31st May, 2023

University of Westminster
School of Computing

SUPERVISOR

Mr. Torin Wirasingha

STUDENT

Ammar Raneez
2019163 | W1761196

Stochronetics: Surpassing Time Series Forecasting Limitations using Liquid Time-stochasticity Networks

Phase 01

I Problem Background

II Existing Work

III Research Gap

IV Algorithm Design

V Research Aim

VI Training These Networks

VII Algorithm Complexity

Problem Background

Time series forecasting

- TS forecasting is a significant business issue and an area where ML could create an impact (Jain, 2017).
- Although ML and DL have outperformed classical approaches for NLP and computer vision, the domain of TS still seems to be a struggle compared to classical statistical methodologies (Makridakis et al., 2018a;b).

Existing TS forecasting systems cannot adapt to incoming data streams with unexpected changes and characteristics that are different from the data on which they were trained

Existing Work

Time series forecasting algorithms

- Existing forecasting solutions are all implemented using traditional deep neural net architectures.

Existing work

- Neural ordinary differential equations (Chen et al., 2019).
- Liquid Time-Constant networks (Hasani et al., 2020), solved this to some extent.

There's a limitation here: areas with greater randomness - instantaneous, miniscule changes cannot be modelled (Raneez and Wirasingha, 2023).

Research Gap

Time series forecasting algorithms

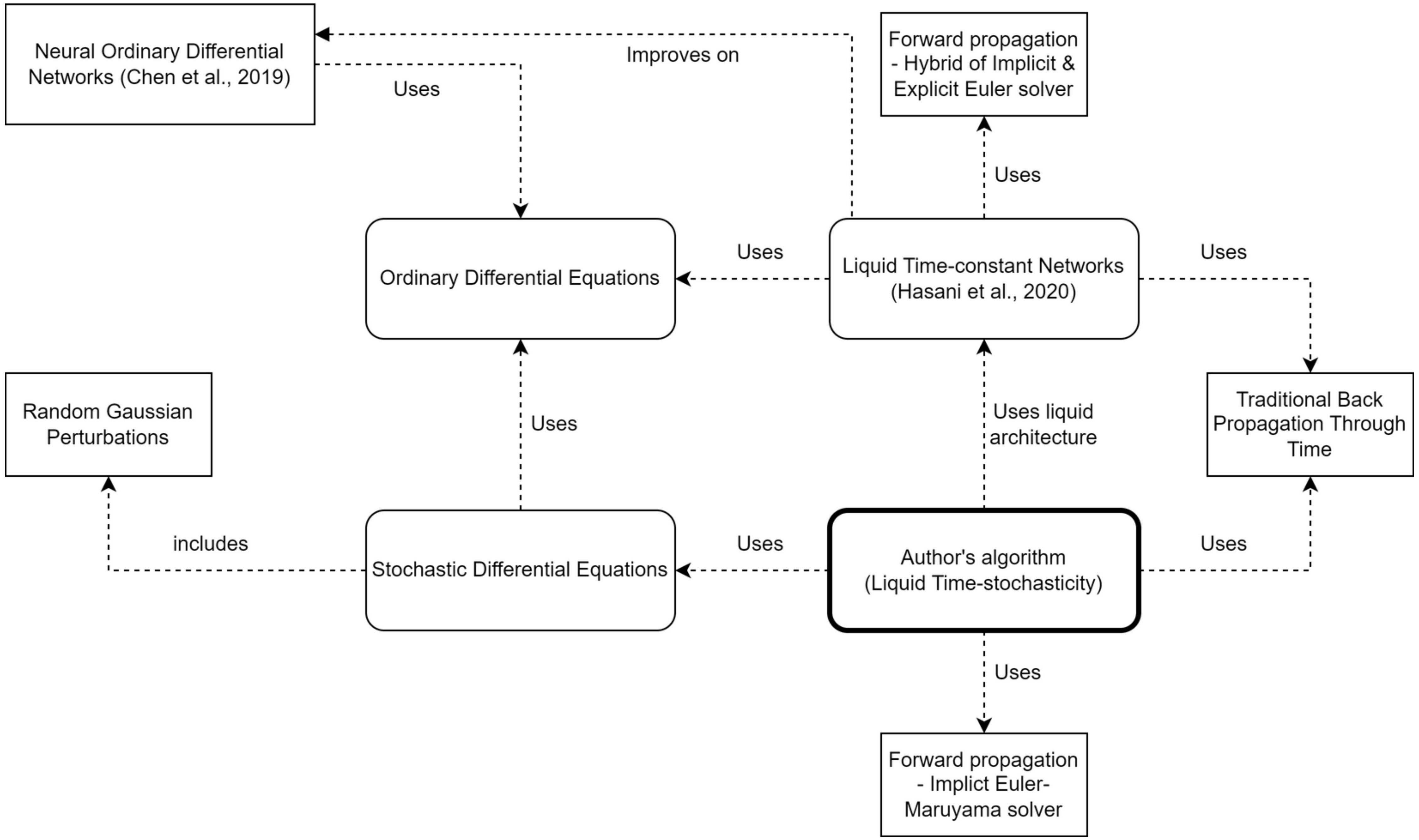
- Existing forecasting solutions are all implemented using traditional deep neural net approaches (ex: LSTM, GRU) that are static and hence have the limitation of not being able to learn and adapt during inference (Hasani et al., 2021), which results in the model's accuracy degrading overtime – a *'data drift'* (Poulopoulos, 2021).

LTC

- The proposed LTC architecture uses a sequence of linear ODEs, which are now considered obsolete and lack rapid adaptability (Duvenaud, 2021).

Algorithm Design

Stochastic differential equations can be used instead of ordinary differential equations!



Research Aim

Research aim

- *The aim of this research is to design, develop & evaluate the author-proposed LTS algorithm for TS forecasting, which could be the stepping stone for breaking TS forecasting limitations.*

Specifically, this research project will produce a TS forecasting system utilizing the LTS algorithm to forecast BTC.

Training These Networks

$$\boxed{\frac{dx(t)}{dt} = -\left[\frac{1}{\tau} + f(x(t), I(t), t, \theta) - \sigma B\right] x(t) + f(x(t), I(t), t, \theta) A}$$

(Raneez and Wirasingha, 2023)

To solve the state of these SDEs, implicit Euler-Maruyama solver can be used (Raneez and Wirasingha, 2023).

And for backpropogation, the traditional Backpropogation Through Time approach (Raneez and Wirasingha, 2023).

Algorithm Complexity

	BPTT	**Adjoint sensitivity**
Time	O(L)	**O(LlogL)**
Memory	O(L)	**O(1)**
Forward accuracy	High	High
Backward accuracy	**High**	Low

(Raneez and Wirasingha, 2023)

Phase 02

I Problem Domain

II Research Gap

III Insights

IV Proposed Architecture & Tech Stack

V Evaluation

Problem Domain

Cryptocurrencies

- The word ‘crypto’ has been an enormous buzzword recently, especially BTC. It has even come to the point where crypto and BTC are used interchangeably (Rahouti et al., 2018).
- Its a fully decentralized means of exchange/digital currency.

However, being at the forefront of the digital currency world, it has developed high volatility, making it difficult to predict future rates (Kervanci and Akay, 2020).

Research Gap

Bitcoin forecasting

- The work available on BTC forecasting has yet to consider exogenous factors that could have an impact (Roy et al., 2018; Rizwan et al., 2019; Fleisher et al., 2022). Therefore, a significant concern is that they cannot adapt well.

Factors that could influence the price are as follows (Abraham et al., 2018):

- Tweet sentiment & volume
- Google Trends

Insights

Forecasting the price of crypto is implausible without considering other factors other than the past historical prices.

Therefore, include the following factors as well:

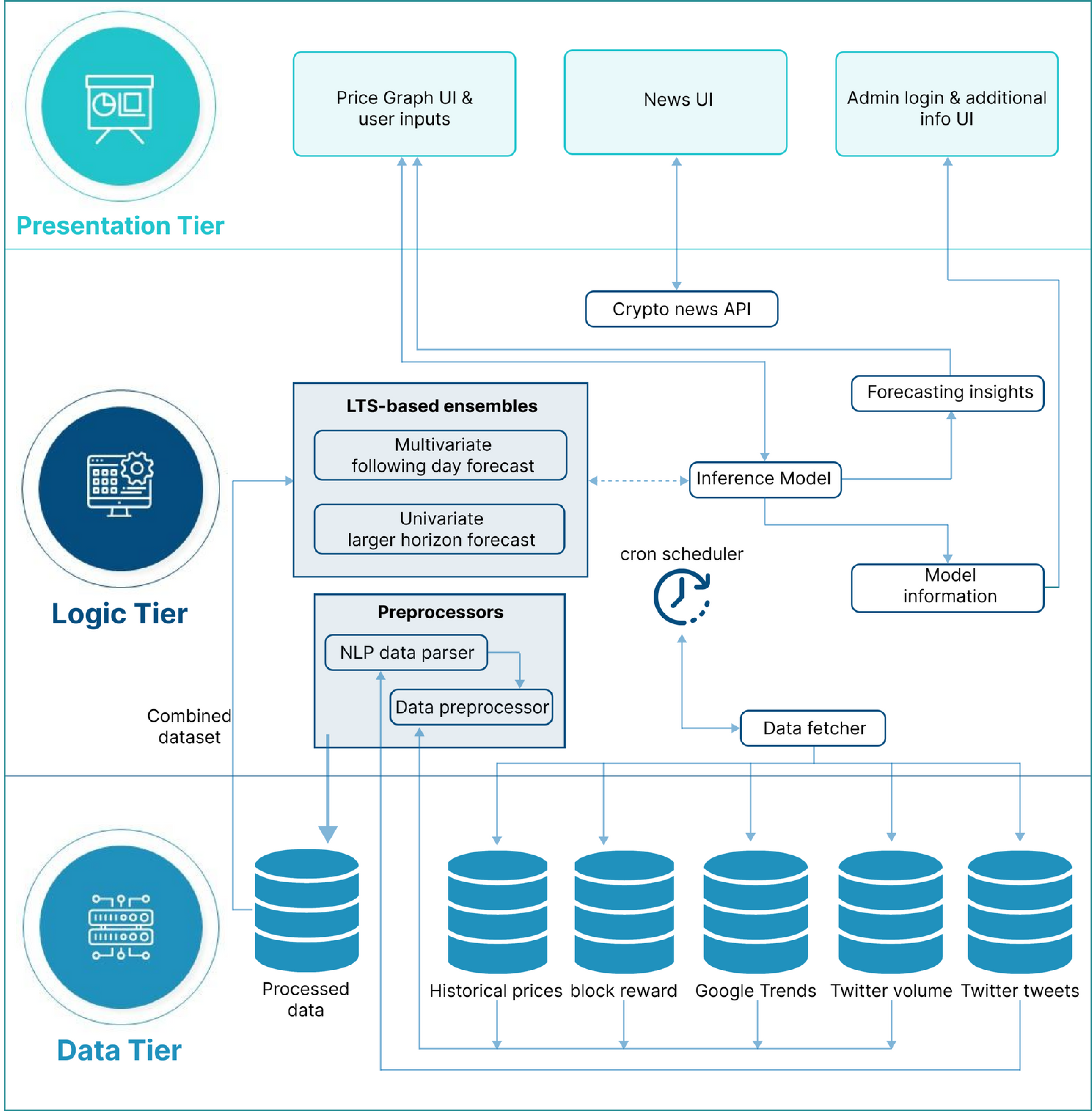
- Google Trends
- Twitter tweet volume
- Twitter thoughts/sentiments
- Block reward size

$$influencer_{sum} = \alpha \log_{10}(followers_{count} + 1) + \beta \log_{10}(lists_{count} + 1)$$

$$tweet_{sum} = \gamma \log_{10}(retweets_{count} + 1) + \delta \log_{10}(like_{count} + 1)$$

$$\boxed{weighted_{score} = \frac{tweet_{sum} + influencer_{sum}}{tweet_{sum} + influencer_{sum} + 1} * compound_{score}}$$

Proposed Architecture



Tech Stack


Presentation Tier
JS
Logic Tier
Flask
BeautifulSoup
Lingua
matplotlib
kaggle
seaborn
HEROKU
scikit learn
jupyter
Data Tier
pandas
Amazon S3
mongoDB
Investing.com
RapidAPI

Evaluation

	MAE	MSE	RMSE	MAPE	MASE
Traditional architectures					
Basic dense	1227	2882849	1697	3.06%	1.07
2x dense	1146	2628006	1621	2.98%	1.05
Stacked dense	1147	2604766	1613	2.88%	1.04
Conv1D	1153	2653370	1628	2.90%	1.02
LSTM	1216	2834756	1683	3.07%	1.06
N-BEATS	1142	2614896	1617	2.86%	1.03
Benchmark & LTS Ensemble					
Naïve forecast	951	2021966	1421	2.56%	1.00
Ensemble	950	2013928	1419	2.56%	0.99

Phase 03

I	Application Demo
II	Future Work
III	Conclusion

Application Demo

Future Work

Algorithm

- Test LTS with other SDE solvers.
- The LTS can use a hybrid SDE solver that combines the implicit and explicit Euler-Maruyama solvers.
- LTS with reverse-mode AD must be evaluated instead of the proposed BPTT approach to determine memory and time efficiency.

Application

- Identify and determine how other features impact the price.
- Enhance the Twitter sentiment weighting formula to consider more factors.

Conclusion

Research contribution

- SDE-based liquid neural network

Problem domain contributions

- Twitter sentiment weighing formula
- Multiple exogenous features BTC forecasting model

Additional contributions

- Data extraction/processing scripts for various sources
- Custom LTC & LTS Keras layers ready to be integrated

New skills

- *Neural network building blocks*
- *Advanced calculus*
- *Data scraping & mining*

Existing skills

- *Full-stack development*
- *Intermediate calculus*
- *Traditional ML/DL*

Evaluators

Research domain

- Google Brain visiting researcher and Associate Professor at University of Toronto
- Professor at the University of Melbourne
- Professor at the University of Colombo

Problem domain

- Blockchain and Web 3.0 developer
- Medical doctor & crypto evangelist
- Avid BTC trader and statistician

References

Abraham, J., Higdon, D., Nelson, J. and Ibarra, J. (2018). Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis. *SMU Data Science Review*: Vol. 1: No. 3, Article 1. Available at: https://scholar.smu.edu/datasciencereview/vol1/iss3/1

Chaman L. Jain. Answers to your forecasting questions. *Journal of Business Forecasting*, 36, Spring 2017.

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].

Duvenaud, D (2021). Directions in ML: Latent Stochastic Differential Equations: An Unexplored Model Class. *YouTube*. Available from https://www.youtube.com/watch?v=6iEjF08xgBg. [Accessed on 30 Sep. 2022].

References

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].

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

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].

References

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].

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].

Raneez, A. and Wirasingha, T. (2023). A Review On Breaking the Limits of Time Series Forecasting Algorithms. *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*. 8 March 2023. Las Vegas, NV, USA: IEEE, 0482–0488. Available from https://doi.org/10.1109/CCWC57344.2023.10099071 [Accessed 24 April 2023].

References

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].

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].

Publications

A Review on Breaking the Limits of Time Series Forecasting Algorithms

- IEEE CCWC 2023
- Las Vegas, Nevada
- Extended literature review & Liquid Time-stochasticity proposal
- Accepted & Presented
- `https://doi.org/10.1109/CCWC57344.2023.10099071`

An Analysis of Factors that Contribute to the Price of Bitcoin

- In review; arXiv preprint

Thank you!

Extras

questions.docx

Google Brain

[redacted]

1) Because they have different advantages and disadvantages. Many are obsolete now.
2) Moving computation to continuous time so it can be adaptively approximated.
3) Yes, at least for irregularly-sampled data.
4) They're expensive to compute.
5) I think latent ODEs are basically obsolete, and you should look into latent SDEs instead. You could apply LTC architectures to those more flexible models, instead.

[redacted]

Ammar <ammar.2019163@iit.ac.lk>
to [redacted]

Dear Professor,

This is great, thank you so much!

Regards,
Ammar

Reply Forward

[redacted]

Dear Ammar

Great to hear that you are doing such research.
[redacted] in my team is doing his PhD generally in that area.

[redacted] Could you send some of the key papers (also your NSF paper) to Ammar?

[redacted]
Optimization and Pattern Recognition Research Group
Department of Mechanical Engineering
School of Electrical, Mechanical and Infrastructure Engineering
Level 1, Melbourne Connect (Building 290) University of Melbourne, VIC, 3010, Australia

THE UNIVERSITY OF MELBOURNE

From: Ammar <ammar.2019163@iit.ac.lk>
Sent: Sunday, November 27, 2022 4:07 PM
To: [redacted]
Subject: [EXT] Final Year Project Research

External email: Please exercise caution

Dear Professor,

Ammar <ammar.2019163@iit.ac.lk>
to [redacted]

Dear professor,

Documentation

README.md

BitForecast

vercel passing | Heroku

CodeQL passing | codefactor A

BitForecast is a Bitcoin forecasting application that uses the Liquid Time-stochasticity network described here: https://github.com/Ammar-Raneez/Liquid-Time-stochasticity-networks for its forecasting model. Additionally, it considers multiple exogenous factors, including Twitter volume, Google Trends, Twitter sentiment, and the block reward size, alongside the basic historical prices, to produce a more robust and effective forecast.

Prerequisities

- Node 16+
- Python 3.8+

Monorepos

- client - frontend application built using React & Redux
- server - Flask API server for local trials & testing
- ml - Machine learning experiments, trails & testing
- deployment - Flask API server hosted in Heroku

README.md

Getting started with the deployment server

Development setup

- Create a new virtual environment by `pip -m venv venv`
- Ensure that it is activated by running `source venv/scripts/activate`
- Install requirements by running `pip install requirements.txt`
- Add required configuration - create a `.env` file containing the keys mentioned in the `.env.example`.
- Run the server via `python app.py`

README.md

Getting started with the BitForecast client application

Development setup

- Add required configuration - create a `.env` file containing the keys mentioned in the `.env.example`.
- Install `node_modules` via `npm i` or `yarn`
- Run the project via `npm start` or `yarn start`

README.md

Liquid Time-stochasticity Networks (LTSs)

CodeQL passing | codefactor A+

This is the official repository for Liquid TIme-stochasticity networks described in paper: https://doi.org/10.1109/CCWC57344.2023.10099071

This implementation utilizes the Euler Maruyama solver to perform forward propagation and relies on the conventional backpropagation through-time (BPTT) to train the models.

Prerequisites

The architecture was built using Keras and TensorFlow 2.0+ and Python 3+ on the Windows 11 machine.

Experiments

`experiments/experiments.ipynb` demonstrates a couple of experiments attempting to model bitcoin prices.

Achievement Of Requirements

Functional requirements

- 75%
  - Model updating; multiple crypto forecasts; real-time forecasting not implemented

Non-functional requirements

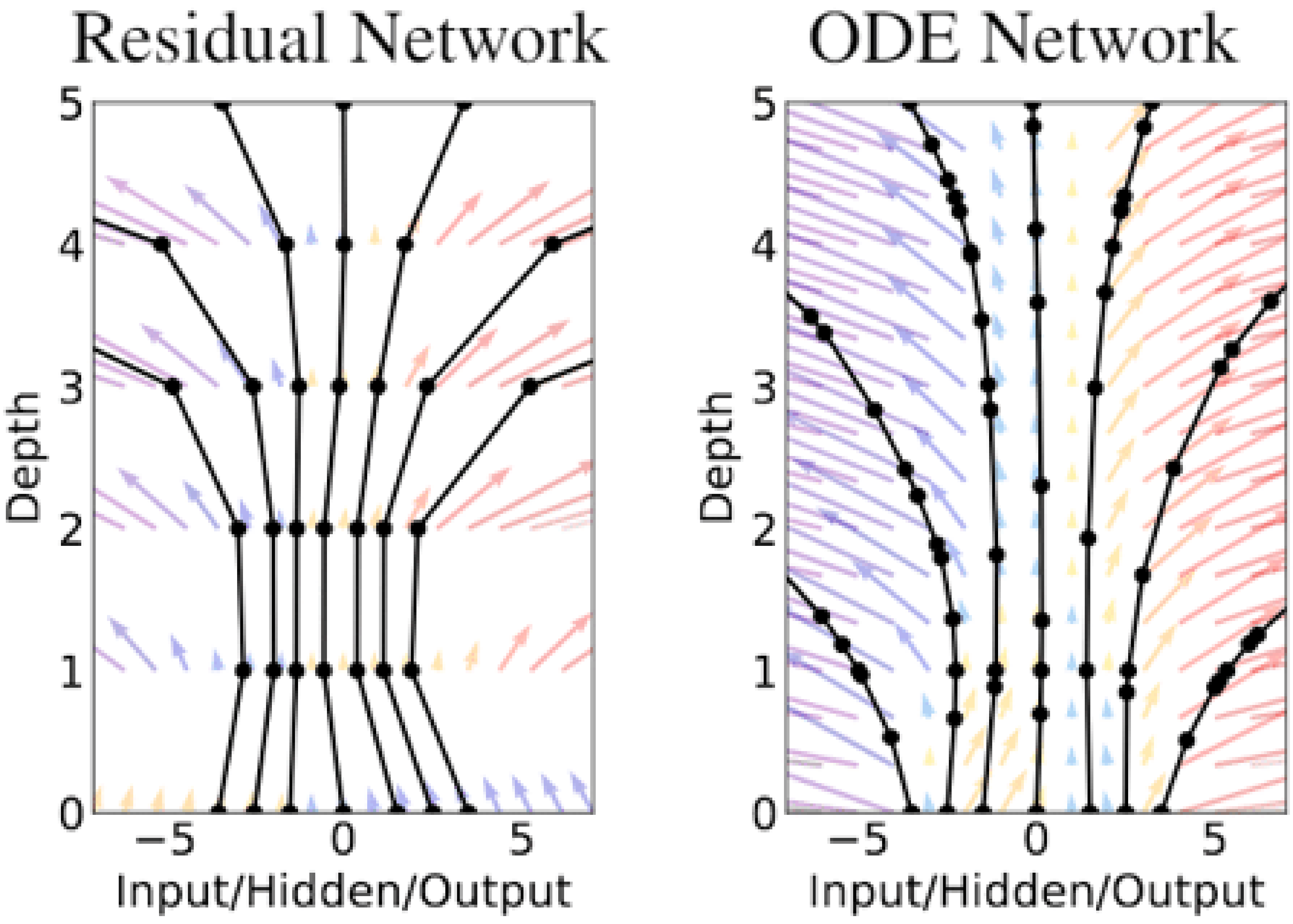
- 83.3%
  - Availability not implemented; Scalability implemented up to an extent

Design goals

- 100%

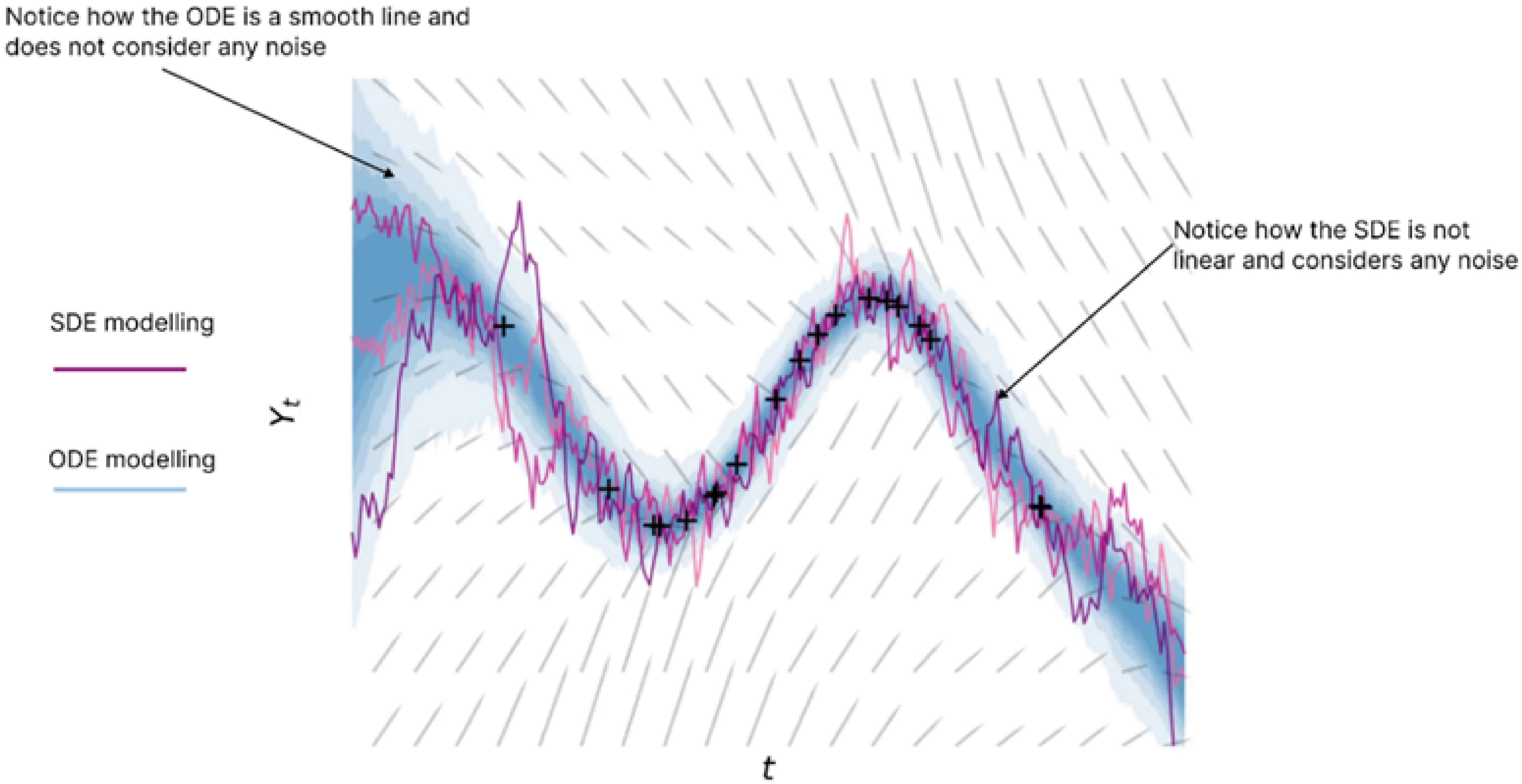
Understanding An ODE

Discrete vs Continuous



(Chen et al., 2019)

Understanding An SDE



(Duvenaud, 2021; Raneez and Wirasingha, 2023)