash Crash: Trading Aggressiveness, Liquidity Supply, and the Impact of Intermarket Sweep Orders. *Financial Review*, *49*(3), 481-509.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259-268.

Rizvanov, T., Vlasenko, A., Rizvanov, R., & Chelnokov, Y. (2024). Volume Profile: Development and Application of a TradingView Indicator.

Rodríguez-González, A., Guldrís-Iglesias, F., Colomo-Palacios, R., Gomez-Berbis, J. M., Jimenez-Domingo, E., Alor-Hernandez, G., Posada-Gomez, R., & Cortes-Robles, G. (2010, 2010//). Improving Trading Systems Using the RSI Financial Indicator and Neural Networks. Knowledge Management and Acquisition for Smart Systems and Services, Berlin, Heidelberg.

Rosenbloom, C. (2010). *The Complete Trading Course: Price Patterns, Strategies, Setups, and Execution Tactics*. John Wiley & Sons.

Sun, J., Xiao, K., Liu, C., Zhou, W., & Xiong, H. (2019). Exploiting intra-day patterns for market shock prediction: A machine learning approach. *Expert Systems with Applications*, *127*, 272-281.

Taroon, G., Tomar, A., Manjunath, C., Balamurugan, M., Ghosh, B., & Krishna, A. V. N. (2020, 26-27 Nov. 2020). Employing Deep Learning In Intraday Stock Trading. 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN),

Wang, M., & Wang, Y. (2019). *Evaluating the Effectiveness of Candlestick Analysis in Forecasting U.S. Stock Market* Proceedings of the 2019 3rd International Conference on Compute and Data Analysis, Kahului, HI, USA.

Zhou, H., Kalev, P. S., & Frino, A. (2020). Algorithmic trading in turbulent markets. *Pacific-Basin Finance Journal*, *62*, 101358.

CHAPTER 4

Conclusion

My overarching objective of this thesis was to examine and demonstrate the effectiveness of advanced ML methods within the domain of time series prediction and their applicability to complex, real-world tasks such as financial day trading.

In doing so, this project establishes a comprehensive study that highlights the critical role of domain-specific knowledge, meticulous data preprocessing, and strategic model selection. It proposes a structured, iterative framework that starts with TBML models such as LightGBM due to their scalability and robustness, and then transitions toward more complex DL architectures like RNNs when necessary. Additionally, the study recommends ensemble approaches to model creation and Bayesian optimized hyperparameter tuning. This structured approach to time series prediction aims to guide future researchers by recommending clear, evidence-backed best practices.

This project supports the efficacy of this framework by putting the theoretical insights studied into practice. Using high-frequency financial data, this project demonstrates that modern ML models when applied to carefully constructed, feature-rich datasets can significantly outperform even the best-documented human traders. Key methodological innovations, including min-max normalization, contextual feature integration, and a nuanced approach to predicting stock movements, were crucial to achieving this performance. Additionally, the model's ability to manage many positions in parallel across the entire U.S stock market, executing close to a thousand trades per day with each lasting about 30 minutes, provides a unique advantage compared to human day traders, outperforming them by a factor of 500.

Despite the encouraging predictive results, I acknowledge some of the practical limitations related to execution realism. Future investigations should incorporate advanced market dynamics through Level II order book data and simulated or real-world trading environments. Additionally, future work should explore the efficacy of under researched models in the realm of time series prediction like CatBoost and Transformer architectures especially in this context of financial markets.

Ultimately, this thesis validates the integration of ML into predictive time series applications and demonstrates boundless possibilities for innovation across virtually every industry with time series use cases.