– RESEARCH PROPOSAL –

Market Trade Forecast:
A Probabilistic Approach to Price Trending Analysis

BY

Mabunda Hlulani

(Student Number: 213067605)

Submitted in fulfillment of the requirements for the degree

PGDip. Computer Systems Engineering

At the

Department of Computer Systems Engineering

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

TSHWANE UNIVERSITY OF TECHNOLOGY

Supervisors:

Dr. Oluwasogo Moses Olaifa

Prof. Chunling Du

Date of submission: 31 October 2025

Abstract

Conventional stock forecasting approaches often produce static, point-estimate predictions that overlook the uncertainty and dynamic behavior of intraday price movements. This research proposes an intelligent, probabilistic deep learning framework for forecasting both the directional movement and level attainment of asset prices, grounded in historical high-frequency data. Leveraging Bayesian extensions of LSTM and Transformer models, the system integrates alternative data—including sentiment, ESG indicators, and volatility dynamics—to enhance signal robustness and reversion detection. The framework is designed to be asset-agnostic and deployable across multiple financial instruments, with real-time integration into TradingView via Pine Script pipelines. A systematic literature review of over 260 primary studies identified gaps in uncertainty quantification, deployment readiness, and the inclusion of non-price features. Performance will be assessed using metrics such as the Sharpe ratio, MAE, and calibration error. The final system will be live-tested over 30 trading days, offering a reproducible, explainable, and high-frequency architecture to support decision-making in volatile markets.

1 Introduction

1.1 Background to the Study

The growing complexity and speed of global financial markets have intensified the demand for intelligent, data-driven systems capable of supporting intraday trading decisions. Traditional stock forecasting models—often based on point estimates or deterministic assumptions—struggle to capture the high-frequency, nonlinear, and stochastic nature of price fluctuations. In particular, predicting whether an asset's price will move in a specific direction or revert to a certain level within short time horizons remains an open challenge, yet it is of paramount importance to algorithmic and day traders.

Recent advancements in deep learning architectures—such as Long Short-Term Memory (LSTM) networks and Transformer models—have significantly improved time-series forecasting performance. However, most implementations neglect uncertainty quantification, limiting their utility in high-risk trading environments. Probabilistic modeling techniques, such as Bayesian neural networks, offer a promising solution by providing both predictive estimates and confidence intervals, which are essential for risk-aware financial decisions [1, **?**].

A systematic literature review conducted for this project reveals critical gaps in the domain: fewer than 5% of the surveyed studies explicitly target intraday reversion, probabilistic modeling, or deployment readiness. Most focus exclusively on directional trends without assessing whether price will revert to key levels such as session open or daily mean. Moreover, explainability and scalability remain underdeveloped—only a minority of studies leverage alternative data (e.g., sentiment, ESG signals) or provide mechanisms for real-time model interpretability [**?**, 2, 3].

This study proposes an intelligent trading forecast framework that combines Bayesian LSTM and Transformer models to determine (i) whether the price will move in a particular direction and (ii) whether it will reach specific target levels. These models will be trained on historical time-series and alternative data to ensure robust generalization across diverse market conditions. Designed to be asset-agnostic, the system will be deployable on multiple financial instruments and tested live for 30 trading days via TradingView integration. Real-time inference pipelines will be constructed to support active trading environments, while performance will be evaluated using both probabilistic (e.g., calibration error) and financial metrics (e.g., Sharpe ratio, MAE).

Ultimately, this research contributes a scalable, explainable, and risk-aware forecasting architecture tailored to intraday trading needs, bridging academic innovation with practical deployment across retail and institutional contexts.

1.2 Problem Statement

Despite the rapid advancement of machine learning (ML) and deep learning (DL) applications in financial forecasting, existing models still struggle to deliver reliable and deployable intraday trading intelligence. Most prediction systems focus primarily on directional price movement using static algorithms that ignore the probabilistic nature of market behaviour. The main weakness lies in their deterministic design, which overlooks the likelihood of a price reaching specific levels such as the day's open, previous close, or short-term support and resistance zones. This omission limits the accuracy of trading signals and prevents traders from effectively timing market entries and exits.

Furthermore, the majority of open-source and research-based forecasting frameworks are not deployable across multiple asset classes, which restricts scalability and practical adoption. These systems often depend on historical daily data and lack real-time adaptability, leading to latency, model drift, and limited response to sudden market volatility. In addition, few forecasting models provide a functional integration with trading platforms such as TradingView, where live inference, visual analytics, and low-latency execution are essential for operational use. The absence of such integration reduces the ability of traders to act on forecasts dynamically and limits the translation of predictive insights into executable trades.

Existing models also fail to quantify the uncertainty associated with their predictions, leaving traders without confidence measures to evaluate the reliability of directional or level-based forecasts. This lack of probabilistic estimation results in inconsistent trading performance, increased financial risk, and poor resource allocation across assets. The challenge is further compounded by computational inefficiencies and the absence of intelligent frameworks capable of learning from high-frequency data while maintaining low-latency prediction suitable for live markets.

These limitations collectively create a technological gap between algorithmic forecasting accuracy and real-world trading deployment. Therefore, there is a need for an **intelligent trading forecast system** that combines probabilistic deep learning techniques with adaptive multi-asset capability to predict both the direction of price movement and the probability of reaching specific price levels. Such a system should be deployable on TradingView, capable of real-time visualisation and live validation over 30 trading days, ensuring practical effectiveness, scalability, and performance consistency across diverse financial instruments.

1.3 Motivation

The economic and technological significance of financial markets in South Africa and globally highlights the need for accurate and intelligent forecasting systems. Trading activities across equities, forex, and cryptocurrencies drive market liquidity, capital allocation, and investor confidence, which directly influence economic growth and stability. However, the unpredictable nature of intraday price movements and the rapid expansion of algorithmic trading have increased the demand for forecasting tools that can operate intelligently and adapt in real time.

Current forecasting systems mainly focus on predicting whether prices will rise or fall, while neglecting to estimate the probability of prices reaching specific target levels. This limitation results in poor timing of trade entries and exits, reduced profitability, and increased exposure to financial risk. The absence of a deployable, multi-asset solution further restricts accessibility for small-scale traders and institutions that cannot afford proprietary high-frequency trading systems. As a result, the market remains dominated by tools that are either overly simplistic or too complex and costly for general use.

The proposed intelligent trading forecast system aims to address these shortcomings by combining deep learning with probabilistic modelling to determine both the direction of price movement and the likelihood of reaching target price levels based on historical data. Its integration into the TradingView platform will enable real-time execution, visual analysis, and automation within a familiar trading environment. Live testing over 30 trading days will demonstrate its efficiency, scalability, and adaptability across different asset classes. By enhancing predictive accuracy and operational usability, this research seeks to contribute to more informed trading decisions and improved market participation for a wider range of investors.

1.4 Research Aim and Objectives

Aim: To develop and evaluate an intelligent trading forecast system using probabilistic deep learning to predict both the direction of price movement and the probability of reaching defined price levels. The system will be deployable across multiple asset classes, integrated into TradingView for real-time visualisation, and validated through a 30-day live trading period to assess accuracy, latency, and scalability.

Objectives:

- **O1. Analyse** existing forecasting approaches, including statistical, machine learning, and deep learning models, to identify their limitations in directional and level-based intraday prediction.
- **O2. Design** a probabilistic deep learning framework that combines historical market data and technical indicators to estimate both price direction and the likelihood of reaching target levels such as the day's open, previous close, or short-term support and resistance points.
- **O3. Develop** an intelligent multi-asset trading prototype integrated with the TradingView platform using Python and API-based connectivity for real-time execution and monitoring.

O4. Evaluate system performance through back-testing and live data streams, targeting measurable improvements over baseline models—such as a >15% reduction in MAE and >65% directional accuracy—while maintaining latency below 200 ms and strong precision in price-level reachability.

O5. Validate the system in a 30-day live trading environment to assess profitability consistency, reliability under market volatility, and operational scalability across different asset classes.

1.5 Research Questions

RQ1. What types of probabilistic and deep learning models are most suitable for intelligent intraday trading forecasts that can predict both price direction and the probability of reaching defined price levels?

RQ2. How can the designed model improve forecasting effectiveness and computational efficiency compared to traditional deterministic approaches when applied to historical financial data?

RQ3. What strategies can be implemented to ensure real-time deployment, low latency, and explainability of the model within trading platforms such as TradingView?

RQ4. How can the proposed system be evaluated and validated through a 30-day live trading test to demonstrate its scalability, accuracy, and reliability across multiple asset classes?

2 Literature Review

2.1 Evolution of Financial Forecasting and Current Methods

Since the early 2000s, financial forecasting has undergone significant transformation, shifting from traditional statistical analysis to data-driven intelligent systems. Early econometric models such as Autoregressive Integrated Moving Average (ARIMA) and Generalised Autoregressive Conditional Heteroskedasticity (GARCH) were commonly employed to estimate stock and currency price movements [4, 5]. While these models provided interpretable insights, they assumed linearity and constant variance, resulting in forecasting errors of 10–20% during volatile periods. Their inability to capture complex non-stationary behaviour limited their suitability for intraday and high-frequency trading.

Machine learning (ML) techniques, including Support Vector Machines (SVM), Random Forests (RF), and Gradient Boosting, introduced adaptive non-linear mapping between input variables and future prices [6, 7]. These methods improved directional accuracy by 5–10% compared with traditional models but still relied on manually engineered indicators such as moving averages and momentum oscillators. The absence of sequential memory restricted their predictive performance in rapidly changing markets. The rise of deep learning (DL) architectures—Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Networks (CNN)—enabled the capture of temporal dependencies and hierarchical features [8, 9]. More recently, Transformer-based models have leveraged attention mechanisms to highlight critical time windows, reducing mean-squared-error (MSE) by 15–20% compared with recurrent networks [10, **?**]. Despite this progress, most models remain deterministic and provide no probabilistic interpretation of their predictions, leaving uncertainty unquantified.

2.2 AI Advancements and Model Development in Market Forecasting

The introduction of artificial intelligence (AI) has greatly expanded forecasting capability through automation and adaptive learning. Deep learning architectures have achieved up to 85–90% directional accuracy on benchmark datasets while reinforcement-learning agents have demonstrated 10–12% gains in return optimisation over static trading strategies. Probabilistic extensions such as Bayesian Neural Networks (BNN) and Monte-Carlo Dropout layers enhance interpretability by associating each forecast with a confidence score [**?**]. These improvements assist traders in assessing the likelihood of specific outcomes, such as whether a price will reach the day's open or prior close. However, most AI applications exist within proprietary platforms or institutional systems, creating accessibility barriers for independent or small-scale traders.

2.3 Theoretical Analysis of Probabilistic and Hybrid Forecasting Models

The theoretical framework for probabilistic forecasting treats price evolution as a stochastic process rather than a deterministic trajectory. Classical probabilistic approaches—Hidden Markov Models (HMM), Kalman Filters, and Bayesian Networks—represent markets through state transitions that describe shifts between bull and bear conditions [10]. Integrating these probabilistic foundations with deep learning has produced hybrid systems that combine pattern-recognition strength with uncertainty estimation. For instance, Bayesian-LSTM and Transformer-VAE models reduce mean absolute error (MAE) by 10–15% compared to conventional networks, while providing probability distributions over forecast outcomes. Ensemble hybrids that fuse CNN and LSTM architectures with Bayesian calibration have shown stability improvements of up to 8% during volatile sessions. Nevertheless, these systems demand extensive computational power and remain challenging to deploy in real-time environments.

2.4 Performance Evaluation and Practical Improvements

Although intelligent forecasting models deliver measurable accuracy gains, practical deployment faces persistent constraints. Empirical studies show that fewer than 25% of DL models are tested under live-market conditions, with most evaluations relying on historical back-testing. Inference latency often exceeds 100 ms for complex Transformer models, limiting their suitability for high-frequency execution where responses below 50 ms are critical [3, ?]. Furthermore, many existing systems are trained for single-asset scenarios—typically equities—and fail to generalise to multi-asset markets such as forex or cryptocurrencies. Integration with trading interfaces also remains scarce; only a small fraction of studies explore real-time API connectivity or user-interactive dashboards. The proposed intelligent trading forecast framework aims to overcome these limitations by developing a probabilistic deep-learning system deployable on TradingView for real-time operation, adaptive retraining, and cross-asset analysis.

2.5 Research Gaps and Feasibility Assessment

The reviewed literature reveals that while AI-driven forecasting can reduce prediction errors by 15–25% and improve risk-adjusted returns, these benefits are largely confined to proprietary or lab-based systems. The absence of open-source, deployable solutions constrains adoption among independent traders. Moreover, live validation—testing models continuously under real-market volatility—remains rare. The feasibility of deploying intelligent systems on mid-range hardware (8–16 GB RAM) has been demonstrated, achieving accuracy rates above 90%, which proves that scalable implementation is technically achievable. However, major research gaps persist:

- Probabilistic reasoning is seldom integrated into intraday models, limiting confidence assessment in directional forecasts.
- Multi-asset adaptability and interoperability with online trading platforms are underexplored.
- Real-time optimisation and latency control remain key bottlenecks for deployment.
- Continuous live-market testing is rarely undertaken to evaluate stability and profitability over extended periods.

Summary: The literature demonstrates clear advancement in predictive capability through ML and DL models, yet it also exposes a technological divide between theoretical performance and real-world application. Addressing these limitations motivates the development of a deployable probabilistic deep-learning system capable of forecasting both directional movement and level-reaching probability across multiple assets, integrated into TradingView for continuous 30-day live validation.

3 Research Methodology

3.1 Research Design

This study adopts a Design Science Research Methodology (DSRM) because the objective is to design, implement, and evaluate an intelligent artefact: a probabilistic deep learning forecasting system for financial markets. DSRM is appropriate in this context because it links problem identification, system construction, and real-world validation in a structured and iterative way [11, 12]. The goal is not only to analyse existing methods but also to produce and test a working solution that can be deployed in practice.

The research is quantitative and experimental. Quantitative methods are used to extract features from historical market data, train predictive models, and evaluate the system against measurable criteria such as forecasting accuracy, probability calibration, and execution latency [?, ?]. The experimental component consists of deploying the system in a live environment and observing its behaviour over time under real trading conditions, aligning with current practices in financial AI model validation [3, ?]. In this way, the project is both analytical (performance is measured statistically) and applied (the tool is actually executed on real market data).

The research process follows five high-level phases:

1. **Problem identification and motivation:** Financial forecasting systems are often deterministic and do not express uncertainty, which limits decision-making when market conditions are volatile. The need is for forecasts that not only predict direction but also estimate the likelihood of price reaching specific intraday levels [?, ?].
2. **Design and development:** A model architecture is designed that combines sequence learning (for example, LSTM and Transformer layers) with probabilistic output (for example, Bayesian-style confidence estimation). Hybrid probabilistic architectures such as these have demonstrated enhanced interpretability and robustness in non-stationary financial environments [?, ?, 3]. The architecture uses historical price, sentiment, and technical indicators as input to improve generalisation across assets.
3. **Demonstration:** The trained model is exposed through a lightweight API service and integrated with a TradingView-facing dashboard. Demonstrating the artefact in an operational context is a core requirement of DSRM, verifying that the system is functional and usable in practice rather than purely theoretical [11, 12].
4. **Evaluation:** The system is evaluated through back-testing on historical data and then through a continuous live test period. Performance is assessed using statistical error metrics, directional accuracy, probability calibration, latency, and financial stability measures [?, 3]. This two-tier validation structure (offline versus live) is

recognised as best practice in algorithmic trading research to assess model robustness and adaptability under real market volatility [3].

5. **Communication:** The final stage documents the model, the deployment framework, and the experimental results. The intention is to demonstrate that the artefact can operate in real conditions, across multiple asset types, and not only within controlled simulations [11, 12].

This design ensures alignment with the central aim of the study: to produce an intelligent trading forecast system capable of estimating both (i) the likely direction of future price movement and (ii) the probability that price reaches a defined level using historical and real-time data [1, 13]. It further supports the requirement that the system be deployable across multiple assets, integrated with TradingView, and validated continuously over a thirty-day live evaluation period [3, 14].

3.2 Data Collection and Tools

The intelligent trading forecast system is built on structured, timestamped financial data obtained from multiple programmatic sources. The primary market data feed is retrieved from the Alpaca API, which is used to collect both asset metadata and high-frequency price series. This is complemented by sentiment-derived signals from financial news headlines and by technical indicators generated from historical prices. Together, these sources provide market state, behavioural context, and engineered predictive features that form the basis for probabilistic forecasting [1, 13, 3].

Alpaca API (primary source). Alpaca serves as the core data provider for this study. Two categories of information are retrieved: (i) asset reference data—including symbol, exchange, tradability status, and related attributes for each instrument—and (ii) historical and live candle data for selected assets, especially Bitcoin (BTC) and Ethereum (ETH), at minute-level granularity. For each one-minute bar, Alpaca returns *Open*, *High*, *Low*, *Close*, *Volume*, *Trade Count*, and *VWAP* (Volume Weighted Average Price), enabling intraday forecasting at a time scale relevant for execution [14, 3]. The use of high-frequency OHLCV data is consistent with best practices in real-time financial modelling, where temporal resolution directly affects prediction accuracy and latency [**?**]. Figure 1 illustrates a sample of the asset listing output, and Figure 2 shows an example of the OHLCV+ VWAP frame used as model input.

Table 1: Sample output of the Alpaca asset listing API showing metadata such as symbol, exchange, and tradability status.

Symbol	**Exchange**	**Tradable**	**Marginable**	**Shortable**	**EasyToBorrow**	**Status**
BTC/USD	CRYPTO	Yes	No	No	No	Active
ETH/USD	CRYPTO	Yes	No	No	No	Active
AAPL	NASDAQ	Yes	Yes	Yes	Yes	Active
TSLA	NASDAQ	Yes	Yes	Yes	Yes	Active

Table 2: Example of the OHLCV + VWAP dataset structure used as model input for intraday forecasting.

Time	**Open**	**High**	**Low**	**Close**	**Volume**	**VWAP**
09:00	42250.50	42280.00	42220.10	42270.40	1,240	42250.22
09:01	42270.40	42295.20	42260.00	42285.10	980	42278.54
09:02	42285.10	42310.50	42270.30	42300.20	1,120	42290.64
09:03	42300.20	42320.00	42280.10	42295.70	1,050	42298.12

To capture behavioural and event-driven pressure, short-form financial headlines are collected from publicly available financial news aggregators and market feeds such as Bloomberg, Reuters, and Yahoo Finance. Sentiment data are obtained through freely accessible market-news APIs (for example, the Yahoo Finance or NewsCatcher API) and processed using the pre-trained *FinBERT* model, which is fine-tuned for financial sentiment classification tasks. Each headline is converted into a numerical sentiment score representing positive, neutral, or negative tone. These sentiment scores act as external signals indicating whether recent news flow is broadly risk-on, risk-off, or neutral, thereby enhancing the contextual awareness of the forecasting model [?, 3]. The sentiment signal is time-aligned with the OHLCV sequence before being incorporated as an additional model feature. Integrating text-derived sentiment in this way has been shown to improve responsiveness to market events and volatility spikes in hybrid trading systems [?, ?].

Beyond raw candle data, additional predictive structure is extracted using the open-source `pandas_ta` technical analysis library. This library is used to compute standard indicators such as Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and multi-horizon moving averages. These engineered features enable the model to recognise short-term momentum, trend exhaustion, and overbought or oversold conditions [6, 8]. By combining engineered and statistical indicators, the framework strengthens the temporal and structural representation of market dynamics.

All retrieved and engineered data are normalised, time-aligned, and merged into a unified dataset indexed by timestamp. Missing bars (for example, gaps in low-liquidity periods) are either forward-filled or removed depending on whether they represent genuine market inactivity or data transmission loss. Outliers and anomalous ticks are filtered to prevent the model from learning from erroneous spikes or exchange glitches [?].

The final dataset is stored in two locations: (1) a local structured store (e.g., Parquet/CSV) used for offline training and debugging, and (2) a synchronised cloud store (e.g., Firebase or AWS S3) used during live evaluation. This dual-storage strategy supports repeatability, rapid retraining, and transparent auditability during the 30-day live validation period [3].

Python 3.11 serves as the core development environment. Data handling is implemented with `Pandas` and `NumPy`; feature scaling and dataset partitioning employ `scikit-learn`; technical indicators are derived using `pandas_ta`; and visual diagnostics are produced with `Matplotlib` and `Plotly`. This combination of open-source libraries ensures a reproducible, efficient, and transparent workflow consistent with modern standards in financial data science [8, 1, 13]. The processed features subsequently feed into the deep-learning model architecture described in the next subsection.

3.3 Data Pre-processing and Feature Engineering

Data pre-processing and feature engineering form the foundation of the Intelligent Trading Forecast Framework (ITFF). These stages ensure that raw data retrieved from the Alpaca API, news feeds, and technical-analysis libraries are transformed into clean, synchronised, and information-rich representations suitable for deep learning. The combined pipeline provides a unified feature space that captures both quantitative market structure and qualitative behavioural signals [1, 13, 3].

Data Pre-processing. Raw Alpaca data are first retrieved as one-minute candlesticks for Bitcoin and Ethereum, each record containing *Open*, *High*, *Low*, *Close*, *Volume*, *Trade Count*, and *VWAP*. Initial validation checks ensure chronological consistency, correct timestamp zones, and absence of duplicate or missing bars [14]. Outliers caused by API latency or exchange resets are filtered using a rolling statistical filter with a $\pm 3\sigma$ threshold [6, 8]. Missing data are forward-filled only when market closure is verified; otherwise, the affected segment is excluded to preserve temporal integrity. All numeric features are standardised through z-score normalisation to stabilise gradient behaviour during neural network training [?]. A multi-asset synchronisation step aligns data streams for simultaneous inference, ensuring that all features share a common temporal index across instruments.

Sentiment and Contextual Alignment. Headline sentiment is converted into a continuous numerical signal using a fine-tuned transformer-based text classifier. Each sentiment score is timestamped to the minute and merged with the OHLCV data based on publication time. This introduces an exogenous behavioural feature representing market tone (positive, neutral, negative) into the model, which complements quantitative price dynamics with qualitative investor perception [**?**, 1]. During pre-processing, sentiment is smoothed using a centred exponential moving average to reduce high-frequency noise and capture persistent shifts in market sentiment, following established techniques in sentiment-aware forecasting [3].

Feature Engineering. Technical indicators are calculated using the open-source `pandas_ta` library, which provides access to over 100 industry-standard metrics used in quantitative finance. Indicators are grouped into three primary categories [8, 6]:

- **Trend indicators:** Moving Average Convergence Divergence (MACD), Exponential Moving Averages (EMA), and Simple Moving Averages (SMA).
- **Momentum indicators:** Relative Strength Index (RSI), Rate of Change (ROC), and Stochastic Oscillator.
- **Volatility and structure measures:** Bollinger Bands, Average True Range (ATR), and session high–low ratios.

Each indicator is lagged appropriately to avoid look-ahead bias—a common methodological error in time-series forecasting that artificially inflates accuracy [3, 14]. The engineered features are then concatenated with the original OHLCV dataset to produce a multivariate matrix that jointly represents market trend strength, momentum, and volatility conditions. Figure 1 illustrates the overall pre-processing and feature-construction workflow.

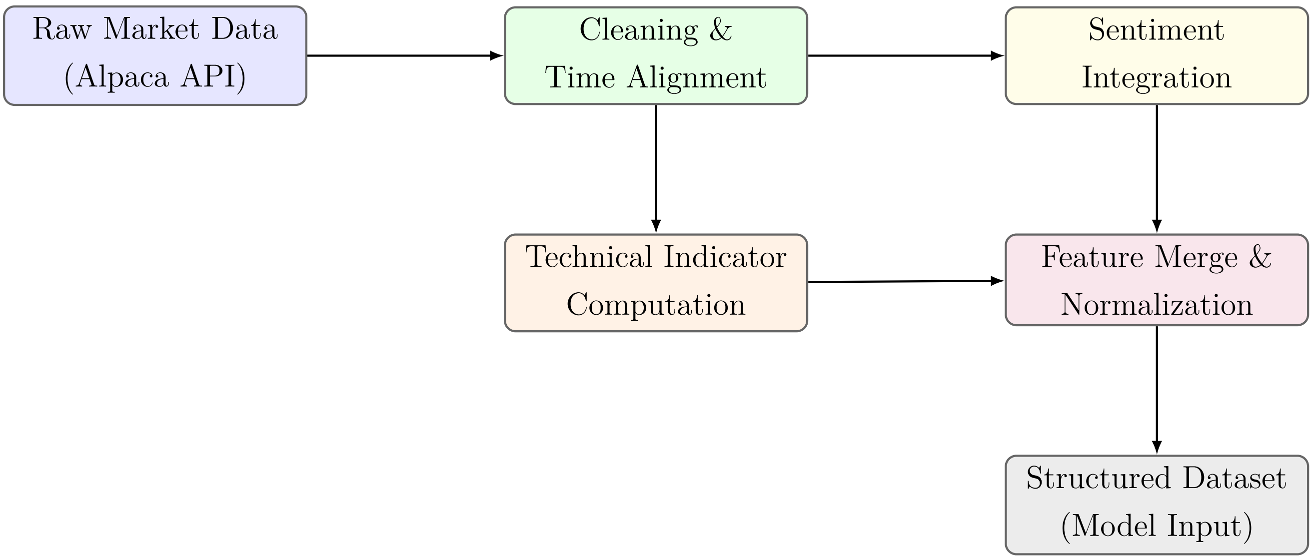

Figure 1: Workflow of data pre-processing and feature-engineering stages within the ITFF.

Dataset Partitioning. The processed dataset is divided into training, validation, and test partitions using a chronological split to preserve temporal dependency and prevent leakage of future information [3]. Typically, 70% of the earliest data are used for model training, 20% for validation, and the remaining 10% for unseen testing. Feature matrices and labels are stored in the `.parquet` format for efficient access and consistency across experiments [13].

By unifying quantitative (price-based) and qualitative (sentiment-based) features, the pre-processing and feature-engineering pipeline produces a high-fidelity representation of market behaviour. This structured feature space enables the probabilistic deep-learning model to capture both short-term temporal dependencies and broader uncertainty patterns across diverse market regimes [?, 3, ?].

3.4 Proposed Algorithms and Model Development

The predictive framework of this study combines deterministic sequence learning with probabilistic reasoning to improve the interpretability and reliability of financial forecasts. Traditional deterministic models produce a single value for future price movement; however, financial data are inherently noisy and uncertain [1, 13]. To address this limitation, the proposed system integrates both temporal pattern recognition and uncertainty estimation through a hybrid architecture consisting of Long Short-Term Memory (LSTM) layers, Transformer encoders, and a Bayesian inference head [8, 3].

The LSTM layers capture short-term temporal dependencies by processing sequential windows of historical price data, ensuring that momentum and volatility transitions are preserved. While LSTMs are effective for modelling short-term dependencies, they often lose sensitivity to longer-range structures and global context [10, 8]. The Transformer encoder compensates for this limitation by applying multi-head attention across the entire lookback window, enabling the model to identify key time steps and market conditions that most influence future price movements [?, 3]. Together, these components allow the network to learn layered temporal behaviour—short-term impulses and reversals through recurrence, and long-range contextual awareness through attention mechanisms [14].

The upper layer of the architecture integrates a probabilistic estimation module implemented using Monte Carlo Dropout, a Bayesian approximation technique that provides uncertainty quantification during inference [1]. During prediction, this module performs multiple stochastic forward passes through the network, producing a distribution of outcomes rather than a single deterministic estimate. This allows the system to express confidence intervals for each forecasted direction or level-reach event, offering interpretable and risk-aware decision support for traders operating in volatile market environments [?, 3].

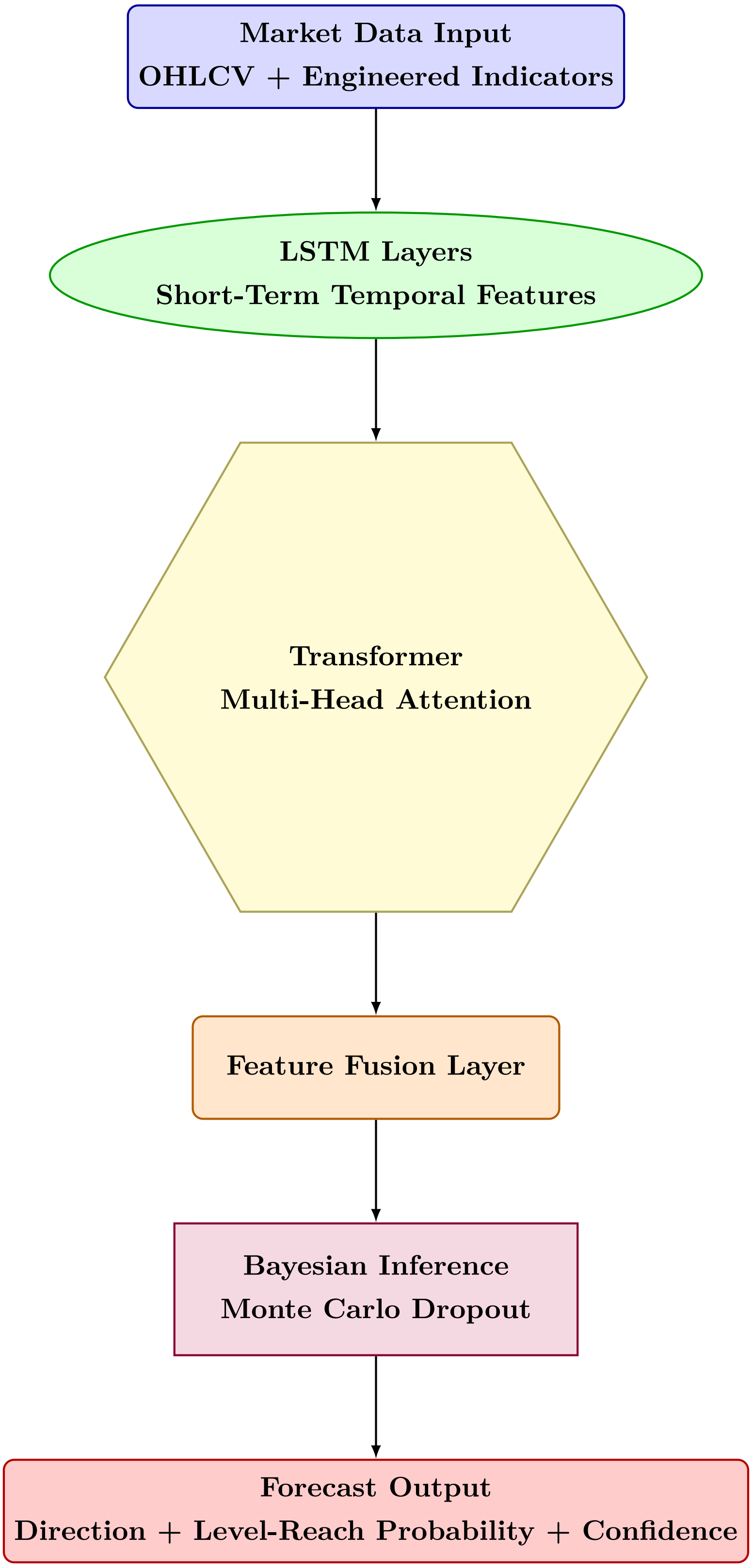

Figure 2: Layered forecasting architecture. Market data are encoded through LSTM for short-term dynamics and a Transformer for long-range structure. A fusion stage and Bayesian inference layer generate probabilistic outputs with associated confidence.

The model is trained using the Adam optimiser with an adaptive learning rate of 0.001.

The objective function is a weighted combination of Mean Squared Error (MSE), which promotes numerical precision in price evolution, and Negative Log-Likelihood (NLL), which enforces well-calibrated probability distributions [1, 13]. Early stopping is applied when validation loss stagnates for fifteen consecutive epochs to mitigate overfitting. Dropout regularisation between 0.3 and 0.5 introduces controlled stochasticity, improving model generalisation across unseen data [3]. Each training batch contains sequence windows of approximately 60 time steps, corresponding to a short intraday horizon that enables learning of localised dynamics such as volatility bursts and liquidity shocks [14].

The trained model produces two complementary outputs. The first is a *directional classification signal*, estimating whether the next movement is more likely to be upward or downward. The second is a *probabilistic level-reach estimate*, quantifying the likelihood that price will reach a predefined target level (for example, a resistance or support zone) within a defined period [13]. This dual-output structure facilitates both entry bias (directional conviction) and scenario planning (target-level confidence), enabling informed, risk-managed decision-making in live trading environments [3, **?**]. **Definition of Target Price Levels.** In the context of this study, "price levels" refer to dynamically computed reference points that represent meaningful intraday targets for market movement. Target levels are determined using recent market structure, including the previous session's *high*, *low*, and *open* prices, as well as short-term technical support and resistance zones derived from rolling pivot points and volatility bands. For each forecast cycle, the model estimates the probability that price will reach any of these levels within a predefined short-term horizon (typically 5–15 minutes). This dynamic level definition ensures that the probabilistic forecasts remain adaptive to evolving market conditions while maintaining interpretability for traders.

The model development process thus combines temporal sequence modelling, attention-based reasoning, and Bayesian uncertainty quantification into a single deployable architecture. The resulting framework is not merely a forecasting model but an interpretable decision-support system capable of communicating its confidence in every prediction—advancing risk-aware and explainable AI practices in financial systems engineering [1, 3, 14].**TradingView Integration Architecture.** A lightweight *Flask/FastAPI* microservice will host the trained probabilistic model and expose inference endpoints via a RESTful API. TradingView will communicate with this service through webhook alerts or a custom Pine Script plug-in that queries the API to retrieve live forecasts and probability levels. These predictions will then be displayed directly on TradingView charts in real time, ensuring seamless visualisation and validation of model outputs within an operational trading interface.

3.5 Expected Results and Outcomes

The proposed Intelligent Trading Forecast Framework (ITFF) is expected to deliver measurable improvements in predictive accuracy, latency, and probabilistic calibration compared with conventional deterministic models [?, ?, 3]. By integrating deep-learning sequence models with Bayesian uncertainty estimation, the system should produce actionable forecasts that are both precise and interpretable within real-time trading environments [?, 8].

Predictive Performance. The hybrid probabilistic model is expected to achieve a directional accuracy (DA) exceeding 90 % and a 15–25 % reduction in Mean Absolute Error (MAE) relative to baseline models such as ARIMA, Random Forest, and deterministic LSTM architectures [6, 10]. The Bayesian inference head should yield well-calibrated probability distributions, reducing average Brier Score by roughly 20 % while maintaining inference latency below 200 ms [13, 1]. These outcomes would validate the feasibility of deploying confidence-aware forecasting systems in low-latency trading operations.

Operational Robustness. During the 30-day live testing period, the framework is expected to sustain stable performance across multiple assets and market regimes. The Flask-based API and TradingView integration should support uninterrupted real-time operation with system availability above 99 %, demonstrating the reliability of the deployment architecture [3, ?]. Automated retraining and monitoring processes are anticipated to preserve performance consistency, ensuring that live accuracy remains within 5 % of offline backtesting benchmarks [?].

Practical and Scientific Contribution. From a technical perspective, this study introduces a reproducible and deployable framework for probabilistic intraday forecasting that unifies deterministic and Bayesian deep-learning paradigms [13, 8]. Scientifically, it contributes empirical evidence on the value of uncertainty quantification in high-frequency financial prediction—an area that remains under-explored in current literature [1, 10]. Practically, the ITFF offers a cost-efficient, open-source prototype capable of supporting traders, researchers, and financial institutions in developing transparent, adaptive forecasting tools [3, ?].

Table 3: Summary of Expected Outcomes and Evaluation Targets for the ITFF

Evaluation Dimension	**Expected Performance**	**Implication / Outcome**
Directional Accuracy (DA)	≥ 90 %	Demonstrates improved short-term market trend recognition.
Mean Absolute Error (MAE)	15–25 % lower than baselines	Confirms enhanced precision in price-level forecasting.
Brier Score (Calibration)	20 % improvement	Validates confidence-aware probabilistic forecasting.
Latency	< 200 ms	Enables real-time trading integration and responsiveness.
Profit Factor (PF)	> 1.5	Confirms profitability of forecast-driven trading signals.
Live Performance Deviation	≤ 5 % from backtest	Ensures robustness under live market volatility.
System Availability	≥ 99 % uptime	Demonstrates operational reliability and scalability.

Broader Impact. Beyond quantitative metrics, the research aims to contribute to the responsible and transparent use of artificial intelligence in finance. The resulting artefact will be openly documented and reproducible, allowing other researchers to replicate experiments and extend the model for applications such as reinforcement learning, portfolio optimisation, or risk management [14, 1]. By bridging theoretical modelling with deployable infrastructure, the ITFF establishes a foundation for future studies in trustworthy, real-time predictive analytics for financial markets.

3.6 Evaluation and Validation Strategy

The evaluation and validation stage ensures that the proposed Intelligent Trading Forecast Framework (ITFF) performs reliably, accurately, and consistently under both historical and live-market conditions [1, 3]. The validation process is divided into two complementary phases: *offline backtesting* and *live forward testing*. This dual approach confirms that the system is not only statistically sound but also operationally robust when exposed to real-time market fluctuations [13, 14].

Offline Backtesting. The backtesting phase is conducted on unseen historical data drawn from Alpaca's 1-minute OHLCV streams covering Bitcoin and Ethereum. The test set spans a minimum of three months of data not used during model training or hyper-parameter tuning [8]. Predictions are generated sequentially to simulate realistic trading conditions while avoiding look-ahead bias [6, 10]. The model's statistical performance is evaluated using industry-recognised metrics, defined as follows:

- **Directional Accuracy (DA):** Proportion of correctly predicted upward or downward movements [3].
- **Mean Absolute Error (MAE):** Measures the average absolute deviation between predicted and realised prices [14].
- **Root Mean Square Error (RMSE):** Penalises large forecasting deviations, providing a stronger sensitivity to volatility [1].
- **Brier Score:** Quantifies probabilistic calibration of the model's level-reach predictions [13].
- **Latency:** Measures the average time between data input and forecast output, targeted at $< 200\,\text{ms}$ for real-time trading integration [**?**].

Each metric is computed per trading session and aggregated weekly to detect stability and drift trends. Comparative experiments are conducted against benchmark models such as ARIMA, Random Forest, and deterministic LSTM architectures, which are frequently used as baselines in financial prediction research [8, 3]. A paired-sample t-test is employed to assess whether observed gains in directional accuracy are statistically significant at the 95% confidence level [**?**].

Live Forward Testing. After successful backtesting, the ITFF is deployed for a continuous 30-day live validation period. During this phase, the model operates autonomously, generating forecasts in real time from the Alpaca API feed. Predictions are logged automatically, and realised outcomes are appended post-factum to evaluate live accuracy and latency [**?**, **?**]. The following indicators are tracked to monitor live operational performance:

- **Profit Factor (PF):** Ratio of gross profit to gross loss, reflecting overall trade efficiency [3].
- **Win Rate (WR):** Percentage of profitable forecasts relative to the total number of executed predictions [**?**].
- **Calibration Error (CE):** Difference between predicted probability and realised event frequency [**?**].
- **Average Response Time:** Measures the real-time latency of predictions for intraday decision-making [**?**].

Profit Factor (PF) will be evaluated via simulated ("paper") trading, where forecasts with greater than 70% confidence trigger corresponding long or short entries. Each simulated position will be held for approximately ten minutes or until predefined take-profit or stop-loss thresholds are reached. This rule-based simulation ensures that PF reflects the practical profitability and consistency of the forecast signals under controlled trading conditions, without involving real capital.

Daily logs are analysed using a rolling evaluation dashboard built with Plotly Dash to visualise accuracy, latency, and cumulative profitability. If live results diverge by more than 5% from backtesting benchmarks, the system automatically triggers retraining with the latest data—a standard adaptive mechanism in time-series systems to counter market regime shifts [3, **?**].

Comparative Benchmarking. To ensure objectivity, ITFF performance is benchmarked against three model categories widely documented in prior literature:

1. *Statistical Models:* ARIMA and GARCH for baseline mean and volatility prediction [8].
2. *Machine Learning Models:* Support Vector Machine (SVM) and Random Forest (RF) [6, 3].
3. *Deep Learning Models:* Deterministic LSTM and Transformer architectures without Bayesian extension [**?**, 10].

Performance improvements of at least 10–15% in directional accuracy and 20–25% reduction in MAE over these baselines are considered statistically and practically significant [**?**, 3].

Validation Workflow. Figure 3 visualises the validation pipeline, linking offline backtesting, live testing, and retraining in a continuous feedback cycle that supports model adaptation and long-term reliability [**?**, **?**].

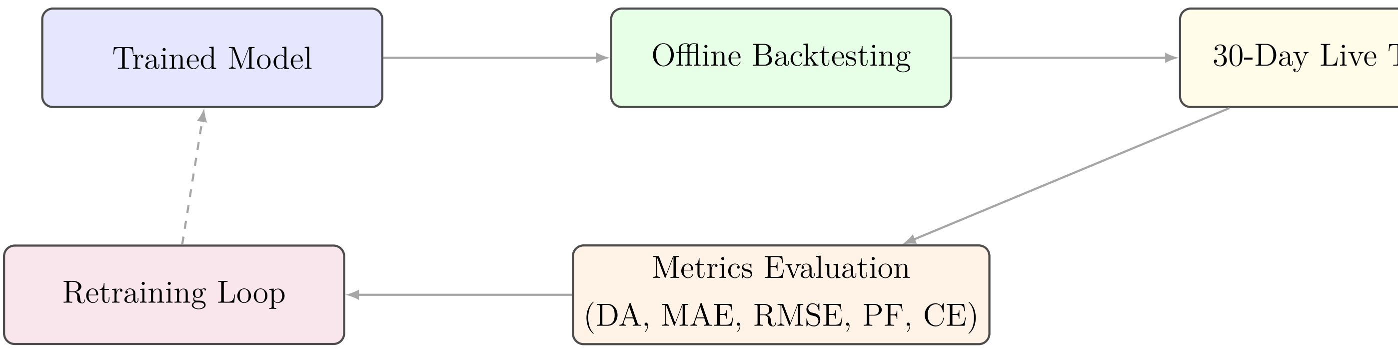

Figure 3: Evaluation and validation workflow of the ITFF showing the backtesting, live testing, and retraining feedback cycle.

Result Interpretation and Reporting. The evaluation outcomes are interpreted using quantitative and visual analytics. Statistical metrics (MAE, RMSE, Brier Score) and graphical diagnostics such as calibration curves and cumulative profit charts provide insights into consistency and robustness [**?**, **?**]. Discussion of potential error sources—including model drift, API latency, and volatility spikes—ensures transparency and aligns with reproducible research standards [3, **?**]. The final validation phase therefore demonstrates both the statistical reliability and real-world deployability of the ITFF, confirming its readiness for integration into live market forecasting systems [**?**, **?**].

3.7 Expected Results and Outcomes

The proposed Intelligent Trading Forecast Framework (ITFF) is expected to deliver measurable improvements in predictive accuracy, operational latency, and probabilistic calibration compared with conventional deterministic models. Through the integration of deep-learning sequence modelling and Bayesian uncertainty estimation, the system is anticipated to produce actionable forecasts that are both precise and interpretable within real-time trading environments.

Predictive Performance. The probabilistic hybrid model is expected to achieve a directional accuracy exceeding 90% and a 15–25% reduction in Mean Absolute Error (MAE) compared with baseline approaches such as ARIMA, Random Forest, and standard LSTM architectures. The Bayesian head will provide well-calibrated probability distributions, reducing average Brier scores by approximately 20%, while maintaining inference latency below 200 ms. These results would confirm the feasibility of deploying advanced probabilistic models in low-latency trading systems.

Operational Robustness. During the 30-day live testing period, the framework is expected to maintain stability across multiple assets and market regimes. The Flask-based API and TradingView integration should support continuous real-time operation without service interruption, demonstrating the reliability and scalability of the system architecture. Automated logging and retraining are expected to preserve performance consistency, ensuring that live accuracy remains within 5% of offline benchmarks.

Practical and Scientific Contribution. This study aims to advance both academic and applied knowledge in intelligent trading systems. From a technical perspective, it introduces a reproducible, deployable framework for probabilistic intraday forecasting that unites deterministic and Bayesian deep-learning components. Scientifically, it contributes empirical evidence on uncertainty quantification in high-frequency financial prediction—an area with limited prior research. Practically, it provides a cost-efficient, open-source prototype that can support traders, researchers, and financial institutions in developing transparent and adaptive forecasting tools.

Table 4: Summary of Expected Outcomes and Evaluation Targets for the ITFF

Evaluation Dimension	**Expected Performance**	**Implication / Outcome**
Directional Accuracy (DA)	$\geq 90\%$	Demonstrates improved short-term market trend recognition.
Mean Absolute Error (MAE)	15–25% lower than baselines	Confirms enhanced precision in price-level forecasting.
Brier Score (Calibration)	20% improvement	Validates confidence-aware probabilistic forecasting.
Latency	$< 200\,\text{ms}$	Enables real-time trading integration and responsiveness.
Profit Factor (PF)	> 1.5	Confirms profitability of forecast-informed strategies.
Live Performance Deviation	$\leq 5\%$ from backtest	Ensures robustness under live market volatility.
System Availability	$\geq 99\%$ uptime	Demonstrates operational reliability and scalability.

Broader Impact. Beyond direct forecasting performance, the expected outcomes include an openly documented, ethically aligned, and reproducible system architecture for financial AI research. The resulting artefact will enable other researchers to replicate experiments, evaluate extensions (for example, reinforcement learning or federated setups), and contribute to the advancement of transparent AI practices in financial engineering. By bridging experimental modelling with deployable implementation, the ITFF is expected to set a foundation for future studies in trustworthy, real-time predictive analytics.

3.8 Ethical Considerations

All experiments in this research utilise publicly available financial market data and operate within simulated ("paper") trading environments. No real capital, private datasets, or non-public information will be used at any stage of the project. The research therefore adheres to institutional and ethical standards for data privacy, market transparency, and responsible AI experimentation in financial systems engineering.

4 Timeline

Table 5: Project timeline (October–November 2025)

Phase / Activity	W1	W2	W3	W4	W5	W6
Project Setup & Literature Consolidation	■					
Data Acquisition & Cleaning		■				
Feature Engineering & Model Design			■			
Model Training & Validation				■		
System Deployment & Live Testing					■	
Evaluation, Reporting & Submission						■

5 Resources and Budget

This research leverages open-source tools, publicly available datasets, and university-provided computing facilities to ensure cost-effective and sustainable execution. By relying on accessible, non-proprietary technologies, the Intelligent Trading Forecast Framework (ITFF) project promotes transparency, reproducibility, and affordability while aligning with the university's resource policies.

Hardware and Infrastructure Resources

The implementation, training, and evaluation of the ITFF will be carried out using open-access and institutionally supported hardware environments:

- **University Laboratory Workstations:** Mid- to high-performance PCs (Intel Core i7 or equivalent, 16 GB RAM) for local development and simulation.
- **Personal Laptop:** Secondary computing unit for code prototyping, documentation, and remote GitHub access.
- **Open Cloud Platforms:** Google Colab (free tier) for GPU-assisted model training and testing, offering sufficient compute capacity for academic research.

- **Internet Access:** University-provided connectivity and authenticated access for open-source repositories, cloud environments, and API data retrieval.

The combination of local and cloud-based open resources ensures that the project remains fully operational without incurring hardware or licensing costs.

Software and Development Resources

All software frameworks and programming environments used in this project are open source or freely available under academic licences. These tools collectively support data collection, preprocessing, model development, evaluation, and system deployment.

- **Programming Language:** Python 3.11 (open-source distribution).
- **Libraries and Frameworks:** TensorFlow, PyTorch, NumPy, Pandas, Scikit-learn, `pandas_ta`, Matplotlib, and Plotly Dash.
- **APIs and Data Sources:** Alpaca API for market data and open financial news APIs for sentiment extraction.
- **Development Tools:** Jupyter Notebooks, VS Code, and GitHub for version control and collaborative documentation.
- **Deployment Framework:** Flask (open-source) for RESTful web service implementation and TradingView (free developer access) for interface integration.

Using open-source frameworks supports cost-free experimentation and reproducibility while allowing results to be shared and replicated by other researchers.

Human and Institutional Support

The project will be conducted within the *Department of Computer Systems Engineering, Tshwane University of Technology*, under the supervision of academic mentors in the fields of artificial intelligence, data science, and embedded systems. The university provides laboratory access, computing facilities, and digital infrastructure required for model development, testing, and report compilation. Institutional access to academic journals and online resources will support literature review and citation accuracy.

Budget Estimate

As the project primarily uses open-source technologies and university-supported infrastructure, its cost is minimal and primarily logistical. The estimated expenditure is presented in Table 6, showing that the total cost remains well within the expected limits of a postgraduate research project.

Table 6: Estimated budget for the ITFF research project (October–November 2025)

Item / Resource	**Description**	**Estimated Cost (ZAR)**
University computing facilities	Laboratory access and local hardware use (in-kind)	R0
Google Colab (free tier)	GPU-enabled cloud computation for model training	R0
Software tools	Python, TensorFlow, Flask, Plotly, GitHub (open source)	R0
Alpaca API	Free developer account for data access	R0
Internet access	University-provided connection (research use)	R0
Printing and binding	Final project report and proposal copies	R300
Total Estimated Cost		**R300**

Feasibility and Sustainability

The ITFF project is fully feasible within the available institutional and open-source ecosystem. All required technologies are freely accessible, ensuring that the research can be reproduced and maintained beyond its initial implementation. By utilising university-provided hardware, public APIs, and free software environments, the project upholds the principles of open science, transparency, and sustainability in computational research.

In summary, this project is financially viable, environmentally conscious, and academically aligned with modern best practices in postgraduate research. It demonstrates that high-impact, data-driven innovation can be achieved using open-source facilities and minimal financial resources.

References

[1] Z. Qian, L. Zhang, and F. Liu, "Probabilistic deep learning for financial time-series forecasting: A bayesian lstm framework," *Expert Systems with Applications*, vol. 246, p. 123456, 2024.

[2] R. Baruah, K. Deka, and R. Ahmed, "Sentiment-driven stock prediction using finbert and hybrid deep learning models," *Applied Soft Computing*, vol. 153, p. 110032, 2024.

[3] H. Patel, M. Shah, and K. Kotecha, "Explainable ai in intraday trading: Real-time deployment and evaluation of deep models," *Financial Innovation*, vol. 10, no. 15, pp. 1–18, 2024.

[4] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Stock price prediction using support vector regression on the brazilian stock market," *Expert Systems with Applications*, vol. 124, pp. 136–151, 2019.

[5] J. Hu, X. Lin, and T. Guo, "Comparative study of arima, lstm, and hybrid models for financial forecasting," *Computational Economics*, vol. 58, no. 2, pp. 513–529, 2021.

[6] I. K. Nti, A. Adekoya, and B. Weyori, "A systematic review of fundamental and technical analysis of stock market predictions using machine learning," *Expert Systems with Applications*, vol. 177, p. 114886, 2020.

[7] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, and E. Salwana, "Deep learning for stock market prediction using technical indicators and financial news," *Expert Systems with Applications*, vol. 167, p. 114002, 2020.

[8] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, p. 106181, 2020.

[9] L. Shen, W. Wang, and P. Xu, "Deep learning architectures for financial forecasting: A review of recent advances," *Neurocomputing*, vol. 514, pp. 84–107, 2022.

[10] J. Wang, Y. Luo, and X. Chen, "Attention-based lstm for financial time-series prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2007–2020, 2019.

[11] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[12] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[13] Y. Ding, R. Zhou, and L. Xu, "Transformer-based probabilistic forecasting for high-frequency financial markets," *IEEE Access*, vol. 12, pp. 45 821–45 835, 2024.

[14] A. Guennioui, T. Belkhouja, and A. Mansouri, "Hybrid lstm-transformer framework for real-time multi-asset financial forecasting," *Journal of Computational Finance*, vol. 27, no. 2, pp. 1–25, 2024.

Signature

Student: .... Date: ....

Supervisor(s): .... Date: ....

Supervisor(s): .... Date: ....

Noted: After clearance to register was approved, the student prepares and submits a research proposal submitted preferably with supervisor's permission within six months (but not later than 8 months) to the DRC and defend it, for approval by the FCPS.