Informatics Institute of Technology

In Collaboration With

University of Westminster, UK

The University of Westminster, Coat of Arms

A Novel Approach to Time Series Forecasting using Liquid Time-constant Networks

A Project Proposal by

Mr Ammar Raneez

W1761196 | 2019163

Supervised by

Mr Torin Wirasingha

September 2022

This Project Proposal is submitted in partial fulfilment of the requirements for the BSc (Hons) Computer Science degree at the University of Westminster.

Contents

List of Tables

List of Figures

Acronyms

AI	Artificial Intelligence.	**ML**	Machine Learning.
BPTT	Back-Propagation Through Time.	**(s)MAPE**	Symmetric Mean Absolute Product Error.
BTC	Bitcoin.	**MASE**	Mean Absolute Scaled Error.
CTN-GRU	Continuous-time Gated Recurrent Unit.	**MSE**	Mean Squared Error.
CT-RNN	Continuous-time Recurrent Neural Network.	**NLP**	Natural Language Processing.
DL	Deep Learning.	**ODE**	Ordinary Differential Equations.
GUI	Graphical User Interface.	**RMSE**	Root Mean Squared Error.
LSTM	Long Short-Term Memory.	**RNN**	Recurrent Neural Network.
LTC	Liquid Time-constant.	**TS**	Time Series.

1. INTRODUCTION

In this document, the author aims to identify and provide the reader with an overview of the current issues in time series forecasting and highlight what a liquid time-constant network is and what it aims to solve. To elaborate, the problem will be defined and the necessary literature will be evaluated to come up with a justifiable research gap and respective research challenges. The proposed methodology and deliverables are also justified.

2. PROBLEM DOMAIN

2.1 Time Series Forecasting

TS forecasting is a significant business issue and an area where ML could create a promising impact. It serves as the foundation for contemporary business practices, including pivotal domains like customer management and inventory control, marketing and finance. As a result, it has a comprehensive financial impact, with millions of dollars for each additional point of forecasting accuracy (Jain, 2017).

Having said that, although ML and DL have outperformed classical approaches for NLP and computer vision, the domain of TS still seems to be a point of struggle when compared to classical statistical methodologies (Makridakis et al., 2018a;b). For instance, out of a total of 60 submissions, the six "pure" ML methods submitted to the M4 competition were ranked 23, 37, 38, 48, 54, and 57, and the majority of the top-ranking methods were ensembles of traditional statistical techniques (Makridakis et al., 2018b).

It is therefore worth mentioning that the winner of this competition was a hybrid model of an LSTM (Smyl, 2020), which went on to claim that the only way forward for improving TS forecasting accuracy was by creating hybrid models, which the author aspires to challenge on this research project.

2.2 Liquid Time-Constant Networks

LTCs are neural ODEs: hidden layers aren’t specified, instead, a neural network is used to parameterize the derivative of the hidden state (Chen et al., 2018). RNNs with continuous time hidden states determined by ODEs are effective algorithms for TS data modelling (Chen et al., 2018). Studies show that existing algorithms such as the CT-RNN (Funahashi and Nakamura 1993; Rubanova, Chen, and Duvenaud 2019) and CT-GRU (Mozer, Kazakov, and Lindsey 2017) produce such performance, however, have issues in expressivity and a fixed behaviour once trained (Hasani et al., 2020). Therefore, the question arises, what would happen if there were unexpected changes to the characteristic of the inputs during inference? Additionally, these algorithms lose in generalization in comparison to even a simple LSTM network (Hasani et al., 2021), which arises another question on what is the point of defining a different and *"fancy"* approach if they cannot work in real world applications well?

Hasani et al., states that LTCs can *"identify specialized dynamical systems for input features arriving at each time point"* (2020, p1). The ability to exhibit stable and bounded behaviour demonstrates that the proposed approach yields more predominant expressivity (Hasani et al., 2020).

LTCs state and their respective time-constant exhibit bounded dynamics and assure the stability of the output dynamics, which is a prominent factor when inputs increase relentlessly (Hasani et al., 2020).

2.3 Cryptocurrencies

The word "Crypto" is a huge buzzword in recent times, BTC, especially. It has even come to the point where crypto and BTC are used interchangeably.

Cryptocurrencies are a form of digital currency that is fully decentralized (Rahouti et al., 2018); it's a form of a peer-to-peer system without the need for a third party, thereby enabling safer online transactions (S. Nakamoto, 2008). In the world of digital currencies, BTC is the first and the most popular to date, which has piqued the interest of many academic researchers (Rahouti et al., 2018).

However, being at the forefront of the digital currency world, it has developed high volatility, making it difficult to predict future rates and a disadvantage for multiple investors (Kervanci and Akay, 2020). Despite that, recent advances in ML and statistics have shown somewhat okay results in the analysis and prediction of cryptocurrencies, yet, the root cause of these algorithms persists: they are static.

3. PROBLEM DEFINITION

As of writing this report, there is no application of liquid time-constant networks in any domain since this novel neural ODE has only recently been announced. Existing intelligent systems utilize more traditional approaches of neural nets developed some time ago.

Having mentioned that, most applications of ML available do perform quite well (Ex: image classification, transfer learning, NLP etc.), yet, as mentioned, the field of TS forecasting seems to be subpar. Existing TS forecasting algorithms cannot adapt to unforeseen changes in data streams and could perform quite poorly when used in areas of high volatility (In this case: the forecasting of BTC).

To aid with further research on this new concept of neural ODE networks, it is identified that the building of an LTC and its application on an ML domain that still can struggle could be the stepping stone for future intelligent systems – and as a supplement, provide hope to crypto investors for easier predictions.

3.1 Problem Statement

Existing TS forecasting systems cannot adapt to incoming data streams with unexpected changes and characteristics that are different to the data they were trained on. Implementing an algorithm capable of having the mentioned *"liquid"* adaptability could be an advancement for more capable, accurate and interpretable TS forecasting systems.

4. RESEARCH MOTIVATION

The field of AI, particularly neural networks, has been growing exponentially recently, alongside intriguing research. However, as mentioned by Hasani et al., (2020), the issue of networks being static and unable to adapt to varying characteristics could prove to be a limitation for the future of intelligent systems, TS in particular. This research, therefore, is expected to facilitate further exploration by attempting to aid in driving the domain of TS forward.

5. RELATED WORK

Since there is no existing work on LTCs, the author will break down work towards general TS forecasting and its application in BTC forecasting.

Table 1: Related Work

Citation	Summary	Contributions	Limitations
TS forecasting (general)			
(Hochreiter and Schmidhuber, 1997)	LSTM. An algorithm that learns to bridge minimal time lags by enforcing constant error flows. It learns much faster, creates more successful runs and can solve complex tasks that have not been solved before.	Improved performance for short sequence predictions. Overcame error back-flow problems present in conventional BPTT, where they tended to blow up or vanish.	Prediction capacity limits long sequence performance, where the MSE and RMSE rise unacceptably. Therefore, is not an ideal solution for predictions of the distant future.
("Autoregressive Integrated Moving Average (ARIMA)", 2021)	ARIMA. A statistical analysis model to understand the dataset or predict future trends. This model depends on past values to predict the future and uses lagged moving averages to smoothen the data.	Improved performance for TS forecasting data that correlate with values ahead of time and before.	Does not handle well with nonlinear data and long-term forecasting. Further, it performs best on univariate analysis.
(Oreshkin et al., 2020)	N-Beats. An architecture that solves the univariate time series point forecasting problem. It carries some benefits some of which are being understandable, easily applicable to multiple other fields and being fast to train.	Outperformed the M4 competition's winner and improved statistical benchmark forecast accuracy.	Tailored specifically for univariate TS analysis, therefore, would not perform well on multivariate analysis.

Existing algorithms all exhibit static behaviour			
BTC forecasting			
(Roy et al., 2018)	Applied statistical analysis to predict the price of BTC using data from 2013 to 2017. Applied the ARIMA model and obtained an overall accuracy of 90% for deciding weighted cost volatility.	Improved overall insights obtained and added context to future predictions based on past values, alongside scoring an overall lower RMSE than other ML solutions.	Trained on data only between 2013 and 2017, capable of 10 consecutive day predictions and does not consider other input parameters.
(Rizwan et al., 2019)	Compared the usage of LSTM and ARIMA models for the prediction of BTC, however, found that these models aren't very efficient. Used GRU and eventually gained a higher overall accuracy.	Improved existing models built using RNN and LSTM by producing better accuracy and lower MSE, considered other parameters (high, low, open) alongside taking much less time to train.	Trained on data only between 2014 and 2019.
(Fleischer et al., 2022)	Focused on volatility and understanding the behaviour of cryptocurrencies. Trained an LSTM model using BTC close price values to predict future prices.	Beat performance of ARIMA on longer runtime training.	Limited to univariate: does not consider other input parameters, and is capable of forecasting only one day.
All the above work has the limitation of not being updated with the latest available data.			

6. RESEARCH GAP

The literature defines only a single paper for the proposed algorithmic solution - where every other piece of work is not directly related to the algorithm - but is to the family of neural ODEs (CT-

RNN and CT-GRU) and the secondary problem domain of cryptocurrencies and TS. In addition, no algorithmic solution exists for the proposed LTC architecture for model implementation.

It is also worth noting that, because of this, existing forecasting solutions are all implemented using traditional deep neural net approaches (Ex: LSTM ((Hochreiter and Schmidhuber, 1997)) that are static and hence have the limitation of not being able to learn and adapt during inference (Hasani et al., 2020), which results in the model's accuracy degrading overtime – a "*data drift*" (Poulopoulos, 2021).

7. RESEARCH CONTRIBUTION

7.1 Research Domain Contribution

An implementation of the LTC algorithm will be developed, following the architecture proposed, to facilitate the model creation. Additionally, the algorithm built will be generalized without being problem-specific so that it could be applied elsewhere - to evaluate its performance and identify whether the LTC would also be an advancement to other domains.

In addition, the hypothesis of whether it would be an advancement for TS forecasting will be evaluated by identifying whether the newly developed LTC does provide strong robustness and accuracy and outperforms currently existing TS forecasting approaches. Or whether it could be enhanced to be used in other domains altogether.

7.2 Problem Domain Contribution

Having understood the issues in the current literature, it is likely that a solution capable of solving the mentioned issues could be an advancement for future research. Being able to adapt to unforeseen changes and being highly expressive could mean that the highly volatile market of cryptocurrencies would be able to be predicted much more efficiently and be the way forward for investors.

8. RESEARCH CHALLENGE

Existing architectures scale up, and the LTC scales down - with more expressive nodes. Having adapted to the "deeper is usually better" mindset of deep neural nets, a challenge opens up in identifying the requirement of scaling down and what a neural ODE aims to solve.

LTCs are a new approach with only a single research paper regarding its proposed solution. Currently, it is only in the experimental stage and utilizes a novel formulation compared to other existing neural ODEs. The broader domain of neural ODEs is also relatively new; hence the scarcity of references could create more challenges for further research or implementation of systems.

Currently, existing TS forecasting systems are built using ensemble statistical methods or traditional neural net architectures. This creates a new challenge where neural ODEs have not been utilized in implementation yet.

The chosen domain of application is an open system. Open system forecasting is usually poor and is generally difficult to beat the Naïve forecast (A naive forecast is not necessarily bad, 2014).

9. RESEARCH QUESTIONS

RQ1: What are the recent advancements in neural ODEs that can be considered when building the LTC algorithm?

RQ2: How well does the implemented model justify the mentioned hypothesis?

RQ3: What will the implemented algorithm contribute to TS forecasting?

10. RESEARCH AIM

The aim of this research is to design, develop & evaluate the LTC algorithm in a way such that it is capable of building intelligent systems by developing a novel approach to TS forecasting, which could be the stepping stone to be further expanded to other domains as well.

Specifically, this research project will produce a TS forecasting system utilizing the said algorithm, focused on the forecasting of BTC.

The researched knowledge will be put forward and the hypothesis of whether the LTC algorithm can be applied to the selected domain will be evaluated.

11. RESEARCH OBJECTIVES

The accomplishment of the ensuing research objectives is anticipated to meet the aims and provide answers to the research questions listed above. These goals represent milestones that must be achieved for the research to be considered successful.

Table 2: Research Objectives

Objective	Description	Learning Outcomes	Research Questions
Literature Review	Collate relevant information by reading, understanding and evaluating previous work • **R**O1: Conduct preliminary studies and investigations on existing TS forecasting systems. • **R**O2: Analyze the requirement for specialized TS algorithms. • **R**O3: Conduct a preliminary study on ODEs & LTCs. • **R**O4: Obtain deep insights into the architecture behind the LTC.	LO2, LO4, LO5	RQ1
Requirement Analysis	Collect and analyze project requirements, using appropriate tools and techniques • **R**O1: Gather requirements and architectures of LTCs. • **R**O2: Collate the most up-to-date details of BTC. • **R**O3: Get insights from technology and domain experts.	LO1, LO2, LO3	RQ1
Design	Design the architecture and a corresponding system capable of effectively solving the identified problems.	LO1	RQ2

	• **R**O1: Design an efficient approach for the LTC algorithm. • **R**O2: Design an automated flow to update the built network with the latest data. • **R**O3: Design an ML pipeline for easy deployments.		
Implementation	Implement a system that is capable of addressing the mentioned research gaps. • **R**O1: Implement the LTC algorithm in a way capable of model building. • **R**O2: Integrate the developed algorithm into a TS forecasting application. • **R**O3: Integrate the developed intelligent system into the prototype to display forecasts.	LO1, LO5, LO6, LO7	RQ2
Evaluation	Effectively test the implemented algorithm, the system, and the respective data science model using recommended techniques. • **R**O1: Create a test plan & test cases and perform unit, performance and integration testing. • **R**O2: Evaluate the developed algorithm and the respective model against the mentioned benchmarking metrics.	LO4	RQ2, RQ3
Documentation	Document the progression of the research project and notify of any faced challenges.	LO6, LO8	-

12. PROJECT SCOPE

Concerning the granted time for this research project, the scope is as follows.

12.1 In-scope

- Implementation of the LTC algorithm capable of being used like currently existing solutions (Ex: LSTMs), and the corresponding creation of a system.
- Periodical updates of the model with the latest available data.
- Evaluation and comparison of the implemented system against currently existing solutions to validate or invalidate the hypothesis.

12.2. Out-scope

- Application of the implemented algorithm in other domains to justify whether it could be an advancement in those domains as well.
- Forecast multiple different cryptocurrencies.
- Usage of live, on-demand data instead of daily data & incremental learning.
- Ability to consider other external factors, such as social media, legislation and laws, and country advertisements for handling digital currency.

12.3 Desirables

- Evaluate implementation against the M4 competition to further justify the future of TS forecasting algorithms.
- Evaluate other neural ODEs (CT-RNN, CT-GRU, Latent ODE) for time series forecasting and compare them with the LTC.
- Consider twitter volume and the “block reward size” as external factors by combining them with the BTC historical data.

12.4 Prototype Diagram

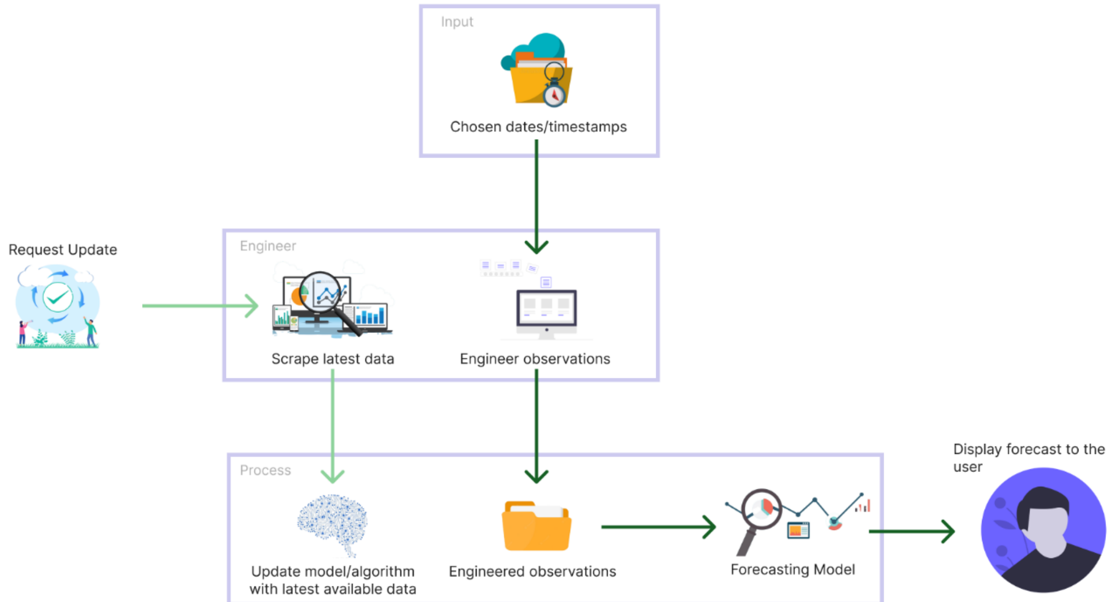

Figure 1: Prototype Feature Diagram (*Self-Composed*)

13. PROPOSED METHODOLOGY

13.1 Research Methodology

Methodologies suitable for the research project have been evaluated and chosen from the predefined Saunders Research Onion Model (Saunders, Lewis, and Thornhill, 2007, p102).

Table 3: Research Methodology

Philosophy	The **Pragmatism** philosophy was chosen since the research question is of paramount importance. Additionally, as the outcome of this research, it is expected to validate/invalidate the developed hypothesis alongside necessary benchmarking comparisons.
Approach	The **deductive** approach was chosen over the inductive since the final analysis and evaluation will be quantitative that aims to deduce the hypothesis.
Strategy	**Archival Research** and **Action Research** were chosen as the strategy for data collection. Archival Research since the research topic is more modern, hence the principal source of data collection would be research documents. Action

	Research will also be included since the development process will likely be an iterative approach of diagnosis, planning, taking action & evaluation.
Choice	**Multi-method** will suit the proposed research project most since qualitative analysis would be a suitable supplement to the primary quantitative approach, however, will not be used as a combination.
Time Horizon	The **Cross-Sectional** time horizon was chosen over the longitudinal time horizon. Even though the latest available data will have to be obtained often to update the model, there will be no interlinking between the times when the data is gathered as they will be independent of each other.
Techniques and procedures	As a form of **Data Collection & Analysis**, as many sources as possible will be used since there are finite resources. Statistics, reports, journals, articles and observations will be the primary mediums.

13.2 Development Methodology

13.2.1 Life Cycle Model

Agile was chosen as the research development life cycle to implement the prototype since heavy iterative development is required.

13.2.2 Design Methodology

Object-Oriented Analysis & Design (OOAD) was chosen as the Design Methodology since it supports increments and extensions with reusability.

13.2.3 Software Development Methodology

Object Oriented Programming (OOP) & structural programming will be used to accompany the OOAD Design Methodology and create modules of reusable code which can be interlinked.

13.2.4 Evaluation Methodology

Based on research, the most suitable method for evaluating TS forecasting systems is a Walk-forward validation. This is identified to be a realistic way of evaluation since the model must be updated once new data is available (Falessi et al., 2020). Additionally, a specialized version of the K-fold cross-validation: cross-validation on a rolling basis (Shrivastava, 2020) will also be used.

Benchmarking

MAE, RMSE, (s)MAPE and MASE (Hyndman et al., 2021) will be used to benchmark the system to produce adequate comparisons against existing solutions and validate or invalidate the hypothesis.

13.3 Solution Methodology

As mentioned, a BTC forecasting prototype will be built to create justification.

A summarized workflow that will be followed upon creating the model is depicted below.

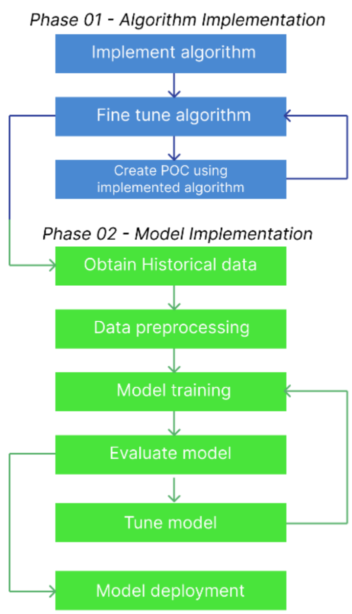

Figure 2: Model creation workflow (*Self-Composed*)

Each of the above-depicted steps is elaborated below.

Phase 01

13.3.1 Implement Algorithm

The first, and most important step is to implement the LTC algorithm. This step is critical since it will give the author an idea of whether the development is feasible, whether pivoting is necessary, or even if the project must change entirely. Furthermore, it must be done initially since the steps that follow all depend on the mentioned algorithm. The paper authored by Hasani et al., (2020), will be used as a guide to developing a sample of the LTC.

13.3.2 Fine Tune Algorithm

Once satisfactory progress has been made the code must be cleaned and fine-tuned so that it is scalable and generalizable.

13.3.3 Create Example POC

An example POC must be implemented to validate whether the supplementary forecasting application is feasible. This step is also important since it will give the author an idea of how the software will have to be built.

The creation of the POC and fine-tuning will be an iterative process since minor tweaks will be done whilst developing the POC.

Phase 02

13.3.4 Obtain Historical Data

As identified in the literature: existing systems had been trained on data that are outdated now. To address this limitation, the data in this project will be scraped using an API, which will be the most up-to-date.

Furthermore, to keep the model as updated as possible, the model will be retrained periodically with the existing new data.

13.3.5 Data Preprocessing

Once the data has been fetched it must be cleaned. The API returns redundant & unneeded columns (ex: repeated features with different names) that must first be removed.

Processing of data for TS forecasting applications is not the same as classification or regression problems since the data is temporal – therefore, the order must be given prominence.

The creation of the train and test sets is not similar to other problems, as random splits will not work. The data will be split sequentially, at a point in time such that the observations before it is the train data and after it the test data - a “pseudo future”. It is so that there is no “leakage” between the two sets (Hyndman et al., 2021): the past data must forecast the future.

Finally, the data must be “windowed” to convert it into a supervised learning problem and split into features and labels (BI4ALL, 2021). This is required since windows of the past will predict the future.

13.3.6 Model Training

Once the data windows are ready, the model can be created. Here, the developed LTC cell will be used within an RNN layer to provide a fair comparison against other existing cells like the LSTM.

13.3.7 Evaluation

Once the model has been trained, sufficient evaluation & benchmarking must be conducted to shed light on the model's performance. The model will be evaluated and benchmarked against metrics discussed under the Evaluation Methodology.

13.3.8 Tuning

If the performance obtained is sub-par, the model's hyperparameters must be tuned (Ex: no. of epochs, batch size, learning rate, optimizer, activation function, no. of units & layers). Tuning mentioned hyperparameters could cause a drastic change in performance – even worsen the performance. However, this is an important step that must be carried out, as it could drastically improve performance.

Training, Evaluation & Tuning will be an iterative process, as it is unlikely to obtain the best-performing model in the first experiment. It will also be unexpectedly long since there exists no algorithm of the LTC and solution. Therefore, "common" hyperparameter values documented for other algorithms do not exist.

13.3.9 Deployment

The final step in the implementation is to deploy the forecasting model so that it is available to be accessed from anywhere – in this case, especially the client application.

In addition, a deployment pipeline must be built to facilitate automatic future deployments whenever the model is updated periodically.

13.4 Project Management Methodology

A combination of PRINCE2 and Agile will be followed by the author. The project will require many iterations and improvements since the implementation is novel and there exists no reference. Alongside multiple iterations, it's best implemented by being divided into multiple chunks and focusing on each chunk at a time with a plan-based approach.

13.4.1 Schedule

Gantt Chart

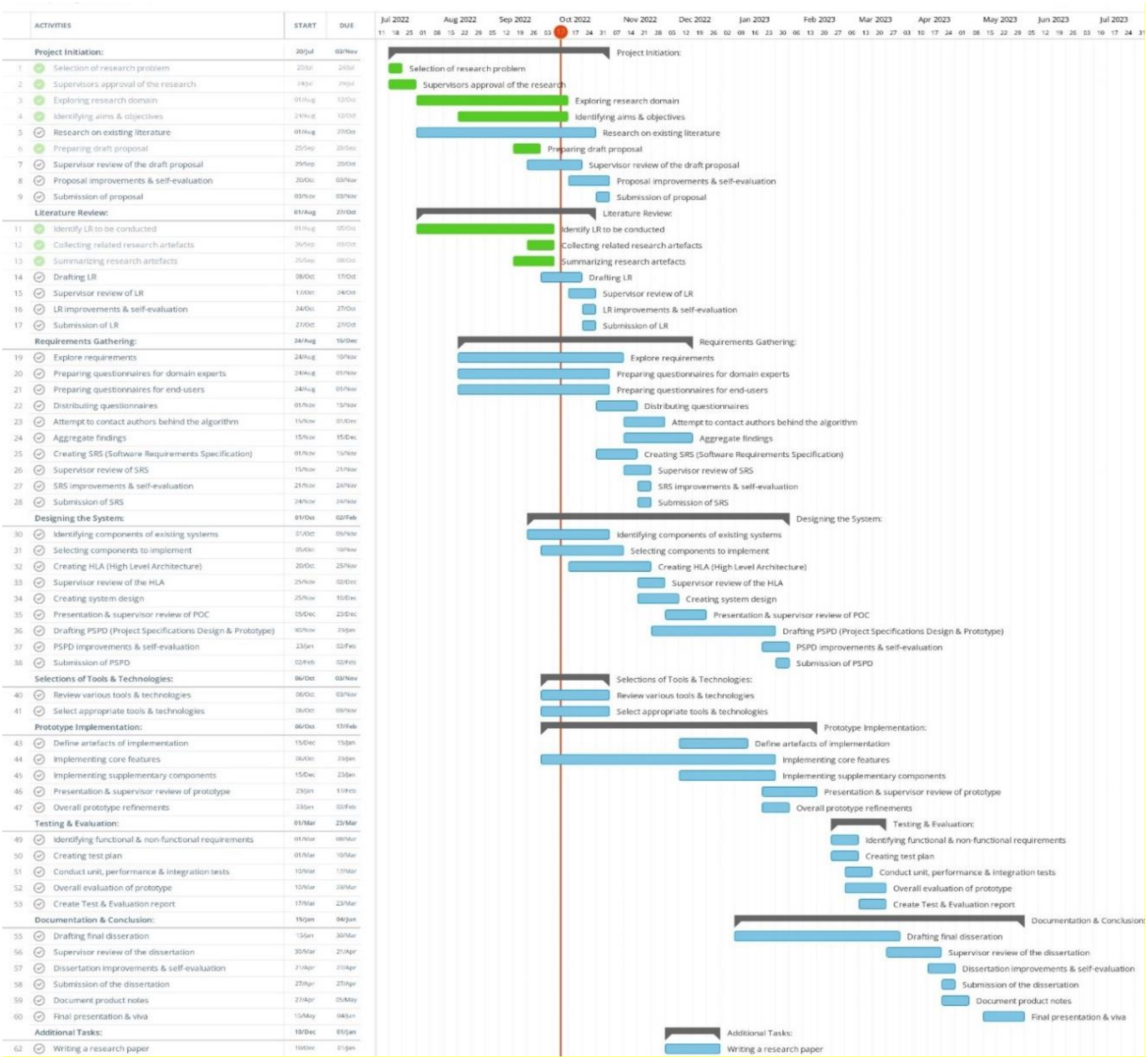

Figure 3: Gantt Chart (*Self-Composed*)

Deliverables

Table 4: Deliverables & Dates

Deliverable	**Date**
Literature Review Critical analysis of related work & solutions.	27th October 2022

Project Proposal & Ethics Forms The initial proposal of the research to be conducted.	3rd November 2022
Software Requirement Specification Defines the requirements that must be met to prototype and collect data.	24th November 2022
Proof Of Concept & Implementation Presentation An initial Implementation of the proposed system.	23rd December 2022
Project Specifications Design & Prototype A prototype of the system with the core features and an accompanying document specifying the design followed & an overview of the implemented algorithm.	2nd February 2023
Test & Evaluation Report Documentation of test findings and evaluations conducted on the prototype.	23rd May 2023
Draft Project Report A draft submission of the final Thesis to get evaluations.	30th May 2023
Final Thesis Final submission of the thesis with complete documentation of the project's journey.	27th April 2023

13.4.2 Resource Requirements

Software Requirements

- **Operating System (Windows / Linux / macOS)** – Windows will be the default since it provides easy access to required development environments and tools.

- **Python / R** – Will be used to create the network & the respective model. Python will be used since it has a much smaller learning curve and provides easy integration with other mentioned software.
- **TensorFlow** – Provides libraries that facilitate DL in Python & R.
- **Flask / Node** – For seamless communication and integration between the client and the model. Flask will be the primary choice since the ML component will also be built using Python.
- **React / Angular / Vue** – To develop the client side of the application. A fast performant library is required to prevent lags and other performance issues. React will be the option only because of the author's familiarity, else it does not have any impact whatsoever.
- **VSCode | PyCharm** – Environment to facilitate application development.
- **Google Colab / Jupyter Notebook** – Development environment for building the forecasting model.
- **Zotero / Mendeley** – Manage references and research artefacts.
- **Overleaf | MS Office | GSuite | Figma | Canva | Draw.io** – Tools to create reports, figures, diagrams & documents and backup artefacts.
- **GitHub / Bitbucket** – Track, version & manage development code & research documents. GitHub will be the choice also due to the author's familiarity.

Hardware Requirements

- **Core i5 Processor (8^{th} gen) or above** – for long-running intensive workloads.
- **8GB Ram or above** – to manage model training, multiple development environments & multitasking.
- **Disk space of approx. 20GB** – to store application code & data.

Data Requirements

- **BTC price observations** – scraped from a financial website (Ex: investing.com).

Skill Requirements

- Creation of TS forecasting systems.
- Knowledge on ODEs & ODE solvers.

- Implementation of a raw neural ODE.
- Ability to create optimized & scalable DL models.
- Ability to develop optimized client-side charts & user interfaces that dynamically update.
- Research & Academic writing skills.

13.4.3 Risk Management

The following table identifies possible risks the author could face and how they could mitigate them.

Table 5: Risk Management Plan

Risk Item	Severity	Magnitude	Mitigation Plan
Lose access to development code	5	2	Backup code on source control and cloud storage.
Invalid hypothesis	3	2	Continue researching since the final output is a research contribution regardless.
Corrupted documentation	4	4	Store all necessary documentation on the cloud as well as external storage.
Inability to deliver all expected deliverables	4	2	Follow a list of priorities and deliver accordingly.
Lack of required knowledge	5	5	Get insights from domain experts and, if necessary, the author of the proposed algorithm.

REFERENCES

S. Nakamoto, (2020). *Bitcoin: A peer-to-peer electronic cash system*. Available from https://bitcoin.org/bitcoin.pdf [Accessed 25 September 2022].

Rahouti, M., Xiong, K. and Ghani, N. (2018). Bitcoin Concepts, Threats, and Machine-Learning Security Solutions. *IEEE Access*, 6, 67189–67205. Available from https://doi.org/10.1109/ACCESS.2018.2874539 [Accessed 25 September 2022].

Kervanci, I. sibel and Akay, F. (2020). Review on Bitcoin Price Prediction Using Machine Learning and Statistical Methods. *Sakarya University Journal of Computer and Information Sciences*. Available from https://doi.org/10.35377/saucis.03.03.774276 [Accessed 25 September 2022].

Makridakis, S., Spiliotis, E. and Assimakopoulos, V. (2018a). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13 (3), e0194889. Available from https://doi.org/10.1371/journal.pone.0194889 [Accessed 25 September 2022].

Makridakis, S., Spiliotis, E. and Assimakopoulos, V. (2018b). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34 (4), 802–808. Available from https://doi.org/10.1016/j.ijforecast.2018.06.001 [Accessed 25 September 2022].

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36 (1), 75–85. Available from https://doi.org/10.1016/j.ijforecast.2019.03.017 [Accessed 25 September 2022].

Hasani, R. et al. (2020). Liquid Time-constant Networks. Available from https://doi.org/10.48550/arXiv.2006.04439 [Accessed 25 September 2022].

Chen, R.T.Q. et al. (2019). Neural Ordinary Differential Equations. Available from https://doi.org/10.48550/arXiv.1806.07366 [Accessed 25 September 2022].

Poulopoulos, D. (2021). Is “Liquid” ML the answer to autonomous driving? *Medium*. Available from https://towardsdatascience.com/is-liquid-ml-the-answer-to-autonomous-driving-bf2e899a9065 [Accessed 25 September 2022].

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9 (8), 1735–1780. Available from https://doi.org/10.1162/neco.1997.9.8.1735 [Accessed 25 September 2022].

Oreshkin, B.N. et al. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. Available from http://arxiv.org/abs/1905.10437 [Accessed 26 September 2022].

Autoregressive Integrated Moving Average (ARIMA). (2021). *Investopedia*. Available from https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp [Accessed 26 September 2022].

Roy, S., Nanjiba, S. and Chakrabarty, A. (2018). Bitcoin Price Forecasting Using Time Series Analysis. *2018 21st International Conference of Computer and Information Technology (ICCIT)*. December 2018. Dhaka, Bangladesh: IEEE, 1–5. Available from https://doi.org/10.1109/ICCITECHN.2018.8631923 [Accessed 25 September 2022].

Rizwan, M., Narejo, S. and Javed, M. (2019). Bitcoin price prediction using Deep Learning Algorithm. *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. December 2019. Karachi, Pakistan: IEEE, 1–7. Available from https://doi.org/10.1109/MACS48846.2019.9024772 [Accessed 26 September 2022].

Fleischer, J.P. et al. (2022). Time Series Analysis of Cryptocurrency Prices Using Long Short-Term Memory. *Algorithms*, 15 (7), 230. Available from https://doi.org/10.3390/a15070230 [Accessed 26 September 2022].

Saunders, M.N.K., Lewis, P. and Thornhill, A. (2007). *Research methods for business students*, 4th ed. Harlow, England ; New York: Financial Times/Prentice Hall.

Falessi, D. et al. (2020). On the Need of Preserving Order of Data When Validating Within-Project Defect Classifiers. Available from http://arxiv.org/abs/1809.01510 [Accessed 27 September 2022].

Shrivastava, S. (2020). Cross Validation in Time Series. *Medium*. Available from https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4 [Accessed 12 October 2022].

BI4ALL. (2021). Supervised Machine Learning in Time Series Forecasting. *BI4ALL - Turning Data Into Insights*. Available from https://www.bi4all.pt/en/news/en-blog/supervised-machine-learning-in-time-series-forecasting/ [Accessed 12 October 2022].

Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. Available from https://otexts.com/fpp3/. [Accessed on 30 Sep. 2022].

Hasani, R. et al. (2021). Liquid Neural Networks. *YouTube*. Available from https://www.youtube.com/watch?v=IlliqYiRhMU&t=350s. [Accessed on 30 Sep. 2022].

Mozer, M.C., Kazakov, D. and Lindsey, R.V. (2017). Discrete Event, Continuous Time RNNs. Available from https://doi.org/10.48550/ARXIV.1710.04110 [Accessed 14 October 2022].

Funahashi, K. and Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6 (6), 801–806. Available from https://doi.org/10.1016/S0893-6080(05)80125-X [Accessed 14 October 2022].

A naive forecast is not necessarily bad. (2014). *The Business Forecasting Deal*. Available from https://blogs.sas.com/content/forecasting/2014/04/30/a-naive-forecast-is-not-necessarily-bad/ [Accessed 15 October 2022].